

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Государственное учреждение
«Администрация Парка высоких технологий»

Открытые семантические технологии проектирования интеллектуальных систем

OSTIS-2016

Open Semantic Technologies for Intelligent Systems

МАТЕРИАЛЫ
VI МЕЖДУНАРОДНОЙ
НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ

(Минск, 18–20 февраля 2016 года)

УДК 004.822+004.89-027.31
ББК 32.813-02+73
О-83

Редакционная коллегия:

*В. В. Голенков (отв. ред.), Л. С. Глоба, В. А. Головки, Н. А. Гулякина, О. П. Кузнецов, Б. М. Лобанов,
Г. С. Осипов, С. В. Смирнов, П. И. Соснин, Д. Ш. Сулейманов, В. Ф. Хорошевский*

Организаторы конференции:

Министерство образования Республики Беларусь
Учреждение образования «Белорусский
государственный университет информатики и
радиоэлектроники»
Российская ассоциация искусственного интеллекта
Государственное учреждение «Администрация Парка
высоких технологий» (Республика Беларусь)
Объединённый институт проблем информатики
Национальной академии наук Беларуси
Министерство связи и информатизации
ФГБНУ «Институт управления образованием
Российской академии образования»

ФГБОУ высшего профессионального образования
«Югорский государственный университет»
ООО «Интеллектуальные процессоры»
ЗАО «Qulix Systems»
ЗАО «Итранзишен»
ОДО «ВирусБлокАда»
ООО «Рубиройд Лэбс»
ООО «Апсилон ай-ти»
ЧТУП «АйДжи Софт Бел»
ООО «Октонион технолоджи»

Техническая и информационная поддержка:

Международный журнал «Программные продукты и
системы»
Научный журнал «Информатика»
Научный журнал «Онтология проектирования»

Научно-технический журнал «Автоматизация процессов
управления»
Научно-практический журнал для специалистов
«Электроника ИНФО»

*Издание осуществлено по заказу государственного учреждения
«Администрация Парка высоких технологий»*

О-83 **Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2016) :** материалы VI междунар. науч.-техн. конф. (Минск, 18–20 февраля 2016 года)/ редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2016. – 596 с.
ISBN 978-985-543-034-7.

Сборник включает прошедшие рецензирование статьи VI международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем».

Сборник предназначен для преподавателей высших учебных заведений, научных сотрудников, студентов, аспирантов, магистрантов, а также для специалистов предприятий в сфере проектирования интеллектуальных систем.

Материалы сборника одобрены Программным комитетом OSTIS-2016 и печатаются в виде, представленном авторами.

УДК 004.822+004.89-027.31
ББК 32.813-02+73

ISBN 978-985-543-034-7

© УО «Белорусский государственный университет информатики и радиоэлектроники», 2016

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

Батура Михаил Павлович	председатель , ректор Белорусского государственного университета информатики и радиоэлектроники
Голенков Владимир Васильевич	сопредседатель , заведующий кафедрой интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники
Живицкая Елена Николаевна	проректор по учебной работе и менеджменту качества Белорусского государственного университета информатики и радиоэлектроники
Кузнецов Александр Петрович	проректор по научной работе Белорусского государственного университета информатики и радиоэлектроники
Никольшин Борис Викторович	проректор по учебной работе и информатизации Белорусского государственного университета информатики и радиоэлектроники
Кулаженко Юрий Иванович	начальник Управления подготовки научных кадров высшей квалификации
Шилин Леонид Юрьевич	декан факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники
Титович Анна Францевна	директор центра международного сотрудничества Белорусского государственного университета информатики и радиоэлектроники
Боярко Алла Викторовна	руководитель прес-службы Белорусского государственного университета информатики и радиоэлектроники

РАБОЧАЯ ГРУППА ОРГКОМИТЕТА КОНФЕРЕНЦИИ

Гулякина Наталья Анатольевна	руководитель группы , заместитель заведующего кафедрой ИИТ Белорусского государственного университета информатики и радиоэлектроники
Гракова Наталья Викторовна	заместитель заведующего кафедрой ИИТ по воспитательным вопросам Белорусского государственного университета информатики и радиоэлектроники
Губаревич Анастасия Владимировна	инженер кафедры ИИТ Белорусского государственного университета информатики и радиоэлектроники
Лучина Татьяна Леонидовна	инженер кафедры ИИТ Белорусского государственного университета информатики и радиоэлектроники
Русецкий Кирилл Валерьевич	инженер-программист кафедры ИИТ Белорусского государственного университета информатики и радиоэлектроники

ORGANIZATIONAL COMMITTEE

Mikhail Batura	Chairman , Rector, Belarusian State University of Informatics and Radio Electronics
Vladimir Golenkov	Co-chairman , Head of Informational Intelligent Technologies Department, Belarusian State University of Informatics and Radio Electronics
Elena Zhivitskaya	Vice-Rector for Academic Work and Quality Management, Belarusian State University of Informatics and Radio Electronics
Alexander Kuznetsov	Vice-Rector for Research and Development, Belarusian State University of Informatics and Radio Electronics
Boris Nikulshin	Vice Rector for Education and Information, Belarusian State University of Informatics and Radio Electronics
Yury Kulazhenko	Postgraduate Office Head
Leonid Shilin	Dean of the Faculty of Information Technologies and Control, Belarusian State University of Informatics and Radio Electronics
Anna Titovich	Director of the Center for International Cooperation, Belarusian State University of Informatics and Radio Electronics
Alla Boyarko	Head of Press Service, Belarusian State University of Informatics and Radio Electronics

CONFERENCE ORGANIZATIONAL COMMITTEE WORKGROUP

Natalia Guliakina	Group leader assistant head of IIT Department at Belarusian State University of Informatics and Radioelectronics
Natalia Grakova	assistant head of IIT department at Belarusian State University of Informatics and Radioelectronics
Anastasia Hubarevich	engineer at IIT department of Belarusian State University of Informatics and Radioelectronics
Tatiana Lutchina	engineer at IIT department of Belarusian State University of Informatics and Radioelectronics
Kirill Rusetski	software engineer at IIT department of Belarusian State University of Informatics and Radioelectronics

ПРОГРАММНЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

д.т.н., проф.	Кузнецов Олег Петрович	Председатель , председатель Совета Российской ассоциации искусственного интеллекта, заведующий лабораторией, Институт проблем управления РАН, г. Москва, Российская Федерация
д.т.н., проф.	Голенков Владимир Васильевич	Сопредседатель , заведующий кафедрой интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь
к.т.н., доц.	Боргест Николай Михайлович	Профессор, Самарский государственный аэрокосмический университет имени Академика С.П.Королева, г. Самара, Российская Федерация
д.т.н., проф.	Борисов Аркадий Николаевич	Профессор, Рижский технический университет, Институт информационных технологий, г. Рига, Латвия
д.т.н., доц.	Валькман Юрий Роландович	Заведующий отделом распределенных интеллектуальных систем Международного научно-учебного центра информационных технологий и систем НАНУ и МОНУ, профессор кафедры математических методов системного анализа УНК «ИПСА» НТУУ «КПИ», г. Киев, Украина
д.ф.-м.н. академик РАН	Васильев Станислав Николаевич	Директор Института проблем управления им. В. А. Трапезникова РАН, председатель Ученого Совета, г. Москва, Российская Федерация
д.т.н., проф.	Гаврилова Татьяна Альбертовна	Заведующая кафедрой информационных технологий в менеджменте, Высшая Школа менеджмента СПбГУ, г. Санкт-Петербург, Российская Федерация
д.т.н., проф.	Глоба Лариса Сергеевна	Заведующая кафедрой, Национальный технический университет Украины «Киевский политехнический институт», г. Киев, Украина
д.т.н., проф.	Головко Владимир Адамович	Заведующий кафедрой интеллектуальных информационных технологий, Брестский государственный технический университет, г. Брест, Беларусь
д.фил.н., проф.	Гордей Александр Николаевич	Директор, Республиканский институт китаеведения имени Конфуция БГУ, г. Минск, Беларусь
д.т.н., с.н.с.	Грибова Валерия Викторовна	Заведующая лабораторией интеллектуальных систем, Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток, Российская Федерация
к.ф.-м.н., доц.	Гулякина Наталья Анатольевна	Заместитель заведующего кафедрой интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники г. Минск, Беларусь
д.т.н., проф.	Додонов Александр Георгиевич	Заместитель директора по научной работе, Институт проблем регистрации информации НАН Украины, г.Киев, Украина
д.т.н., проф.	Еремеев Александр Павлович	Заведующий кафедрой прикладной математики, Национальный исследовательский университет «МЭИ», г. Москва, Российская Федерация
к.фил.н.	Ефименко Ирина Владимировна	Заместитель декана факультета филологии, НИУ ВШЭ; Ведущий научный сотрудник, ЦИАС ИСИЭЗ НИУ ВШЭ г. Москва, Российская Федерация
д.ф.-м.н., проф.	Журавков Михаил Анатольевич	Министр образования, Министерство образования Республики Беларусь
д.т.н., проф.	Заболеева-Зотова Алла Викторовна	Начальник Управления региональных и межгосударственных программ РФФИ, г. Москва, Российская Федерация
к.т.н., с.н.с.	Загоруйко Юрий Алексеевич	Заведующий лабораторией, Институт систем информатики им. А.П. Ершова СО РАН, г. Новосибирск, Российская Федерация

к.т.н., доц.	Карминская Татьяна Дмитриевна	Ректор Государственного образовательного учреждения высшего профессионального образования «Югорский государственный университет»
д.ф.-м.н., проф.	Клещев Александр Сергеевич	Главный научный сотрудник, Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток, Российская Федерация
д.мед.н., проф.	Кобринский Борис Аркадьевич	Главный научный сотрудник, Институт системного анализа Федерального исследовательского центра «Информатика и управление» РАН, профессор, Российский национальный исследовательский медицинский университет им. Н.И. Пирогова Минздрава России, г.Москва, Российская Федерация
д.п.н., проф.	Козлов Олег Александрович	Заместитель директора института, Учреждение Российской Академии Образования «Институт Информатизации Образования», г. Москва, Российская Федерация
д.т.н., проф.	Комарцова Людмила Георгиевна	Профессор кафедры «Компьютерные системы и сети», Московский Государственный технический университет им. Н.Э. Баумана (Калужский филиал), г. Калуга, Российская Федерация
д.т.н., проф.	Курейчик Виктор Михайлович	Заместитель руководителя по научной и инновационной деятельности, Таганрогский кампус ЮФУ, г. Таганрог, Российская Федерация
д.т.н., с.н.с	Ландэ Дмитрий Владимирович	Заведующий отделом, Институт проблем регистрации информации НАН Украины, г. Киев, Украина
д.т.н., с.н.с.	Лобанов Борис Мефодьевич	Главный научный сотрудник, Объединённый институт проблем информатики НАН Беларуси, г. Минск, Беларусь
д.т.н.	Лукашевич Наталья Валентиновна	Ведущий научный сотрудник, Московский государственный университет имени М.В.Ломоносова, г. Москва, Российская Федерация
д.т.н., проф.	Массель Людмила Васильевна	Главный научный сотрудник, заведующая лабораторией, Институт систем энергетики им. Л.А. Мелентьева СО РАН, г. Иркутск, Российская Федерация
к.т.н., доц.	Незорова Ольга Авенировна	Заместитель директора по научной работе, Научно-исследовательский институт «Прикладная семиотика» АН РТ, г. Казань, Российская Федерация
д.ф.-м.н., проф.	Осипов Геннадий Семенович	Президент Российской ассоциации искусственного интеллекта, заместитель заведующего кафедрой «Математические методы системного анализа», Национальный исследовательский университет «Высшая школа экономики», г. Москва, Российская Федерация
д.т.н., проф.	Палюх Борис Васильевич	Заведующий кафедрой Информационных систем, Тверской государственный технический университет, г. Тверь, Российская Федерация
д.т.н., проф.	Петровский Александр Александрович	Заведующий кафедрой ЭВС, Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь
д.т.н., проф.	Петровский Алексей Борисович	Заведующий лабораторией, ФГБУН Институт системного анализа Российской академии наук, г. Москва, Российская Федерация
к.ф.-м.н., проф.	Плесневич Геральд Станиславович	Профессор кафедры «Информационные технологии», Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «МАТИ – Российский государственный технологический университет имени К.Э. Циолковского», г. Москва, Российская Федерация
академик РАО, д.пед.н., проф.	Роберт Ирэна Веняминовна	Директор ФГБНУ «Институт информатизации образования Российской академии образования», г. Москва, Российская Федерация

д.т.н., проф.	Рыбина Галина Валентиновна	Руководитель направления «Интеллектуальные системы и технологии» и учебно-научной лаборатории «Системы искусственного интеллекта» кафедры Кибернетики, Национальный исследовательский ядерный университет МИФИ, г. Москва, Российская Федерация
д.т.н., проф.	Сидоркина Ирина Геннадьевна	Декан факультета информатики и вычислительной техники, Поволжский государственный технологический университет, г. Йошкар-Ола, Республика Марий Эл, Российская Федерация
д.т.н.	Смирнов Сергей Викторович	Директор, ведущий научный сотрудник, заведующий лабораторией анализа и моделирования сложных систем, Институт проблем управления сложными системами Российской академии наук, г. Самара, Российская Федерация
д.ф.-м.н., проф.	Соловьёв Сергей Юрьевич	Профессор, Московский государственный университет имени М.В.Ломоносова, г. Москва, Российская Федерация
д.т.н., проф.	Соснин Петр Иванович	Заведующий кафедрой «Вычислительная техника», Ульяновский государственный технический университет, г. Ульяновск, Российская Федерация
д.т.н., проф.	Стефанюк Вадим Львович	Главный научный сотрудник, Институт проблем передачи информации РАН, г. Москва, Российская Федерация
д.т.н., Академик АН РТ, проф.	Сулейманов Джавдет Шевкетович	Заведующий кафедрой, Казанский (Приволжский) федеральный университет, г. Казань, Российская Федерация
к.т.н., доц.	Тарасов Валерий Борисович	Доцент кафедры компьютерных систем автоматизации производства, Московский государственный технический университет им. Н.Э.Баумана, г. Москва, Российская Федерация
д.э.н., проф.	Тельнов Юрий Филиппович	Заведующий кафедрой, Московский государственный университет экономики, статистики и информатики, г. Москва, Российская Федерация
д.ф.-м.н., проф.	Тузиков Александр Васильевич	Генеральный директор, Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь
д.т.н.	Харламов Александр Александрович	Старший научный сотрудник, Институт высшей нервной деятельности и нейрофизиологии РАН, г. Москва, Российская Федерация
д.т.н., проф.	Хорошевский Владимир Федорович	Заведующий сектором, ВЦ им. А.А. Дородницына РАН Главный научный сотрудник, ЦИАС ИСИЭЗ НИУ ВШЭ, г. Москва, Российская Федерация
Академик НАН Беларуси, д.т.н., проф.	Чернявский Александр Фёдорович	Заведующий кафедрой интеллектуальных систем, Белорусский государственный университет, г. Минск, Беларусь
д.т.н., проф.	Шарипбай Алтынбек Амирович	Директор научного центра «Искусственный интеллект», Евразийский национальный университет имени Л.Н.Гумилева, г. Астана, Казахстан
д.т.н. проф.	Ярушкина Надежда Глебовна	Проректор по научной работе, Ульяновский государственный технический университет, г. Ульяновск, Российская Федерация

PROGRAM COMMITTEE

Borgest N.	c. of t.s., RF ass. Proff.		proff.
Borisov A.	d. of t.s., Latvia proff.	Palyukh B.	d. of t.s., RF proff.
Chernyavsky A.	NAS Belarus academician	Petrovsky A.	d. of t.s., RB proff.
Dodonov A.G.	d. of t.s., Ukraine proff.	Petrovsky A.	c. of t.s., d. of t.s., RF proff.
Efimenko I.	c. of phyl.s., RF	Plesniewicz G.	c. of ph.-m.s., RF
Eremeev A.	d. of t.s., RF proff.	Robert I.	Academician of REA, d. of teach.s., RF proff.
Gavrilova T.	d. of t.s., RF proff.	Rybina G.V.	d. of t.s., RF proff.
Globa L.	d. of t.s., Ukraine proff.	Sharipbay A.	d. of t.s., Kazakhstan proff.
Golenkov V.	d. of t.s., RB proff.	Sidorkina I.	d. of t.s., RF proff.
Golovko V.	d. of t.s., RB proff.	Smirnov S.	d. of t.s., RF proff.
Gribova V.	d. of t.s., RF	Soloviev S.	d. of ph.-m.s., RF proff.
Guliakina N.	c. of ph.-m.s., RB ass. Proff.	Sosnin P.	d. of t.s., RF proff.
Hardzei A.	d. of phyl.s., RB proff.	Stefanuk V.	d. of t.s., RF proff.
Karminskaya T.	c. of t.s., RF	Suleymanov D.	AS Tatarstan academician, KF
Kharlamov A.	d. of t.s., RF	Tarasov V.	c. of t.s., RF ass. Proff.
Khoroshevsky V.	d. of t.s., RF proff.	Telnov Yu.	d. of e.s., RF proff.
Kleshev A.	d. of t.s., RF proff.	Tuzikov A.	d. of ph.-m.s., RB proff.
Kobrinskiy B.	d. of med.s., RF	Vasilyev S.	academician of RAS, d. of ph.-m.s., RB proff.
Komartsova L.	d. of t.s., RF	Walkman Yu.R.	d. of t.s., Ukraine proff.
Kozlov O.	d. of teach.s., RF proff.	Yarushkina N.G.	d. of t.s., RF proff.
Kureychik V.	d. of t.s., RF proff.	Zaboleeva-Zotova A.	d. of t.s., RF
Kuznetsov O.	d. of t.s., RF proff.	Zagorulko Yu.	c. of t.s., RF ass. Proff.
Lande D.	d. of t.s., Ukraine	Zhuravkov M.A.	d. of ph.-m.s., RB Proff
Lobanov B.	d. of t.s., RB proff.		
Loukachevitch N.	d. of t.s., RF		
Massel L.	d. of t.s., RF proff.		
Nevzorova O.	c. of t.s., RF ass. Proff.		
Osipov G.	d. of ph.-m.s., RF		

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	23
СИНТЕЗ СЕРВИСНОГО И АРХИТЕКТУРНОГО ПОДХОДОВ К ПРОЕКТИРОВАНИЮ СИСТЕМ УПРАВЛЕНИЯ ЗНАНИЯМИ Кудрявцев Д. В., Гаврилова Т. А.	27
КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ УПРАВЛЯЕМОЙ СИСТЕМЫ ПОДДЕРЖКИ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ Клещев А.С., Шалфеева Е.А.	31
ТЕХНОЛОГИЯ РАЗРАБОТКИ РЕШАТЕЛЕЙ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНЫХ СЕРВИСОВ ДЛЯ ОБЛАЧНОЙ ПЛАТФОРМЫ IASRAAS НА ОСНОВЕ РАСШИРЯЕМОГО РЕДАКТОРА ОРГРАФОВ ИНФОРМАЦИИ Москаленко Ф.М., Тимченко В.А.	39
КОНЦЕПЦИЯ ИНСТРУМЕНТАЛЬНОГО КОМПЛЕКСА ДЛЯ ПОСТРОЕНИЯ СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИОННЫХ СИСТЕМ Грибова В.В., Иванова А.В.	45
ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЙ ДЛЯ СОЗДАНИЯ СЦЕНАРИЕВ ОБРАБОТКИ ДАННЫХ Глоба Л.С.	51
МЕТОД ФОРМИРОВАНИЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ НА ПОРТАЛАХ ЗНАНИЙ Глоба Л.С., Новогрудская Р.Л.	57
ПОДХОД К ОРГАНИЗАЦИИ КОМПЛЕКСНОЙ ПОДДЕРЖКИ ПРОЦЕССА РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СППР В СЛАБОФОРМАЛИЗОВАННЫХ ПРЕДМЕТНЫХ ОБЛАСТЯХ Загорулько Г.Б., Загорулько Ю.А.	61
РАЗРАБОТКА ЯЗЫКА СИТУАЦИОННОГО УПРАВЛЕНИЯ В ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЕ СЕМИОТИЧЕСКОГО ТИПА НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО ИНЖИНИРИНГА Массель Л.В., Массель А.Г.	65
ИНТЕГРАЦИЯ СЕМИОТИКИ, КОГНИТИВНОЙ ГРАФИКИ И СЕМАНТИЧЕСКОГО МОДЕЛИРОВАНИЯ В ИНТЕЛЛЕКТУАЛЬНЫХ СЕМИОТИЧЕСКИХ СИСТЕМАХ СИТУАЦИОННОГО УПРАВЛЕНИЯ Массель Л.В., Массель А.Г.	71
ЕДИНАЯ ОНТОЛОГИЧЕСКАЯ ПЛАТФОРМА ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ Филиппов А.А., Мошкин В.С., Шалаев Д.О., Ярушкина Н.Г.	77
АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ОНТОЛОГИЧЕСКОГО АНАЛИЗА ДАННЫХ Зубцов Р.О., Семенова В.А., Смирнов С.В.	83
ФОРМИРОВАНИЕ МЕДИЦИНСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ Кобринский Б.А.	89
СРЕДСТВА СТРУКТУРИЗАЦИИ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ БАЗ ЗНАНИЙ Давыденко И.Т., Гракова Н.В., Сергиенко Е.С., Федотова А.В.	93

СЕМАНТИЧЕСКАЯ МОДЕЛЬ КОЛЛЕКТИВНОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ Давыденко И.Т.	107
СЕМАНТИЧЕСКАЯ КЛАССИФИКАЦИЯ ДЕЙСТВИЙ ДЛЯ ВЫВОДА ЗНАНИЙ Бойко И.М.	115
ФОРМАЛЬНОЕ СЕМАНТИЧЕСКОЕ ОПИСАНИЕ ЦЕЛЕНАПРАВЛЕННОЙ ДЕЯТЕЛЬНОСТИ РАЗЛИЧНОГО ВИДА СУБЪЕКТОВ Шункевич Д.В., Губаревич А.В., Святкина М.Н., Моросин О.Л.	125
ВЗАИМОДЕЙСТВИЕ АСИНХРОННЫХ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ ОБРАБОТКИ ЗНАНИЙ В ОБЩЕЙ СЕМАНТИЧЕСКОЙ ПАМЯТИ Шункевич Д.В.	137
ПРИНЦИПЫ ПЛАТФОРМЕННОЙ НЕЗАВИСИМОСТИ И ПЛАТФОРМЕННОЙ РЕАЛИЗАЦИИ OSTIS Ивашенко В.П., Татур М.М.	145
ПРИНЦИПЫ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ СХРАНИЛИЩА ПРОЕКТА OSTIS И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ПЛАТФОРМЕ .NET FRAMEWORK Каешко А.И.	151
«УМНЫЙ ДОМ», УПРАВЛЯЕМЫЙ ЗНАНИЯМИ Корончик Д. Н.	155
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АВТОМАТИЧЕСКОГО РЕШЕНИЯ ЗАДАЧ ПО ПЛАНИМЕТРИИ Курбатов С.С., Лобзин А.П., Хахалин Г.К.	159
ОБ ОНТОЛОГИИ ИНТЕЛЛЕКТУАЛЬНОГО МОДЕЛИРОВАНИЯ Валькман Ю.Р., Степашко П.В.	165
МОДЕЛИ ПРЕДМЕТНЫХ ОБЛАСТЕЙ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ МОНИТОРИНГА ИНФОРМАЦИОННОГО ПРОСТРАНСТВА Додонов А.Г., Ландэ Д.В., Коваленко Т.В.	171
ИСПОЛЬЗОВАНИЕ ОДНОРОДНОЙ СЕМАНТИЧЕСКОЙ СЕТИ ДЛЯ КЛАССИФИКАЦИИ РЕЗУЛЬТАТОВ ГЕНЕТИЧЕСКОГО АНАЛИЗА Куликов А.М., Харламов А.А.	177
КОНСТРУИРОВАНИЕ ОТНОШЕНИЯ ПСЕВДОХРОНОЛОГИЧЕСКОГО ПОРЯДКА В ТЕРМИНОЛОГИЧЕСКИХ СЕТЯХ Мальковский М.Г., Соловьев С.Ю.	183
АЛГОРИТМ ВЫДЕЛЕНИЯ БИНАРНЫХ ОТНОШЕНИЙ В ТЕРМИНОЛОГИЧЕСКОЙ СЕТИ Тузикова А.В.	187
АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ ДЕРЕВА КЛАССОВ ОНТОЛОГИИ И ИХ ЭКЗЕМПЛЯРОВ НА ОСНОВЕ ТЕКСТА ПО ПРЕДМЕТНОЙ ОБЛАСТИ Андреев И.А., Бексаева Е.А., Желепов А.С., Клейн В.В., Серков И.П.	191
ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА ЗНАНИЙ ХРАНЯЩИХСЯ В МНОГОМЕРНЫХ ДАННЫХ ПОСРЕДСТВОМ РАСШИРЕННОГО БАЗОВОГО СЕМАНТИЧЕСКОГО ГИПЕРГРАФА Шарипбай А.А., Барлыбаев А.Б., Сабыров Т.С.	197

ПОДХОД К АВТОМАТИЗАЦИИ СОЗДАНИЯ БАЗ ЗНАНИЙ НА ОСНОВЕ ТРАНСФОРМАЦИИ КОНЦЕПТУАЛЬНЫХ МОДЕЛЕЙ	203
Дородных Н.О., Юрин А.Ю.	
СИСТЕМА ПРОГРАММИРОВАНИЯ ПРОДУКЦИОННЫХ БАЗ ЗНАНИЙ: PERSONAL KNOWLEDGE BASE DESIGNER	209
Грищенко М.А., Дородных Н.О., Юрин А.Ю.	
ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ СТРУКТУРНОГО СИНТЕЗА	213
Бикмуллина И.И.	
О ДВУХ КЛАССАХ СЕМАНТИЧЕСКОЙ КОРРЕЛЯЦИИ, НАБЛЮДАЕМЫХ В ФУНКЦИОНАЛЬНЫХ СЕМАНТИЧЕСКИХ СЕТЯХ	219
Савкин Л.В.	
РОССИЯ В КОНТЕКСТЕ МИРОВЫХ ЦЕНТРОВ КОМПЕТЕНЦИЙ И ПРЕВОСХОДСТВА	223
Хорошевский В.Ф., Ефименко И.В.	
ПОСТРОЕНИЕ СЕТЕЙ СОАВТОРСТВА ПО ДАННЫМ СЕРВИСА GOOGLE SCHOLAR CITATIONS	233
Ландэ Д.В., Балагура И.В., Андрущенко В.Б.	
РАЗРАБОТКА ОБЪЕКТИВНЫХ МЕТОДОВ ОЦЕНКИ КОМПЕТЕНТНОСТИ УЧАСТНИКОВ НАУЧНЫХ ПРОЕКТОВ НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО АНАЛИЗА	239
Рогущина Ю.В., Гладун А.Я.	
АНАЛИЗ ПРОЦЕССОВ ИНФОРМАЦИОННОГО ОБМЕНА В НАУКОМЕТРИЧЕСКИХ БАЗАХ ДАННЫХ	247
Потебня А.В., Погорелый С.Д.	
РАСПОЗНАВАНИЕ ИНФОРМАЦИОННЫХ ОПЕРАЦИЙ: МУЛЬТИАГЕНТНЫЙ ПОДХОД	253
Додонов А.Г., Ландэ Д.В., В.А. Додонов	
МОДЕЛЬ КООПЕРАЦИИ АГЕНТОВ В СЕМИОТИЧЕСКОЙ СРЕДЕ	259
Кулинич А.А.	
МОДИФИЦИРОВАННЫЙ МЕТОД ЛОГИЧЕСКОГО ВЫВОДА ЗНАНИЙ НА ОСНОВЕ НЕЧЕТКОЙ ОНТОЛОГИИ И БАЗЫ ПРЕЦЕДЕНТОВ	265
Мошкин В.С., Ярушкина Н.Г.	
АЛГОРИТМ ПРЕДОБРАБОТКИ И ВОССТАНОВЛЕНИЯ АНКЕТНЫХ ДАННЫХ	271
Сибирев И.В., Афанасьева Т.В.	
О ФОРМАЛИЗАЦИИ ТЕМПОРАЛЬНЫХ РАССУЖДЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ	275
Вагин В.Н., Еремеев А.П., Гулякина Н.А.	
РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ СИСТЕМ НА ОСНОВЕ ТЕМПОРАЛЬНЫХ СЕТЕЙ ПЕТРИ	283
Еремеев А.П., Королев Ю.И.	
РАЗРАБОТКА МЕТОДОВ РАССУЖДЕНИЙ НА ОСНОВЕ ТЕМПОРАЛЬНЫХ ПРЕЦЕДЕНТОВ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	289
Куриленко И.Е.	

СПЕЦИФИКАЦИЯ ПОТОКА ЗАДАЧ В РАСШИРЕННОЙ ИНТЕРВАЛЬНОЙ ЛОГИКЕ АЛЛЕНА	295
Плесневич Г.С., Нгуен Тхи Минь Ву	
АЛГОРИТМ УНИФИКАЦИИ ВРЕМЕННЫХ РЯДОВ	301
Сапунков А.А., Афанасьева Т.В.	
ОПИСАНИЕ ПОДХОДА К ИЗВЛЕЧЕНИЮ ЗНАНИЙ ОБ АНОМАЛИЯХ ЛОКАЛЬНЫХ ТЕНДЕНЦИЙ ВРЕМЕННОГО РЯДА	305
Афанасьева Т.В., Заварзин Д.В.	
КОГНИТИВНЫЙ ФРАКТАЛЬНЫЙ АНАЛИЗ ВРЕМЕННЫХ РЯДОВ: R/S АНАЛИЗ И V-СТАТИСТИКА	309
Романко О.Р.	
СЕМАНТИЧЕСКОЕ КОДИРОВАНИЕ НА ОСНОВЕ ГЛУБОКИХ АВТОАССОЦИАТИВНЫХ НЕЙРОННЫХ СЕТЕЙ	313
Головкин В.А., Крощенко А.А., Хацкевич М.В.	
НЕЙРОННЫЕ СЕТИ В СЕМАНТИЧЕСКОМ АНАЛИЗЕ	319
Аверкин А.Н., Ярушев С.А.	
НОВЫЙ ПОДХОД К МОДЕЛИРОВАНИЮ ФОРМАЛЬНЫХ НЕЙРОНОВ И ИХ ВЗАИМОДЕЙСТВИЙ	323
Жилиякова Л.Ю.	
РЕАЛИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВИРТУАЛЬНЫХ СРЕД ПО ДЕКЛАРАТИВНОЙ МОДЕЛИ	327
Грибова В.В., Федорищев Л.А.	
ЛОГИКО-АЛГЕБРАИЧЕСКИЕ МЕТОДЫ ПОСТРОЕНИЯ КОГНИТИВНЫХ СЕНСОРОВ	331
Святкина М.Н., Тарасов В.Б.	
ИЛЛЮЗИИ ВОСПРИЯТИЯ И ПРЕДСТАВЛЕНИЯ: ИНФОРМАЦИОННЫЙ ПОДХОД	349
Фоминых И.Б.	
ОНТОЛОГИЧЕСКИЙ ПОДХОД К ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ	355
Коршунов С.А.	
КОМПЛЕКСИРОВАНИЕ ТЕКСТОВЫХ ОПИСАНИЙ И СЕМАНТИЗИРОВАННОЙ ГРАФИКИ В РЕШЕНИИ ПРОЕКТНЫХ ЗАДАЧ	359
Соснин П.И., Галочкин М.В., Лунецкас А.А.	
КОМПЛЕКС ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ, РЕАЛИЗУЮЩИХ ЛОГИКУ ОЦЕНКИ ПСИХОФИЗИОЛОГИЧЕСКОГО СОСТОЯНИЯ ЧЕЛОВЕКА	367
Порядин А.Е., Сидоркина И.Г.	
СЕМАНТИЧЕСКИЕ АСПЕКТЫ ПРЕДСТАВЛЕНИЯ МЕТАДАННЫХ В СИСТЕМЕ «КОРПУС-МЕНЕДЖЕР»	371
Невзорова О.А., Мухамедшин Д.Р., Курманбакиев М.И.	
СОЗДАНИЕ ЛЕКСИКОНА ОЦЕНОЧНЫХ СЛОВ РУССКОГО ЯЗЫКА РУСЕНТИЛЕКС	377
Лукашевич Н.В., Левчик А.В.	
МИКРОСЕРВИСНЫЙ ПОДХОД К ПРОЕКТИРОВАНИЮ РЕЧЕВЫХ ИНТЕРФЕЙСОВ	383
Житко В.А., Лобанов Б.М.	

КОНТЕКСТНЫЕ ПРАВИЛА ДЛЯ РАЗРЕШЕНИЯ МОРФОЛОГИЧЕСКОЙ МНОГОЗНАЧНОСТИ В КОРПУСЕ ТАТАРСКОГО ЯЗЫКА	389
Гильмуллин Р.А., Гатауллин Р.Р.	
СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР КАЗАХСКОГО ЯЗЫКА НА ОСНОВЕ ГРАММАТИКИ СВЯЗЕЙ – LINKGRAMMARPARSER	393
Бегимтай У.Х.	
АЛГАРЫТМ І ЛІНГВІСТЫЧНЫЯ РЭСУРСЫ ДЛЯ НАРМАЛІЗАЦЫІ ТЭКСТАЎ ГЕАГРАФІЧНАГА ДАМЕНА	397
Гецэвіч Ю.С., Качан Я.С., Лысы С.І., Маракуліна П.А., Крывальцэвіч А.В.	
СЕМАНТИЧЕСКИЙ АНАЛИЗ ТЕКСТА ДЛЯ РАСПОЗНАВАНИЯ ЭЛЕМЕНТОВ ВНЕШНЕГО ВИДА ЧЕЛОВЕКА	401
Долбин А.В., Розалиев В.Л., Орлова Ю.А., Заболеева-Зотова А.В.	
О ФИЗИЧЕСКОЙ СТРУКТУРЕ ПРОСТЫХ ГЛАСНЫХ ЗВУКОВ РЕЧИ ЧЕЛОВЕКА	405
Митянок В.В.	
МЕТОДИКА ГОЛОСОВОЙ ИДЕНТИФИКАЦИИ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ	411
Меньшаков П.А., Мурашко И.А.	
ОНТОЛОГИЧЕСКИЙ ПОДХОД К ФОРМИРОВАНИЮ КОНТЕКСТНЫХ ЗАПРОСОВ В ЭЛЕКТРОННОМ АРХИВЕ ТЕХНИЧЕСКИХ ДОКУМЕНТОВ	415
Наместников А.М.	
СЕМАНТИЧЕСКИЙ АНАЛИЗ ТЕКСТОВОЙ ЧАСТИ РУКОВОДСТВА ПОЛЬЗОВАТЕЛЯ ПО СБОРКЕ ДВИГАТЕЛЯ С ПОМОЩЬЮ ОНТОРЕДАКТОРА FLUENTEDITOR	421
Орлова А.А., Боргест Н.М.	
СВЯЗЬ МОДУЛЬНОГО ИНТЕРФЕЙСА ИНТЕЛЛЕКТУАЛЬНОГО ПОМОЩНИКА ПРОЕКТАНТА С САЕ СИСТЕМОЙ ANSYS	427
Власов С.А., Коровин М.Д.	
АЛГОРИТМЫ ОБРАБОТКИ ИНФОРМАЦИИ В АДАПТИВНОМ РЕКОНФИГУРИРУЕМОМ МОДУЛЕ САПР ОК ДЛЯ ВИЗУАЛИЗАЦИИ КОНТУРОВ ТЕПЛОВЫХ ПОТОКОВ	431
Сорокин О.Л. Сидоркина И.Г.	
ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДЕЛЕЙ ЗРЕЛЫХ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ В КОНТЕКСТЕ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ПОТОКАМИ КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКИХ РАБОТ	435
Ларин С.Н.	
РАЗРАБОТКА СПРАВОЧНИКОВ ОБЪЕКТОВ ОПЕРАТИВНО-ПРОИЗВОДСТВЕННОГО ПЛАНИРОВАНИЯ НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО ПОДХОДА	441
Карпаев С.А., Ларин С.Н.	
АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УПРАВЛЕНИЯ ЭВОЛЮЦИЕЙ НЕЧЕТКОЙ МНОГОСТАДИЙНОЙ ТЕХНОЛОГИЧЕСКОЙ СИСТЕМЫ	445
Палюх Б.В., Егерев И.А.	
МОДЕЛИ ИНТЕЛЛЕКТУАЛЬНОЙ АСУТП ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ	449
Бурдо Г.Б., Федотова А.В.	

САМООРГАНИЗУЮЩИЕСЯ СЕТЕВЫЕ ЭКСПЕРТНЫЕ СРЕДЫ В ЗАДАЧАХ РАЗВИТИЯ НАУКОЕМКИХ ПРОИЗВОДСТВ	457
Виноградов Г.П., Виноградова Н.Г.	
НЕЧЕТКИЕ МОДЕЛИ ПОВЕДЕНИЯ ПРЕДПРИНИМАТЕЛЯ ПРИ ПЕРЕГОВОРАХ ОБ ИНВЕСТИРОВАНИИ В УСЛОВИЯХ ТОЧНОГО ЗНАНИЯ ХАРАКТЕРА ПОТЕНЦИАЛЬНОГО ИНВЕСТОРА	465
Палюх Б.В., Ключин А.Ю., Мутовкина Н.Ю.	
СЕНСОРНАЯ МОДЕЛЬ ПОДРАЖАТЕЛЬНОГО ПОВЕДЕНИЯ РОБОТОВ	471
Карпов В.Э.	
ОНТОЛОГИЧЕСКИЙ ПОДХОД К ПОСТРОЕНИЮ ИНТЕРАКТИВНОЙ ВИРТУАЛЬНОЙ СРЕДЫ ДЛЯ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ ПЛАНИРУЕМЫХ ДЕЙСТВИЙ ПРИ ДИАЛОГОВОМ УПРАВЛЕНИИ РОБОТОМ-ПОМОЩНИКОМ КОСМОНАВТА НА МКС	477
Крючков Б.И., Усов В.М., Карпов А.А.	
ПРИМЕНЕНИЕ ДАННЫХ ТЕЛЕМЕТРИИ ДЛЯ ЦЕЛЕЙ УПРАВЛЕНИЯ НЕШТАТНЫМИ СИТУАЦИЯМИ НА БОРТУ СПУТНИКОВЫХ СИСТЕМ	483
Данилин Н.С., Димитров Д.М., Сабилов И.Х.	
ОСОБЕННОСТИ РЕАЛИЗАЦИИ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ СОСТОЯНИЯ КОСМИЧЕСКИХ АППАРАТОВ	487
Юрыгина Ю.С., Лахин О.И., Донсков А.В., Мишурова Н.В., Сысоев Д.В., Скорюпина Е.Г.	
ОСОБЕННОСТИ ПОСТРОЕНИЯ ОНТОЛОГИИ ДЛЯ АДАПТИВНОГО ПЛАНИРОВАНИЯ ПРОГРАММЫ ПОЛЕТА И ГРУЗОПОТОКА РОССИЙСКОГО СЕГМЕНТА МЕЖДУНАРОДНОЙ КОСМИЧЕСКОЙ СТАНЦИИ	493
Лахин О.И., Полников А.С., Редькина К.В.	
НЕКОТОРЫЕ ОСОБЕННОСТИ РАЗРАБОТКИ И ПРИМЕНЕНИЯ ОБУЧАЮЩИХ ИНТЕГРИРОВАННЫХ ЭКСПЕРТНЫХ СИСТЕМ В УЧЕБНОМ ПРОЦЕССЕ НИЯУ МИФИ	497
Рыбина Г.В., Рыбин В.М., Сергиенко Е.С.	
МЕТОДЫ И МОДЕЛИ СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ ОБРАЗОВАТЕЛЬНОГО НАЗНАЧЕНИЯ	503
Шихнабиева Т.Ш.	
О СОЗДАНИИ АДАПТИВНОЙ СИСТЕМЫ ГЕНЕРАЦИИ МУЛЬТИМОДАЛЬНЫХ ОБУЧАЮЩИХ МАТЕРИАЛОВ	507
Хусаинов А.Ф., Русецкий К.В.	
РАЗРАБОТКА МОДЕЛИ УЧЕТА ВЛИЯНИЯ ЭМОЦИЙ НА ФОРМИРОВАНИЕ ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ ОБУЧЕНИЯ СТУДЕНТА НА ОСНОВЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ	511
Козлов О.А., Михайлов Ю.Ф.	
ОЦЕНКА ЭФФЕКТИВНОСТИ ПРОЦЕССА ОБУЧЕНИЯ САМ-ОБРАБОТКЕ СТУДЕНТОВ МАШИНОСТРОИТЕЛЬНОГО ПРОФИЛЯ	517
Барабанова Е.Ю.	
СЕМАНТИЧЕСКАЯ МОДЕЛЬ ПОЛЬЗОВАТЕЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СПРАВОЧНОЙ СИСТЕМЫ ПО ПРОЕКТИРОВАНИЮ ПРОГРАММ	521
Пивоварчик О.В.	

ЭКСПЕРТНЫЕ МЕТОДЫ ОЦЕНКИ, КАЧЕСТВА И НАДЕЖНОСТИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНЫМ ПРОЦЕССОМ	527
Шарипбай А.А., Ниязова Р.С., Кузенбаев Б.А.	
ПРОБЛЕМЫ СЕМАНТИЧЕСКОЙ ОРГАНИЗАЦИИ ИНФОРМАЦИОННО- ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ И ПОДХОДЫ К ИХ РЕШЕНИЮ	531
Горбачёв Н.Н.	
ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ ПОТРЕБНОСТИ В КАДРАХ НА ОСНОВЕ СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	535
Ровба Е.А., Бойко В.К., Войтукевич Ю.А., Лявшук В.Е., Петров С.В.	
АГЕНТНЫЙ ПОДХОД К УПРАВЛЕНИЮ БЕЗОПАСНОСТЬЮ ПРИРОДНО- ТЕХНОГЕННОЙ СИСТЕМЫ	539
Богатиков В.Н., Виноградов Г.П., Воронин Ю.А.	
КОНЦЕПЦИЯ ИНСТРУМЕНТАЛЬНОЙ ПЛАТФОРМЫ МАРКЕТИНГОВОЙ ДЕЯТЕЛЬНОСТИ В СРЕДЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ	545
Вишняков В.А., Казак Е.А.	
СЕМАНТИЧЕСКИЙ АНАЛИЗ СОЦИАЛЬНЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ WOLFRAM MATHEMATICA	549
Тякунов А.С., Славский В.В.	
ИСПОЛЬЗОВАНИЕ СОЦИАЛЬНЫХ СЕТЕЙ ДЛЯ ПОИСКА КОМПЕТЕНТНЫХ ГРУПП ЛЮДЕЙ	553
Стратнев П.Ю.	
ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ ПОДДЕРЖКИ ЗАПРОСОВ НАЛОГОВЫХ ИНФОРМАЦИЙ	557
Турмаганбетова Ш.К.	
ГЕНЕРАЦИЯ ЗВУКОВ ПО ЦВЕТОВОЙ ГАММЕ ИЗОБРАЖЕНИЯ	561
Никитин Н.А., Розалиев В.Л., Орлова Ю.А., Заболеева-Зотова А.В.	
ПРИНЦИП ДОСТОВЕРНОЙ ЭКВИВАЛЕНТНОСТИ В ЗАДАЧАХ АДАПТИВНОГО УПРАВЛЕНИЯ В УСЛОВИЯХ АПРИОРНОЙ НЕОПРЕДЕЛЕННОСТИ	565
Бабамухамедова М.З., Дошанова М.Ю., Эргашев Ф.А.	
МЕТОДИКА И ПРОГРАММНОЕ СРЕДСТВО АНАЛИЗА СЛОЖНЫХ ДИНАМИЧЕСКИХ СИСТЕМ	569
Бурак Т.И., Лукашевич М.М.	
КАЧЕСТВЕННОЕ МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКИХ СИСТЕМ НА ОСНОВЕ МЕТОДОВ РАСПРОСТРАНЕНИЯ ОГРАНИЧЕНИЙ	573
Зуенко А.А.	
ПРИМЕНЕНИЕ СЕТЕЙ БАЙЕСА ДЛЯ АНАЛИЗА ЭКГ	579
Саган В.Ю., Фридман Р.	
ПОСТРОЕНИЕ БАЗЫ ЗНАНИЙ ДЛЯ ХРАНЕНИЯ И АНАЛИЗА БИМЕДИЦИНСКИХ КТ ИЗОБРАЖЕНИЙ ПОЯСНИЧНОГО ОТДЕЛА ПОЗВОНОЧНИКА	583
Куручка К.С., Карабчикова Е.А.	
АВТОРСКИЙ УКАЗАТЕЛЬ	587

TABLE OF CONTENTS

FOREWORD	23
SYNTHESIS OF SERVICE-BASED AND ARCHITECTURAL APPROACHES TO KNOWLEDGE MANAGEMENT SYSTEM DESIGN Kudryavtsev D.V., Gavrilova T.A.	27
CONCEPTUAL DESIGN OF MAINTAINABLE SYSTEM OF INTELLIGENT ACTIVITY AUTOMATION Kleschev A. , Shalfeeva E.	31
A TECHNOLOGY FOR DEVELOPMENT OF PROBLEM SOLVERS OF INTELLIGENT SYSTEMS FOR IACPAAS CLOUD PLATFORM ON THE BASIS OF EXTENSIBLE EDITOR FOR DIGRAPHS OF INFORMATION Moskalenko Ph.M., Timchenko V.A.	39
CONCEPT OF A TOOL FOR CREATING OF INFORMATION SECURITY SYSTEMS Gribova V.V., Ivanova A.V.	45
ONTOLOGIES USING FOR DATA PROCESSING SCENARIOS DESIGNING Globa L.S.	51
THE METHOD OF COMPLEX ENGINEERING CALCULATION OF KNOWLEDGE PORTALS COMPOSITION Globa L.S., Novogradskaya R.L.	57
AN APPROACH TO ORGANIZATION OF INTEGRATED SUPPORT OF THE DEVELOPMENT OF INTELLIGENT DSS IN WEAKLY FORMALIZED DOMAINS Zagorulko G.B., Zagorulko Yu.A.	61
THE DEVELOPMENT OF SITUATIONAL MANAGEMENT LANGUAGE IN INTELLECTUAL SYSTEM OF SEMANTIC TYPE BASED ON ONTOLOGICAL ENGINEERING Massel L.V., Massel A.G.	65
INTEGRATION OF SEMIOTICS, COGNITIVE GRAPHICS AND SEMANTIC MODELING IN INTELLIGENT SEMIOTIC SYSTEM FOR SITUATIONAL MANAGEMENT Massel L.V., Massel A.G.	71
UNIFORM ONTOLOGICAL DATA MINING PLATFORM Filippov A.A., Moshkin V.S., Shalaev D.O., Yarushkina N.G.	77
ALGORITHMS AND SOFTWARE FOR ONTOLOGICAL DATA ANALYSIS Zubtsov R.O., Semyonova V.A., Smirnov S.V.	83
FORMING OF MEDICAL INFORMATION SYSTEMS, MANAGED BY KNOWLEDGES Kobrin B.A.	89
THE TOOLS OF STRUCTURING THE SEMANTIC MODELS OF KNOWLEDGE BASES Davydenko I.T., Grakova N.V., Sergienko E.S., Fedotova A.V.	93
THE SEMANTIC MODEL OF KNOWLEDGE BASES COLLECTIVE DESIGN Davydenko I.T.	107
SEMANTIC CLASSIFICATION OF ACTIONS FOR KNOWLEDGE INFERENCE Boyko Igor	115

FORMAL SEMANTIC DESCRIPTION OF THE VARIOS TYPES OF ACTORS PURPOSEFUL ACTION	125
Shunkevich D.V., Hubarevich A.V., Svyatkina M.N., Morosin O.L.	
ASYNCHRONOUS AND PARALLEL KNOWLEDGE PROCESSING INTERACTION IN COMMON SEMANTIC MEMORY	137
Shunkevich D.V.	
FOUNDATIONS OF PLATFORM INDEPENDENCY AND OSTIS PLATFORM IMPLEMENTATION	145
Ivashenko V.P., Tatur M.M.	
PRINCIPLES AND SOFTWARE IMPLEMENTATION OF INTERACTION OSTIS SC- MEMORY AND .NET FRAMEWORK PLATFORM	151
Kayeshko A.I.	
SMART HOME, THAT CONTROLS BY KNOWLEDGES	155
Koronchik D. N	
SOFTWARE FOR THE AUTOMATIC SOLUTION OF TASKS ON PLANE GEOMETRY	159
Curbatow S.S., Lobzin A.P., Khakhalin G.K.	
ON ONTOLOGIES OF INTELLECTUAL MODELING	165
Valkman Y.R., Stepashko P.V.	
MODELS OF SUBJECT DOMAINS IN DECISION-MAKING SUPPORT SYSTEMS ON THE BASIS OF INFORMATION SPACE MONITORING	171
Dodonov A.G., Lande D.V., Kovalenko T.V.	
HOMOGENOUS SEMANTIC NETWORK FOR GENETIC ANALYSIS RESULT CLASSIFICATION	177
Kulikov A.M., Kharlamov A.A.	
THE DESIGNING OF PSEUDO CHRONOLOGICAL ORDER RELATION IN THE TERMINOLOGICAL NETWORKS	183
Malkovsky M.G., Soloviev S.Y.	
ALGORITHM OF BINARY RELATIONS EXTRACTION FROM TERMINOLOGICAL NETWORK	187
Tuzikova A.V.	
AUTOMATIC CREATION OF CLASS HIERARCHY AND SEARCH INSTANCES FROM TEXT FOR ONTOLOGY	191
Andreev I.A., Beksaeva E.A., Zhelepov A.S, Klein V.V., Serkov I.P.	
PRESENTATION AND PROCESSING KNOWLEDGE STORED IN MULTIDIMENSIONAL DATA THOUGHT ENHANCED BASE SEMANTIC HYPERGRAPH	197
Sharipbay A.A., Barlybayev A.B., Sabyrov T.S.	
AN APPROACH FOR DESIGN OF KNOWLEDGE BASES ON THE BASIS OF COMPUTER-AIDED TRANSFORMATION OF CONCEPTUAL MODELS	203
Dorodnykh N.O., Yurin A.Yu.	
SOFTWARE FOR RULE KNOWLEDGE BASES DESIGN: PERSONAL KNOWLEDGE BASE DESIGNER	209
Grishenko M.A., Dorodnykh N.O., Yurin A.Yu.	

PROGRAMMING PARADIGMS INFORMATION TECHNOLOGY STRUCTURAL SYNTHESIS Bikmullina I.I.	213
ABOUT TWO CLASSES OF SEMANTIC CORRELATION WATCHED ON THE FUNCTIONAL SEMANTIC NETWORKS Savkin L.V.	219
RUSSIA AMONG THE WORLD CENTERS OF EXCELLENCE Khoroshevsky V.F., Efimenko I.V.*	223
CREATION OF NETWORKS OF THE CO-AUTHORSHIP ACCORDING TO THE GOOGLE SCHOLAR CITATIONS SERVICE Lande D.V., Balagura I.V., Andrushchenko V.B.	233
DEVELOPMENT OF OBJECTIVE ONTOLOGY BASED METHODS FOR COMPETENCE EVALUATION OF SCIENTIFIC PROJECTS AUTHORS Rogushina J., Gladun A.	239
ANALYSIS OF INFORMATION EXCHANGE PROCESSES IN SCIENTOMETRIC DATABASES Potebnia A.V., Pogorilyy S.D.	247
RECOGNITION OF INFORMATION OPERATIONS: MULTI-AGENT APPROACH Dodonov A.G., Lande D.V., V.A. Dodonov	253
MODEL OF AGENTS COOPERATION IN THE SEMIOTIC ENVIRONMENT Kulinich A. A.	259
MODIFIED KNOWLEDGE INFERENCE METHODS BASED ON FUZZY ONTOLOGY AND SET OF USE CASES Moshkin V.S., Yarushkina N.G.	265
ALGORITHM FOR PREPROCESSING AND RECOVERY OF QUESTIONNAIRES DATA Sibirev I.V., Afanasyeva T.V.	271
ABOUT THE TEMPORAL REASONING FORMALIZATION IN THE INTELLIGENT SYSTEMS Vagin V.N., Ereemeev A.P., Guliakina N.A.	275
DEVELOPMENT OF THE SOFTWARE FOR SYSTEMS MODELLING BASED ON TEMPORAL PETRI NETS Ereemeev A.P., Korolev Y.I.	283
ON TEMPORAL MODELS FORMALISATION FOR INTELLIGENT SYSTEMS Kurilenko I.E.	289
SPECIFICATION OF WORKFLOWS IN THE EXTENDED INTERVAL ALLEN'S LOGIC Plesniewicz G.S., Nguyen Thi Minh Vu	295
TIME SERIES REDUCTION ALGORITHM TO THE SPECIFIED FORMAT Sapunkov A.A. Afanasjeva T.V.	301
DESCRIPTION APPROACH TO EXTRACTING KNOWLEDGE ABOUT GLOBAL TRENDS ANOMALIES TIME SERIES Afansieva T.V., Zavarzin D.V.	305

COGNITIVE FRACTAL ANALYSIS OF TIME SERIES: R / S ANALYSIS AND V-STATISTICS	309
Romanko A.R.	
SEMANTIC CODING BASED ON DEEP AUTOASSOCIATIVE NEURAL NETWORK	313
Golovko V.A., Kroshchanka A.A., Hachevich M.V.	
NEURAL NETWORKS IN SEMANTIC ANALYSIS	319
Averkin A.N., Yarushev S.Y.	
A NEW APPROACH TO MODELING OF ARTIFICIAL NEURONS AND THEIR INTERACTION	323
Zhilyakova L. Yu.	
REALIZATION OF USER'S INTERFACES OF VIRTUAL ENVIRONMENTS BY DECLARATIVE MODEL	327
Gribova V.V. , Fedorischev L.A	
LOGICAL-ALGEBRAIC METHODS IN CONSTRUCTING COGNITIVE SENSORS	331
Svyatkina M.N., Tarassov V.B.	
ILLUSIONS OF PERCEPTION AND REPRESENTATION: INFORMATION APPROACH	349
Fominykh I.B.	
ONTOLOGY-BASED APPROACH FOR VISUALIZATION OF SIMULATION RESULTS	355
Korshunov S.A.	
AGGREGATION OF TEXTUAL DESCRIPTION WITH SEMANTICIZED GRAPHICS IN SOLVING THE PROJECT TASKS	359
Sosnin P.I., Galochkin M.V., Lunecaks A.A.	
SET OF SOFTWARE TOOLS WITH IMPLEMENTING LOGIC FOR EVALUATION OF PSYCHOPHYSIOLOGICAL STATE	367
Poryadin A.Y., Sidorkina I.G.	
SEMANTIC ASPECTS OF METADATA REPRESENTATION IN CORPUS MANAGER SYSTEM	371
Nevzorova O.A., Mukhamedshin D.R., Kurmanbakiev M.I.	
CREATING RUSSIAN SENTIMENT LEXICON	377
Loukachevitch N.V., Levchik A.V.	
MICROSERVICE DESIGN APPROACH FOR DEVELOPING SPEECH USER INTERFACES	383
Lobanov B.M., Zhitko V.A.	
CONTEXTUAL RULE METHOD FOR MORPHOLOGICAL DISAMBIGUATION IN THE TATAR LANGUAGE	389
Gilmullin R.R., Gataullin R.R.	
SYNTACTICAL ANALYZER OF KAZAKH LANGUAGE BASED BY "LINK GRAMMAR PARSER"	393
Begimtay U.H.	
ALGORITHM AND LINGUISTIC RESOURCES FOR TEXT NORMALIZATION OF GEOGRAPHIC DOMAIN	397
Hetsevich Y.S., Lysy S.I., Kachan E.S., Marakulina P.A, Krivaltsevich A.V.	

SEMANTIC ANALYSIS OF TEXT FOR RECOGNITION OF THE ELEMENTS OF HUMAN APPEARANCE	401
Dolbin A.V., Rozaliev V.L., Orlova Y.A., Zaboлева-Zotova A.V.	
ABOUT THE PHYSICAL STRUCTURE OF SIMPLE VOWEL SOUNDS OF HUMAN SPEECH	405
Mitsianok V.V.	
VOICE USER IDENTIFICATION IN ACCESS CONTROL SYSTEMS	411
Menshakov P.A., Murashko I.A.	
ONTOLOGICAL APPROACH TO FORMATION OF CONTEXTUAL QUERIES IN ELECTRONIC ARCHIVE OF TECHNICAL DOCUMENTATION	415
Namestnikov A.M.	
SEMANTIC ANALYSIS OF TEXT USER MANUAL ENGINE ASSEMBLY WITH ONTOLOGY EDITOR FLUENT EDITOR	421
Orlova A.A., Borgest N.M.	
CONNECTION OF MODULE INTERFACE INTELLECTUAL ASSISTANT WITH CAE-SYSTEM ANSYS	427
Korovin M.D., Vlasov S.A.	
DATA PROCESSING ALGORITHMS IN ADAPTIVE RECONFIGURABLE MODULE PROTECTING DESIGNS CAD FOR RENDERING A PROFILE OF HEAT FLOW	431
Sorokin O.L., Sidorkina I.G.	
PRACTICAL IMPLEMENTATION OF MODELS OF MATURE FIELDS PROCESSES IN THE CONTEXT OF AUTOMATED CONTROL OF THE FLOWS OF DESIGN-ENGINEERING WORKS	435
Larin S. N.	
DEVELOPMENT OF A DICTIONARY OPERATIVE INDUSTRIAL PLANING BASED BY ONTOLOGICAL APPROACH	441
Karpaev S.A., Larin S.N.	
ALGORITHMIC SUPPORT TO MANAGING FUZZY MULTI-STAGE TECHNOLOGICAL SYSTEM EVOLUTION	445
Palyukh B.V., Egereva I.A.	
MODEL OF INTELLIGENT PROCESS CONTROL SYSTEMS MAINTENANCE	449
Burdo G.B., Fedotova A.V.	
SELF-ORGANIZING NETWORK EXPERT COMMUNITY IN THE TASK OF DEVELOPING SCIENCE-INTENSIVE INDUSTRIES	457
Vinogradov G.P., Vinogradov N.G.	
A FUZZY MODEL OF THE BEHAVIOR OF THE ENTREPRENEUR IN NEGOTIATIONS ABOUT INVESTING IN THE CONDITIONS OF ACCURATE KNOWLEDGE OF NATURE A POTENTIAL INVESTOR	465
Palyukh B.V., Klyushin A.Yu., Mutovkina N.Yu.	
SENSORY MODEL OF IMITATIVE BEHAVIOUR OF ROBOTS	471
Karpov V.E.	

AN ONTOLOGICAL APPROACH FOR DESIGNING INTERACTIVE VIRTUAL ENVIRONMENTS FOR A VISUAL REPRESENTATION OF PLANNED ACTIONS DURING DIALOGUE CONTROLLING A ROBOT-ASSISTANT ON-BOARD OF THE ISS	477
Kryuchkov B.I., Usov V.M., Karpov A.A.	
TELEMETRY DATA USAGE IN ON-BOARD EMERGENCY SITUATIONAL CONTROL	483
Danilin N.S., Dimitrov D.M., Sabirov I.H.	
FEATURES OF REALIZATION OF INTELLIGENT SYSTEM OF ANALYSIS AND FORECASTING OF SPACE VEHICLES	487
Iurygina I. S., Lakhin O. I., Donskov A.V., Mishurova N. V. , Sysoev D.V., Skoryupina E.G.	
DEVELOPING ONTOLOGIES FOR USE IN ADAPTING SCHEDULING OF FLIGHT PROGRAM FOR THE INTERNATIONAL SPACE STATION	493
Lakhin O., Polnikov A., Redkina K.	
SOME FEATURES OF DEVELOPMENT AND USING OF TUTORING INTEGRATED EXPERT SYSTEMS IN EDUCATIONAL PROCESS MEPHI	497
Rybina G.V., Rybin V.M., Sergienko E.S.	
METHODS AND MODELS SEMANTIC REPRESENTATION OF KNOWLEDGE IN INTELLECTUAL SYSTEMS EDUCATIONAL PURPOSES	503
Shikhnabieva T.Sh	
ABOUT THE CREATION OF ADAPTIVE SYSTEM FOR MULTIMODAL LEARNING MATERIALS GENERATION	507
Khusainov A.F., Ruseckiy K.V.	
DEVELOPMENT OF A MODEL CONSIDERING THE INFLUENCE OF EMOTIONS ON THE FORMATION OF INDIVIDUAL TRAJECTORY OF STUDENT LEARNING BASED ON NEURAL NETWORK TECHNOLOGY	511
Kozlov O.A., Mikhailov J.F.	
EVALUATING THE EFFECTIVENESS OF THE LEARNING PROCESS OF STUDENTS CAM-PROCESSING MACHINE BUILDING	517
Barabanova E.Yu.	
SEMANTIC USER MODEL OF INTELLECTUAL REFERENCE SYSTEM OF THE PROGRAM DESIGN	521
Pivovarchyk O.	
EXPERT EVALUATION METHODS, QUALITY AND RELIABILITY OF AUTOMATIC SYSTEM MANAGEMENT OF EDUCATIONAL PROCESS	527
Sharipbay A.A., Niyazova R.S., Kuzembaev B.A.	
PROBLEMS OF SEMANTIC ARRANGEMENT OF EDUCATIONAL CONTENT AND APPROACHES TO THEIR SOLUTION	531
Gorbachev N.N.	
DESIGN OF INTELLECTUAL SYSTEM TO FORECAST STAFFING NEEDS BASED ON SEMANTIC KNOWLEDGE REPRESENTATION	535
Rouba Y.A., Boika V.K., Vaitukevich Y.A, Liaushuk U.J., Piatrou S.V.,	
AGENT-BASED APPROACH TO SAFETY MANAGEMENT OF NATURAL AND TECHNOGENIC SYSTEM	539
Bogatikov VN, Vinogradov GP, Voronin Yu A	
INSTRUMENTAL PLATFORM CONCEPTION OF MARKETING ACTIVITY IN CLODE COMPUTING AREA WITH INTELLIGENCE TECHNOLOGIES USE	545
Vishniakou U.A., Kazak E.A.	

SEMANTIC ANALYSIS OF SOCIAL NETWORKS USING WOLFRAM MATHEMATICA Tyakunov A.S., Slavsky V.V.	549
USING SOCIAL NETWORKS FOR FINDING QUALIFIED GROUPS OF PEOPLE Stratnev P.U.	553
ONTOLOGY MODELS FOR SUPPORTING EXPLORATORY INFORMATION NEEDS Turmaganbetova Sh.K.	557
SOUND GENERATION BASED ON IMAGE COLOR SPECTRUM Nikitin N.A., Rozaliev V.L., Orlova Yu.A., Zaboлева-Zotova A.V.	561
PRINCIPLE OF AUTHENTIC EQUIVALENCE IN PROBLEMS OF ADAPTIVE CONTROLS IN THE CONDITIONS OF APRIORISTIC UNCERTAINTY M.Z.Babamukhamedova, M.Yu.Doshchanova, F.A.Ergashev	565
METHODOLOGY AND SOFTWARE FOR SEMANTIC ANALYSIS OF COMPLEX DYNAMICAL SYSTEMS Burak T.I., Lukashevich M.M.	569
QUALITATIVE MODELING OF TECHNICAL SYSTEMS BASED ON THE CONSTRAINT PROPAGATION METHODS Zuenko A.A.	573
APPLICATION OF BAYESIAN NETWORKS FOR ANALYSIS OF ECG Fridman R., Sagan V.Y.	579
KNOWLEDGE BASE DESIGN FOR STORAGE AND ANALYSIS BIOMEDICAL CT SCAN OF LUMBAR SPINE Kurochka K.S., Karabchikova E.A.	583
AUTHOR INDEX	587

ПРЕДИСЛОВИЕ

Основным практическим результатом исследований в области искусственного интеллекта является разработка не только интеллектуальных систем, но и технологий, обеспечивающих быстрое и качественное построение таких систем. Создание указанных технологий требует решения следующих задач:

- чёткого выделения логико-семантического уровня интеллектуальных систем, который абстрагируется от всевозможных вариантов технической реализации этих систем (в том числе и от использования принципиально новых компьютеров, ориентированных на аппаратную поддержку интеллектуальных систем);
- разработки онтологии проектирования интеллектуальных систем и унификации описания логико-семантических моделей интеллектуальных систем;
- обеспечения платформенно независимого характера логического проектирования интеллектуальных систем, результатом которого является унифицированное описание логико-семантических моделей проектируемых интеллектуальных систем;
- использования методики компонентного проектирования интеллектуальных систем, в основе которой лежит постоянно пополняемая библиотека многократно используемых компонентов интеллектуальных систем (многократно используемых подсистем, компонентов баз знаний, агентов обработки знаний, компонентов пользовательских интерфейсов);
- обеспечения семантической совместимости многократно используемых компонентов интеллектуальных систем и семантической совместимости самих интеллектуальных систем и технологий их проектирования.

Основной целью ежегодных конференций OSTIS (Open Semantic Technology for Intelligent Systems) является создание условий для расширения сотрудничества различных научных школ, вузов и коммерческих организаций, направленного на разработку и применение массовых и постоянно совершенствуемых технологий компонентного проектирования интеллектуальных систем.

VI международная научно-техническая конференция «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2016) посвящается памяти выдающегося ученого в области информатики и искусственного интеллекта, доктора технических наук, профессора, действительного члена Российской академии естественных наук Эдуарда Викторовича Попова.

Председатель Программного комитета конференции OSTIS-2016
Председатель Совета Российской ассоциации искусственного интеллекта
Кузнецов Олег Петрович

FOREWORD

Development not only intelligent systems but also technologies that ensure fast and efficient construction of intelligent systems are the main practical result of research in artificial intelligence. Development of these technologies requires the following tasks:

- precise separation of logical and semantic level of intelligent systems, which is abstracted from various versions of the technical implementation of these systems (including the use of innovative computer-based hardware support for Intelligent Systems);
- developing an ontology design of intelligent systems and unifying description of logical and semantic models of intelligent systems;
- provide a platform independent nature of the logical design of intelligent systems, which result is a unified description of logical and semantic models of an intelligent systems;
- use of a component design methodology of intelligent systems, which is based on permanently increasing library of reusable components of intelligent systems (reusable subsystems, knowledge bases components, knowledge processing agents, user interfaces components);
- ensuring of semantic compatibility of reusable components of intelligent systems and semantic compatibility of these intelligent systems and technologies of such systems design.

Creating the conditions for the expansion of cooperation between different scientific schools, universities and business organizations, aimed on development and using of mass and continuously improved component design technologies for intelligent systems is the main purpose of annual conferences OSTIS (Open Semantic Technology for Intelligent Systems).

VI International scientific and technical conference “Open Semantic Technology for Intelligent Systems” (OSTIS-2016) is dedicated to the memory of outstanding scientist in the field of computer science and artificial intelligence, doctor of technical sciences, professor, member of the Russian Academy of Natural Sciences Edward Viktorovich Popov.

Programme Committee Chair.

Chairman of the Council of the Russian Association for Artificial Intelligence

Kuznetsov Oleg Petrovich

ЭДУАРД ВИКТОРОВИЧ ПОПОВ

5 января 1940 – 19 октября 2015



Выдающийся ученый в области информатики и искусственного интеллекта, доктор технических наук, профессор, действительный член Российской академии естественных наук.

Он был одним из основоположников искусственного интеллекта в СССР и России. С его уходом завершилась целая эпоха становления и развития искусственного интеллекта в нашей стране.

Вклад Э.В.Попова в эту отрасль науки трудно переоценить: достаточно вспомнить непревзойденный до сих пор прорыв в компьютерном понимании естественного (русского) языка (система ПОЭТ), создание передовых отечественных экспертных систем (системы ЭКСПЕРТ, ЭКО), разработку систем интеллектуального имитационного моделирования, управления знаниями.

Именно он расширил границы исследований по ИИ, занимаясь проблемами взаимодействия отдельных направлений ИИ с компьютерной лингвистикой, программной инженерией, робототехникой, менеджментом и другими науками. Он также стоял у истоков новой научно-практической области на стыке искусственного интеллекта, инженерных наук, теории организаций и моделирования бизнес-процессов, ныне получившей название «Инжиниринг бизнеса» («Инжиниринг предприятий»). Его перу принадлежит целая серия великолепных научных монографий, в том числе: «Алгоритмические основы интеллектуальных роботов и искусственного интеллекта», «Общение с ЭВМ на естественном языке», «Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ», «Общение конечных пользователей с системами обработки данных», «Реинжиниринг бизнеса: реинжиниринг организаций и информационные технологии», один из первых и лучших в нашей стране учебников по интеллектуальным системам «Статические и динамические экспертные системы». Под его редакцией издана первая книга трехтомного Справочника по искусственному интеллекту «Системы общения и экспертные системы». Выход в свет каждой из этих книг был важным событием в нашей научной жизни.

Эдуард Попов родился 5 января 1940г. в Ленинграде. Детство его пришлось на тяжелейшие блокадные годы, которые он вместе со своей матерью пережил в городке на Неве. В 1957г. после окончания с серебряной медалью средней школы Э.В.Попов поступил в Ленинградский институт точной механики и оптики (ЛИТМО), который окончил с отличием в 1963г. по специальности «Математические и счетно-решающие приборы». По распределению был направлен в КБ электронной

техники (КБ-2) – одну из первых советских организаций, занимавшихся проблемами микроэлектроники, которую тогда возглавлял знаменитый ученый и инженер, выходец из США Филипп Старос.

С 1963 по 1974гг. Э.В.Попов работал в Ленинградском конструкторско-технологическом бюро (министерства электронной промышленности), а затем с 1974 по 1989гг. – в НИИ «Восход». В 1967г. защитил кандидатскую диссертацию по вопросам программного обеспечения ЭВМ, а в 1982г. – докторскую диссертацию по проблемам общения с ЭВМ на ограниченном естественном языке. Под руководством профессора Э.В.Попова в НИИ «Восход» совместными усилиями специалистов по ИИ, лингвистов из МГУ и программистов были разработаны первые в СССР промышленные системы доступа на естественном языке к большим базам данных (ПОЭТ, АИСТ, ЛИНГВИСТ).

С 1989 по 1993 гг. был начальником отдела в Международном центре по информатике и электронике, а с 1994г. в течение более десяти лет работал заместителем директора Российского научно-исследовательского института информационных технологий и систем автоматизированного проектирования. В этот период возглавляемый им творческий коллектив разработал ряд оригинальных инструментальных средств для создания экспертных систем, доведенных до уровня коммерческих прототипов.

Эдуард Викторович Попов, являясь в течение длительного времени одним из руководителей ведущих институтов, занимавшихся разработкой автоматизированных информационных систем широкого назначения, внес большой вклад в формирование Межотраслевой государственной программы по прикладным интеллектуальным системам, Государственной программы по перспективным информационным технологиям, и др.

Э.В.Попов – один из отцов-основателей Советской (ныне Российской) ассоциации искусственного интеллекта. Он долгие годы был ее вице-президентом. С 2001 по 2006 гг. – главный редактор научного журнала РАИИ «Новости искусственного интеллекта», также с 2001г. он был председателем редакционного совета книжной серии «Науки об искусственном». Был организатором и главной движущей силой межвузовского коллектива РАИИ, получившего в 2000-м году Премию Президента России в области образования за 1999г. за создание учебно-методического комплекса «Методы, модели и программные средства конструирования интеллектуальных систем принятия решений и управления». Его также следует считать вдохновителем проекта открытых семантических технологий проектирования интеллектуальных систем OSTIS.

У Эдуарда Викторовича Попова был редкий творческий дар: как порождать новые научные идеи, так и превращать их в реальные программные системы, заражая своей энергией и убежденностью сотрудников руководимых им научных коллективов.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СИНТЕЗ СЕРВИСНОГО И АРХИТЕКТУРНОГО ПОДХОДОВ К ПРОЕКТИРОВАНИЮ СИСТЕМ УПРАВЛЕНИЯ ЗНАНИЯМИ

Кудрявцев Д. В., Гаврилова Т. А.

*Санкт-Петербургский государственный университет,
г. Санкт-Петербург, Россия*

d.v.kudryavtsev@gsom.pu.ru

gavrilova@gsom.pu.ru

Доклад посвящен использованию сервисного и архитектурного подходов для проектирования систем управления знаниями. Кратко описаны основы методологии ПРОТЕСИС (ПРОект-ТЕхнология-СИСтема), разработанной в рамках проекта РФФИ ИНС-Порт (Интеллектуальные сервисы поддержки порталов знаний на основе онтологий). В частности, рассмотрено использование «строительных блоков» для проектирования систем управления знаниями и представлены фрагменты каталогов программных сервисов и компонент.

Ключевые слова: система управления знаниями, сервисы, архитектура предприятия.

Введение

Мировой интерес к работам в области интеллектуальных сервисов и систем управления знаниями только нарастает (Lusch & Vargo, 2007; Qui, 2007; Spohrer et al., 2008; Dalkhir, 2013). Интеллектуальные сервисы и онтологии имеют высокий потенциал повышения эффективности порталов знаний, однако применение данных технологий для разработки промышленных СУЗ до сих пор имеет ограниченные масштабы (Holsapple, 2013; Gavrilova & Leshcheva, 2015). Разрабатываемая методология проектирования корпоративных порталов знаний ПРОТЕСИС (ПРОект-ТЕхнология-СИСтема) предназначена для использования инженерами по знаниям и бизнес-аналитиками, ответственными за разработку и внедрение систем управления знаниями (СУЗ) в организации.

Проект ИНС-ПОРТ ориентирован на решение проблемы по формированию системы интеллектуальных сервисов поддержки жизненного цикла порталов знаний — от проектирования до промышленной эксплуатации.

Цель проекта — создание методологии и технологии, позволяющих создавать и тиражировать порталы знаний предприятий, научных проектов, учебно-методических центров и др. Методология ПРОТЕСИС ориентирована на работу с онтологиями предметных областей на основе структурно-визуального подхода. Также она

позволяет объединить сервис-ориентированный подход с методами создания и поддержки СУЗ.

1. Архитектурный и сервис-ориентированный подходы к проектированию СУЗ

В основу разрабатываемой методологии ПРОТЕСИС положен архитектурный подход к проектированию СУЗ, основанный на моделировании архитектуры предприятия (в т.ч. функций, организационных звеньев, бизнес-процессов, информационных систем, данных) (Lankhorst M. et al., 2013; Зиндер, 2008). Данный подход особенно актуален для проектирования СУЗ, которые, по сути, являются системой систем (system of systems) — интегрируют множество информационных систем и организационных механизмов предприятия с целью повышения эффективности процессов преобразования знаний.

Проектирование информационных и бизнес-систем на основе архитектурного подхода предполагает создание визуальных многоаспектных моделей предприятия и его ИТ-архитектуры. Для создания таких моделей используются специальные языки моделирования, например, Archimate (The Open Group, 2012) в программном инструменте Archi (<http://www.archimatetool.com/>).

В ArchiMate выделяют три уровня описания архитектуры: бизнес, приложения (уровень программ) и технологии (уровень оборудования).

Каждый уровень описания, в дополнение, описывается с учетом трех аспектов: Пассивная структура (Passive structure), Деятельность (Behavior) и Активная структура (Active structure).

Преимуществом данного языка с точки зрения нашего проекта ИНС-ПОРТ является его ориентированность на сервисный подход (Hefly, Murphy, 2008; Spohrer, Stephen, 2010) — идентификация многоуровневой системы сервисов на предприятии: бизнес-сервисов (business service), программных сервисов (application service) и инфраструктурных сервисов (infrastructural service), см. рис. 1.

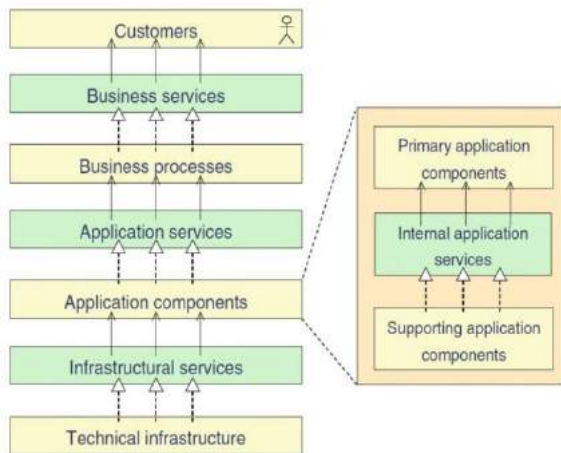


Рисунок 1 – Многоуровневая система сервисов в языке Archimate (Lankhorst M. et al., 2013)

На русском языке примеры моделей архитектуры предприятия на языке Archimate представлены в работе (Кудрявцев и др, 2014; Рубенчик, 2014).

Инструменты УЗ (Young, 2010) могут рассматриваться в качестве поставщиков сервисов УЗ. В соответствии с сервис-ориентированным подходом (Hefly, Murphy, 2008; Spohrer, Stephen, 2010) такие бизнес-сервисы основаны на соответствующих организационных процессах и механизмах (например, системе мотивации), а также будут использовать разнообразные программные сервисы. На рисунке 2 представлено сервис-ориентированное описание одного из инструментов УЗ — сообщества практиков, выполненное в соответствии с архитектурным подходом (Lankhorst M. et al., 2013; Зиндер, 2008) на языке Archimate (The Open Group, 2012) в программном инструменте Archi (<http://www.archimatetool.com/>)

Представленные на рисунке 2 программные сервисы могут использоваться для реализации различных бизнес-сервисов и инструментов УЗ, например, «Хранение информации» и «Форумы» могут применяться в управлении идеями. Такой модульный подход позволяет с одной стороны обеспечить эффективность за счет возможности оптимизации работы отдельных сервисов, а с другой — достичь гибкости за счет быстрой сборки новых сервисов путем комбинирования существующих (подобно блокам конструктора Lego).

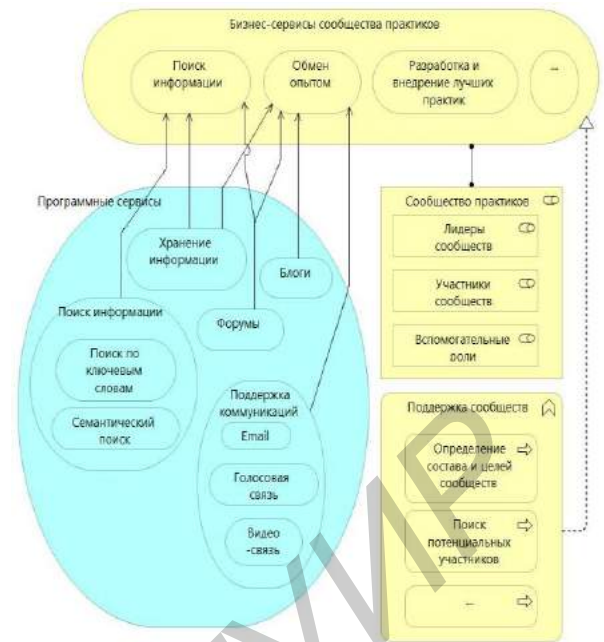


Рисунок 2 – Сервис-ориентированное описание сообщества практиков (одного из инструментов УЗ)

2. Использование «строительных блоков» для проектирования СУЗ

Методология ПРОТЕСИС основана на формализации и систематизации «строительных блоков» корпоративных порталов знаний (в том числе семантических) — бизнес- и программных сервисов УЗ. Соединение вышеуказанных «строительных блоков» планируется реализовать с помощью интеллектуальных (мета)сервисов. Обзор понятия и разновидности интеллектуальных сервисов рассмотрены в работе (Гаврилова, Власов, 2014). Предлагаемые интеллектуальные сервисы помогут подобрать требуемый набор сервисов УЗ для каждой области деятельности / функции предприятия. В части использования «строительных блоков» методология ПРОТЕСИС развивает идеи представленные в компонентной методологии реинжиниринга бизнес-процессов (Тельнов, 2004).

«Строительные блоки» для проектирования корпоративных порталов знаний основаны на каталогах (справочниках) сервисов, программных компонент и информационных ресурсов. Такие каталоги позволяют стандартизировать описание «строительных блоков», облегчают их поиск, отбор и конфигурирование в интеллектуальных сервисах поддержки порталов знаний. На рисунке 3 представлен фрагмент каталога программных сервисов, а на рисунке 4 — верхний уровень каталога программных компонент, используемых в СУЗ.

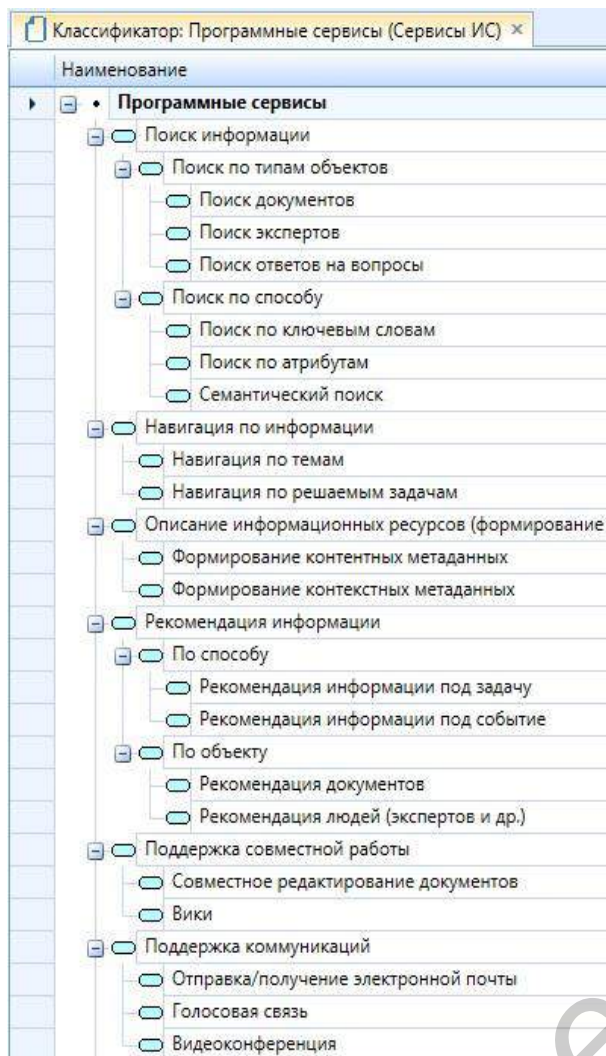


Рисунок 3 – Фрагмент каталога программных сервисов СУЗ



Рисунок 4 - Каталог программных компонент для СУЗ (верхний уровень)

Формирование каталогов (справочников) и «строительных блоков» для проектирования порталов знаний производилось на основе анализа

существующих СУЗ, обследований компаний, занимающихся управлением знаниями, вторичных источников информации и экспертного опыта команды проекта.

Поскольку проектирование СУЗ (состоящей из ИТ и организационных составляющих) можно рассматривать как часть проектирования архитектуры предприятия, интеллектуальный сервис поддержки проектирования СУЗ будет дополнять функциональность стандартных инструментов управления архитектурой предприятия (Matthes, 2008; Кудрявцев и др., 2014), в которых существуют базовые средства для работы с каталогами (справочниками) сервисов и «строительными блоками».

Заключение

В настоящее время существуют набор методов проектирования СУЗ на основе онтологий, представленные в работах [Тузовский и др., 2005; Кудрявцев, 2010; Тельнов, Казаков, 2011]. Однако данные методы требуют дополнительной детализации и конкретизации, причем не столько на уровне низкоуровневого проектирования, сколько на этапах аудита знаний, верхне-уровневого проектирования системы, планирования и выбора низкоуровневых программных сервисов.

Методология ПРОТЕСИС основана на формализации и систематизации «строительных блоков» корпоративных порталов знаний, в том числе семантических – основанных на онтологиях. Соединение вышеуказанных «строительных блоков» реализуется с помощью интеллектуальных сервисов

В целом, разрабатываемая методология ПРОТЕСИС, с одной стороны, находится в русле мировых трендов, с другой стороны, является оригинальным новым методом, ориентированным на создании корпоративных порталов знаний.

Проект поддержан грантом РФФИ 14-07-00294 «Интеллектуальные сервисы поддержки порталов знаний на основе онтологий».

Библиографический список

[Гаврилова, Власов, 2014] Гаврилова Т.А. Власов С.А. (2014) Интеллектуальные сервисы поддержки порталов знаний // Труды 2 –ого Международного Поспеловского симпозиума «Гибридные и синергетические интеллектуальные системы ГИСИС-2014», Светлогорск, 2014. – с. 94–100.

[Зиндер, 2008] Зиндер Е. З. (2008) Архитектура предприятия в контексте бизнес-реинжиниринга // Intelligent Enterprise. Ч. 1. № 4. С. 46; Ч. 2. № 7. – с. 183.

[Кудрявцев, 2010] Кудрявцев Д.В. Системы управления знаниями и применение онтологий: Учеб. пособие / Д.В. Кудрявцев. - СПб.: Изд-во Политехн. ун-та, 2010.

[Кудрявцев и др., 2014] Кудрявцев Д. В. Технологии бизнес-инжиниринга: учеб. пособие / Д. В. Кудрявцев, М. Ю. Арзуманян, Л. Ю. Григорьев. — СПб. : Изд-во Политехн. ун-та, 2014.

[Рубенчик, 2014] Рубенчик А. (2014) Обзор языка ArchiMate. Основные понятия и примеры использования. Information management. № 6.

[Тельнов, 2011] Тельнов Ю.Ф., Казаков В.А. (2011) Проектирование систем управления знаниями. М.: Изд.центр ЕАОИ, 206 с.

[Тельнов, 2004] Тельнов Ю.Ф. (2004) Реинжиниринг бизнес-процессов: компонентная методология. – 2-е изд. - М.: Финансы и статистика. – 320 с.

[Тузовский и др., 2005] Тузовский А. Ф., Чirikov С. В., Ямпольский В. З. (2005) Системы управления знаниями. — Томск : Изд-во науч.-техн. литературы.

[Archi, 2015] Archi. The Free ArchiMate Modelling Tool. Accessed 20 October 2015, <http://www.archimatetool.com>.

[Dalkir, 2013] Dalkir, K. (2013). Knowledge management in theory and practice. Routledge.

[Gavrilova, Leshcheva, 2015] Gavrilova T., Leshcheva I. (2015) Building Collaborative Ontologies: A Human Factors Approach // Chapter in Book “Collaborative Knowledge in Scientific Research Networks“ (Eds. P. Diviaco, P. Fox, C. Pshenichny, A. Leadbetter), IGI publishing, USA. - pp. 305-324.

[Hefke et al, 2006] Hefke, M., Abecker, A., Jäger, K. (2006). Portability of best practice cases for knowledge management introduction. J. Univers. Knowl. Manag, 1(3), 235–254.

[Hefly, Murphy, 2008] Hefly B., Murphy W. (2008) Service Science, Management and Engineering Education for the 21st Century. — Springer.

[Holsapple, 2013] Holsapple, C. ed. Handbook on knowledge management 1: Knowledge matters. Vol. 1. Springer Science & Business Media, 2013.

[Lankhorst, 2013] Lankhorst M. et al. (2013) Enterprise Architecture at Work: Modelling, Communication and Analysis (The Enterprise Engineering Series). Springer, Third Edition.

[Lovelock, Wirtz, 2007] Lovelock, C. and Wirtz, J. (2007). Services Marketing: People, technology, strategy. 6th Edition. Pearson Prentice Hall: US.

[Lusch, Vargo, 2006] Lusch, R.F. and Vargo, S. L. (2006). The Service-Dominant Logic of Marketing: Dialog, Debate, and Directions. M. E. Sharpe, NY.

[Matthes et al, 2008] Matthes, F., Buckl, S., Leitel, J., & Schweda, C. M. (2008). Enterprise architecture management tool survey 2008. Techn. Univ. München.

[Qiu, 2007] Qiu, R.G. (2007). Service Science: Scientific Study of Service systems, published in FAIM 2007

[Spohrer, Stephen, 2010] Spohrer J., Stephen K. (2010) Service science, management, engineering, and design (SSMED): an emerging discipline. Information Systems and New Applications in the Service Sector: Models and Methods: Models and Methods, pp. 194–227.

[Young, 2010] Young R. (Editor) (2010) Knowledge management tools and techniques manual. Asian Productivity Organization.

SYNTHESIS OF SERVICE-BASED AND ARCHITECTURAL APPROACHES TO KNOWLEDGE MANAGEMENT SYSTEM DESIGN

Kudryavtsev D.V., Gavrilova T.A.

Saint-Petersburg State University, Saint-Petersburg, Russia

d.v.kudryavtsev@gsom.pu.ru

gavrilova@gsom.pu.ru

The work describes how methods of service engineering and enterprise architecture can be used in knowledge management system design. The basic principles of PROTESYS methodology, which is developed within RFBR INS-Port (Intelligent services for knowledge portal support), are discussed. Particularly, the usage of building blocks for knowledge management system design is described. The suggested building blocks are based on services – knowledge management business, application and infrastructure services. These services and components, which provide services, are organized in catalogs in order to simplify reuse. The article also demonstrates fragments of these catalogs. The selection of necessary building blocks is supported by knowledge-based intelligent (meta)service.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ УПРАВЛЯЕМОЙ СИСТЕМЫ ПОДДЕРЖКИ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Клещев А.С., Шалфеева Е.А.

*Институт автоматики и процессов управления,
г. Владивосток, Россия*

kleshev@iacp.dvo.ru

shalf@iacp.dvo.ru

Данная работа посвящена проблеме управления качеством баз знаний, используемых в процессе принятия решений повседневной интеллектуальной деятельности специалистов. Выделены Технологические этапы автоматизации интеллектуальной деятельности конкретной предметной области и предложена концептуальная архитектура системы автоматизации, которая позволяет решать множество взаимосвязанных интеллектуальных задач известных классов с использованием управляемых баз знаний.

Ключевые слова: автоматизация интеллектуальной деятельности; управление базами знаний; решатель задачи; концептуальный архитектурный проект.

Введение

Автоматизация профессиональной интеллектуальной деятельности и управления ее качеством требует не только решения проблемы поддержки интеллектуальной деятельности, но и решения проблемы правильности применения знаний при этом. Ключевыми аспектами современной парадигмы автоматизации интеллектуальной деятельности являются: понятные специалистам базы знаний (БЗ); полезные БЗ в течение всего времени их эксплуатации; возможность включения в них новых научных результатов. Недостаточное внимание поддержке процесса управления качеством баз знаний, используемых в процессе принятия решений, по-видимому, объясняет тот факт, что на сегодняшний день экспертные системы не «встроены» в организационную структуру учреждений такой, например, отрасли, как медицина, для систематической поддержки повседневной интеллектуальной деятельности специалистов [Жариков, 2008; РусБИТех, 2013].

Целью настоящего исследования является разработка концептуального архитектурного проекта системы автоматизации предметной области с интеллектуальной деятельностью (на примере медицинской области), в которой решается множество взаимосвязанных задач известных классов и требуется управление множеством взаимосвязанных баз знаний.

1. Место управления качеством в процессе автоматизации произвольной интеллектуальной деятельности

Поскольку полезность интеллектуальной деятельности определяется ее качеством, то параллельно с повседневным выполнением интеллектуальной деятельности, должно осуществляться управление ее качеством. Качество интеллектуальной деятельности определяется величиной риска ошибочных решений и характером ущерба от последствий таких ошибочных решений. Качество интеллектуальной деятельности в значительной степени зависит от качества используемых знаний.

Поэтому процесс управления качеством интеллектуальной деятельности состоит, прежде всего, в повышении качества знаний, используемых специалистами - совершенствования правильности знаний (при правильном применении знаний среди множества решений должно быть правильное), точности знаний (при правильном применении знаний получается единственное правильное решение), а также правильности и точности их применения.

Для оценки текущего уровня качества интеллектуальной деятельности и управления этим качеством обычно используются различного рода отчеты о результатах деятельности, включающие информацию о решениях, принимаемых специалистами в процессе своей деятельности, и *заклучения* (подтвержденные правильные решения

задач), такие отчеты могут быть использованы ответственными лицами (группой экспертов высокой квалификации), в частности, для выявления неверных или несовершенных знаний.

Поскольку для автоматизации интеллектуальной деятельности необходима поддержка не только повседневного выполнением интеллектуальной деятельности, но и управления ее качеством, требуется создавать комплекс из экспертных систем и средств обеспечения качества их решений. Сложность общей задачи управления качеством используемых в многоэтапном процессе принятия решений диктует необходимость разбиения интеллектуальной деятельности на отдельные задачи, с которыми связаны задачи управления качеством используемых знаний. (Для управляемости качеством знаний, используемых на каждом этапе принятия решения, целесообразно использовать принцип системного анализа, диктующий разбиение сложной задачи на ее составляющие - менее сложные.) Поэтому в составе автоматизирующих комплексов предпочтительно иметь множество экспертных систем (ЭС), предназначенных для решения задач разных классов.

В медицине классов таких задач несколько: диагностика, лечение, прогноз изменения состояния пациента, наблюдение за ним (мониторинг). По аналогии с другими предметными областями совокупно эти задачи составляют единую задачу управления сложной системой [Гаврилова и др., 2000; Джексон, 2001] (здесь - задачу управления здоровьем пациента).

2. Идентификация автоматизируемых интеллектуальных задач

Поскольку деятельность в предметной области включает в себя повседневную, в том числе интеллектуальную, контроль принимаемых решений и деятельность по управлению знаниями [Клещев и др., 2015], то (для обеспечения поддержки всех этих трех видов деятельности) при системном анализе выполняется:

- идентификация интеллектуальных задач в предметной области;
- разработка онтологий для каждой такой задачи;
- постановка каждой интеллектуальной задачи;
- разработка или выбор методов решения и соответствующих им алгоритмов для интеллектуальных задач.

По завершении этих работ целесообразно приступить к концептуальному проектированию системы автоматизации и к инженерии баз знаний, причем переход к инженерии баз знаний возможен после реализации ряда подсистем для инженерии знаний, выделенных при концептуальном проектировании [Клещев и др., 2015]. Для того

чтобы при проведении системного анализа для автоматизации профессиональной интеллектуальной деятельности рационально выделять отдельные подзадачи, методы решения которых уже известны, требуется обратиться к современной классификации постановок задач, охватывающей весь спектр задач, относящихся к области разработки систем, основанных на знаниях. Предложенная в [Клещев и др., 2015] многоуровневая классификация задач даёт возможность в процессе системного анализа переходить к математическим постановкам различных задач, конкретизируя те понятия и их абстракции, которые содержатся в этой классификации.

Среди множества подзадач задачи управления организмом в медицине *диагностика* в сочетании с *запросом дополнительной информации* и *составлением плана действий* с параллельно осуществляемым *прогнозом результата воздействий* наиболее очевидны.

В задаче *диагностики* дано: результаты наблюдения признаков, характеристик системы и произошедших событий (R_{Σ}); база знаний о клинической картине каждого заболевания, база знаний о нормальных значениях наблюдаемых признаков (KB_{Σ}).

Требуется найти: все возможные причинно-следственные модели системы, включающие диагноз D ($AS_{\Sigma}(R_{\Sigma})$) и согласованные с результатами наблюдений R_{Σ} , относительно которых все предложения из БЗ KB_{Σ} истинны.

В задаче *запроса дополнительной информации для распознавания* дано: база знаний о клинической картине каждого заболевания (KB_{Σ}), удовлетворяющая условию разделимости классов (диагнозов); R_{Σ} , такое, что для него множество гипотез ($H_{R,KB,\Sigma}$) имеет мощность не меньше двух.

Требуется найти: такой запрос дополнительной информации (Q_{Σ}) для результатов, что множество гипотез ($H_{R',KB,\Sigma}$), построенное для обновленного R_{Σ} , имеет меньшую мощность.

В задаче *лечения (планирования воздействий)* дано: значения характеристик системы (R_0); диагноз D ; желаемый результат наблюдений признаков - состояние *норма* (R^{cond}), база знаний о схемах лечения, база знаний о лечебных средствах - средствах медикаментозного воздействия).

Требуется найти: множество медикаментозных воздействий ($R_{ev} = \{m_1, \dots, m_k\}$), при которых новые результаты наблюдения признаков (как функции времени в соответствии с утверждениями базы знаний) будут удовлетворять условиям R^{cond} .

В задаче *прогноза результата воздействий* дано: значения характеристик системы (R_0); диагноз D , планируемые воздействия (R_{ev}); конечное множество моментов времени T ; база знаний о лечебных средствах.

Требуется найти: значения всех признаков во все моменты времени T в причинно-следственной модели $(AS_{\Sigma}(R_O, D, R_{ev}))$ [Клещев и др., 2015a].

Эти классы задач соответствуют подзадачам повседневной деятельности. Поскольку медицина относится к тем областям деятельности, где предусмотрена процедура подтверждения решений специалиста, для повышения уровня используемых знаний применим Метод монотонного усовершенствования БЗ [Клещев и др., 2015], называемый в данном исследовании *управлением знаниями*. Решение, принимаемое врачом, в дальнейшем признается правильным или ошибочным (например, правильным - по итогу выздоровления пациента в результате лечения, ошибочным - если выздоровление произошло только в результате применения другого, более действенного лечения).

Для монотонного усовершенствования ставшее известным правильное решение конкретной задачи (прецедент) сравнивается с объяснением, сформированным системой, и, если объяснение было неправильным, то пару <известные данные об объекте, заключение о правильности> следует использовать как прецедент для исправления базы знаний системы.

Критерий для оценки базы знаний по диагностике - наличие правильного (согласно заключению) диагноза в объяснении списка альтернативных диагнозов, составленном ЭС.

Минимальный критерий для оценки базы знаний о лечении - наличие каждого рекомендованного лечебного средства (ЛС) в правильно назначенном лечении (из прецедента) в списке R_{ev} альтернативных рекомендаций, составленном ЭС.

Схематическая постановка задачи модификации знаний (для задачи лечения) такова. Дано: Объяснение способа лечения, сформированное системой $(R_{ex} \cup R_O \cup D, R_{ev}, R'_{ex})$ и согласованное с базой знаний версии 1, прецедент $(R_{ex} \cup R_O \cup D, R^{prec}_{ev}, R^{prec}_{ex})$, согласно которому желаемое состояние было достигнуто с помощью воздействия, отсутствующего в предложенном системой списке возможных лечебных воздействий $(m_i \in R^{prec}_{ev} \text{ и } m_i \notin R_{ev})$.

Требуется найти: базу знаний версии 2, обеспечивающую Объяснение списка возможных лечебных воздействий, включающего воздействие m_i $(R_{ex} \cup R_O \cup D, R^{prec}_{ev}, R^{prec}_{ex})$ [Шалфеева и др., 2015].

Подобные классы задач соответствуют подзадачам регулярной деятельности управления качеством знаний, сопутствующей повседневной деятельности. Деятельность по управлению качеством знаний должна осуществляться экспертами, которым требуются удобные средства формирования обучающей выборки из

прецедентов, средства автоматического формирования на основе онтологии очередного варианта модификации БЗ и средства его оценивания.

Важно убедиться, что получен вариант БЗ, позволяющий давать правильные объяснения для всех задач, уже имеющихся на этот момент в базе прецедентов.

3. Концептуальное проектирование

Концептуальное проектирование системы состоит из множества шагов [Pressman, 2001].

В медицине на шаге идентификации (сколько и каких потребуется) программных подсистем для автоматизации повседневной «обычной» деятельности должны быть предусмотрены автоматизированные «рабочие места» (АРМ) регистратора, лаборанта, позволяющие добавлять значения постоянных или измеренных значений признаков в ИБ как часть подсистемы документирования.

На шаге идентификации подсистем для автоматизации повседневной интеллектуальной деятельности (в отдельном медицинском звене или целой отрасли) должны быть идентифицированы все базы знаний, используемые для поддержки решения каждой их интеллектуальных подзадач, должны быть предусмотрены «решатели», генерирующие на основе знаний:

- предложения по получению наиболее показательного признака,
- объяснения диагноза, обусловленного значениями известных наблюдений,
- объяснения выбора одних средств медикаментозного воздействия и отклонения других,
- прогноз развития значений наблюдаемых признаков для дальнейшего их мониторинга.

На шаге идентификации программных подсистем для инженерии баз знаний должны быть идентифицированы редакторы всех баз знаний (о клинической картине каждого заболевания, о нормальных значениях наблюдаемых признаков, о схемах лечения, о ЛС и их воздействиях). Все редакторы обеспечивают работу с информацией в терминах зафиксированной онтологии предметной области. Инженерия баз знаний требует включать в комплекс также и «рабочее место» эксперта для поддержки анализа проверенных решений задач врача и отбора прецедентов, подсистему для проверки соответствия очередного варианта базы знаний - базе прецедентов, подсистемы индуктивного формирования знаний (в частности, знаний о клинической картине каждого заболевания), подсистемы проверки соответствия знаний базе прецедентов.

На шаге построения схемы взаимосвязи всех подсистем в единой системе планируются

интерфейсы взаимодействия всех идентифицированных подсистем с соответствующими типами пользователей, с информационными ресурсами (базами знаний, текущими историями болезней, базами прецедентов и хранилищами другой вышеупомянутой информации), друг с другом в некоторых случаях [Клещев и др., 2015].

В результате концептуального проектирования системы автоматизации появляется модель системы (top-level architecture flow diagram), состоящей из подсистем, связанных информационными потоками друг с другом, с хранилищами информации и с внешним окружением [Pressman, 2001]. Пример фрагмента такой модели для автоматизации медицинской деятельности представлен на рис. 1.

На рисунке показано, с какими частями системы автоматизации взаимодействует специалист, решающий задачи, с какими – команда управления качеством БЗ (эксперты, ответственные за качество знаний). Подсистема-редактор доступна не только врачам,

но и медперсоналу, который с решателями не работает, а занимается только вводом и редактированием данных о пациенте. За кадром осталась подсистема документооборота со специализированными редакторами баз данных и компонентами, генерирующими нужные отчеты по хранимым данным в этих базах для «команды управления» деятельностью (руководителей разных звеньев).

Специалистам каждого первичного звена предоставляется возможность взаимодействовать с совокупностью решателей ЭС, устанавливаемой на их рабочих местах, вводить информацию с помощью специализированных редакторов.

Решатель диагностики формирует один (или несколько возможных диагнозов) и их объяснения. Врач может не согласиться ни с одним из них и поставить свой диагноз. Решатель лечения должен строить план лечения для каждого из сгенерированных возможных диагнозов и их объяснения.

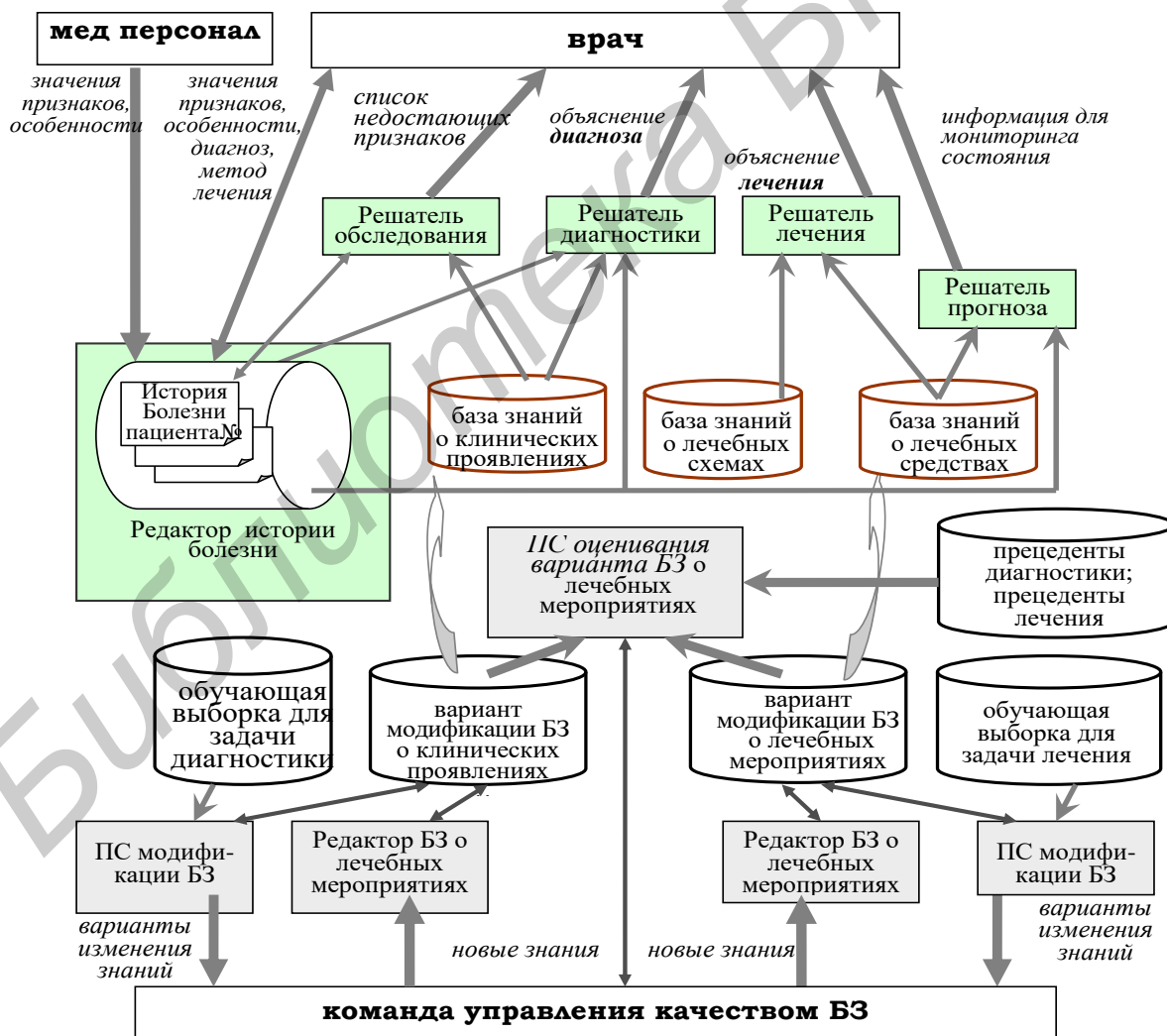


Рисунок 1 – Фрагмент концептуальной архитектуры системы автоматизации медицинской повседневной деятельности

Первичной разработкой баз знаний занимаются эксперты. Они же занимаются их сопровождением и управлением, являясь пользователями подсистем управления качеством знаний в процессе всего периода эксплуатации системы. В медицине с каждым профилем (офтальмология, урология, и т.д.) связана своя команда управления качеством баз знаний. Работая с однотипными АРМами, эксперт каждого профиля «отвечает» за качество своей совокупности баз знаний. Качество улучшаемых БЗ будет зависеть от сложности задач, которые специалистам приходится решать. Привлечение для управления знаниями дополнительных специалистов экономически оправдано в случае облачной реализации систем автоматизации (для уровня всей отрасли), если на центральном сервере установлены решатели, базы знаний, архив решенных задач и все подсистемы для управления.

Предложенная концептуальная архитектура базируется на «модели репозитория» и соответствует взаимозависимости автоматизируемых процессов в рассматриваемой предметной области. Всем пользователям интеллектуальной программно-информационной системы (врачам, медперсоналу, экспертам по знаниям и экспертам по подтверждению принятых решений) обеспечивается доступ к программным средствам.

4. Подход к реализации комплекса для поддержки принятия решений с управляемой базой знаний

Требования, относящиеся к способам формирования и использования баз знаний и другой хранимой информации, к доступности компонентам системы диктуют выбор «среды разработки» для системы автоматизации медицинской деятельности. Одно из главных требований относится к возможности представления знаний о клинической картине заболеваний не только с учетом некоторой динамики развития признаков, но и с учетом вариантов развития такой динамики. Иерархические однородные семантические сети отвечают предъявляемым требованиям к естественной для специалистов структуре описания знаний о клинических проявлениях заболеваний со множеством вариантов развития динамики признаков (рис. 2).

Другое требование связано с возможностью отслеживания прецедентов и своевременного их применения для усовершенствования знаний. В медицине важно использовать информацию о новом прецеденте, обнаруженном в любом учреждении отрасли, чтобы улучшенной с его помощью базой знаний могли пользоваться все. Поэтому облачный доступ к средствам оценивания и модификации БЗ, является наиболее адекватным выбором.

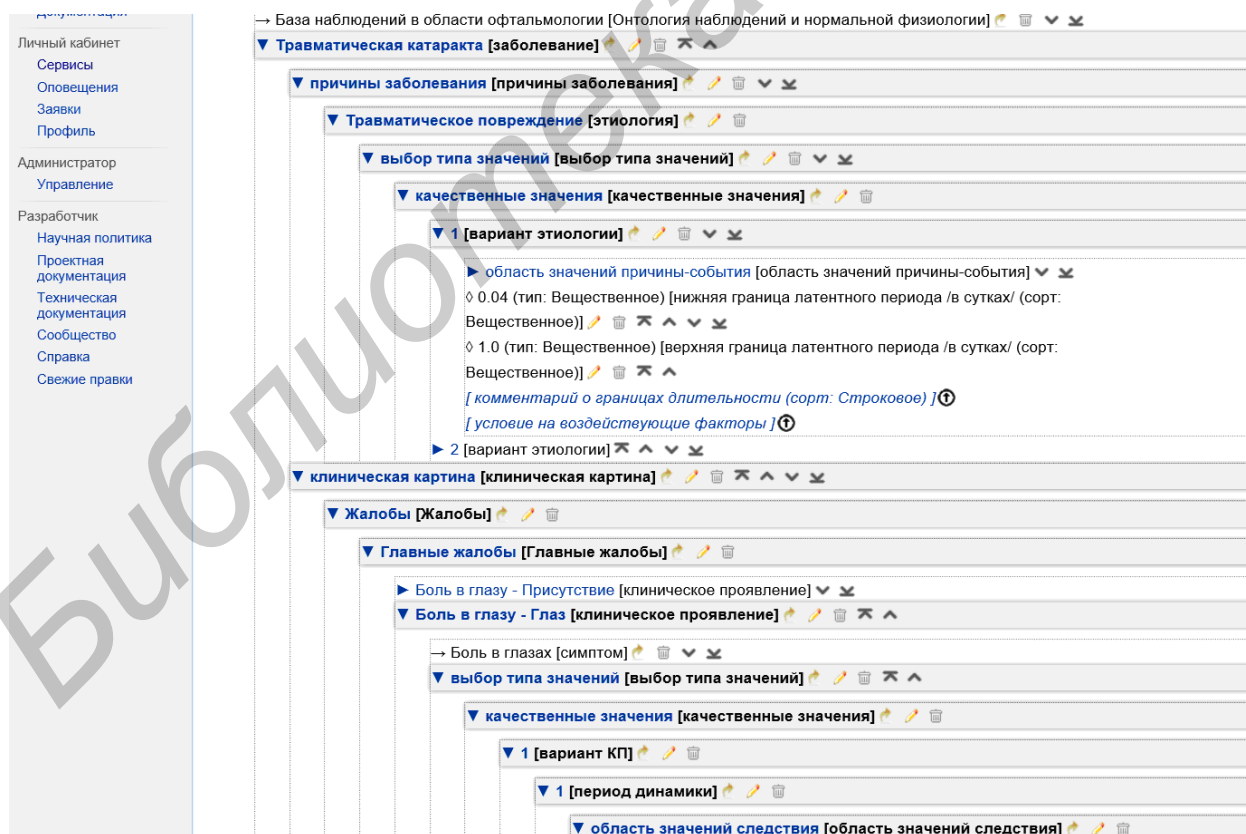


Рисунок 2 – Фрагмент описания клинической картины заболевания с учетом вариантов развития динамики признаков

Примером облачной платформы для реализации программ, обрабатывающих информацию в виде иерархических однородных семантических сетей, поддерживающих естественную для специалистов структуру информации и терминологии, является IASPaas [Грибова и др., 2011]. Технология IASPaas подразумевает создание специализированных решателей, обрабатывающих знания, представленные декларативно (в виде иерархических семантических сетей). Технология IASPaas поддерживает декларативность представления программных компонентов для создаваемых ИС. При разработке автоматизирующего медицинского комплекса каждый решатель реализуется чаще как самостоятельный облачный сервис, формирующий объяснение гипотез о решении задачи. Структура подаваемых на вход входных инфоресурсов, в том числе баз знаний, описывается в этой декларации. Замена версий подаваемой на вход базы знаний позволяет оперативно приспособлять решатель к уточняемым знаниям предметной области. Фрагмент примера входного информационного ресурса, соответствующего указанной в декларации метаинформации, и обрабатываемого этим решателем, показан на рис. 3. Пример декларации решателя задачи лечения, сформированной в среде IASPaas, показан на рис. 4.

При декларировании решателя указываются форматы обрабатываемой ими информации (входной и выходной). Входными для задачи

лечения будут истории болезни и используемые знания, а выходные – объяснение. Объяснение, в котором действительно заинтересован специалист в реальной практике, не должно давать свой вариант назначения ЛС пациенту. Врач нуждается в информации о том, какие конкретные ЛС не следует назначать данному пациенту с его «особенностями», потому что анализ противопоказаний десятков, а то и сотен известных ЛС с нужным фармакологическим воздействием связан с существенными затратами времени. Онтология объяснения решателя лечения формируется, исходя из этих обстоятельств.

Если решение задачи терапевтического лечения к получению списка ЛС, рекомендуемых пациенту с указанным диагнозом, к последующему устранению из него ЛС, не подходящих конкретному пациенту (с указанными возрастом и прочими «особенностями» организма); далее (особенно в случае сочетанной патологии) важно провести анализ множества рекомендованных ЛС с учетом их сочетаемости при одновременном приеме. Пример результата, сформированного решателем IASPaas, и сохраняемого для использования системой управления знаниями, показан на рис. 5.

Пример результата, сформированного подсистемой проверки соответствия очередной версии базы знаний - базе прецедентов, показан на рис. 6.

▼ Ампициллин [ЛС] ↗

- ◇ Ампициллин (тип: Строковое) [названиеЛС (сорт: Строковое)]
- ▼ противопоказания [противопоказания] ↗
 - ◇ бронхиальная астма (тип: Строковое) [бронхиальная астма (тип: Строковое)]
 - ◇ лейкопения (тип: Строковое) [лейкопения (тип: Строковое)]
 - ◇ почечная недостаточность (тип: Строковое) [почечная недостаточность (тип: Строковое)]
 - ◇ поллиноз (тип: Строковое) [поллиноз (тип: Строковое)]
- ◇ 0.1 (тип: Вещественное) [нижний возраст (сорт: Вещественное)]
 - ▶ Пенициллины [фармакологическая группа]
 - ▶ Взаимодействие с другими ЛС [Взаимодействие с другими ЛС]
 - ▶ показания [показания]
 - ▶ Фармакологическое действие [Фармакологическое действие]
 - ▶ торговые названия [торговые названия]

▼ Бенциклан [ЛС] ↗

- ◇ Бенциклан (тип: Строковое) [названиеЛС (сорт: Строковое)]
- ▶ показания [показания]
- ▼ противопоказания [противопоказания] ↗
 - ◇ Гиперчувствительность (тип: Строковое) [Гиперчувствительность (тип: Строковое)]
 - ◇ почечная недостаточность (тип: Строковое) [почечная недостаточность (тип: Строковое)]
 - ◇ дыхательная недостаточность (тип: Строковое) [дыхательная недостаточность (тип: Строковое)]

Рисунок 3 – Пример базы знаний о воздействиях ЛС

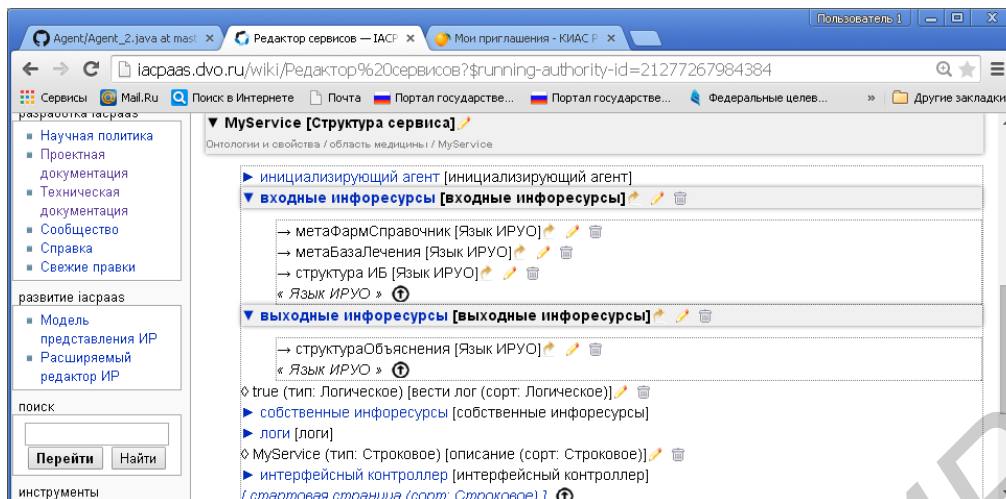


Рисунок 4 – Пример декларации решателя, реализующего решатель лечения

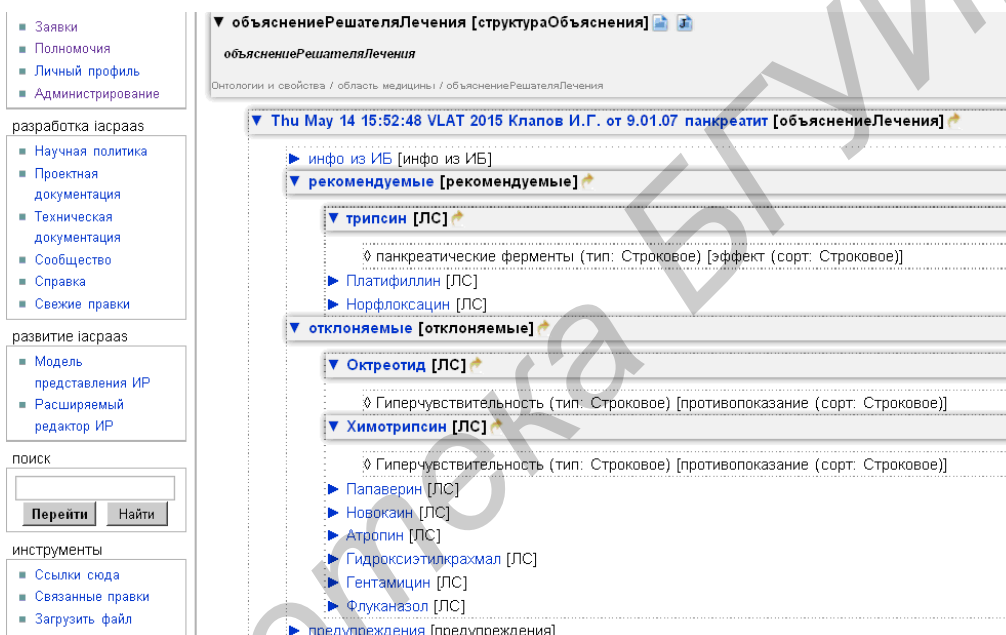


Рисунок 5 – Фрагмент объяснения назначения лечебных средств пациенту

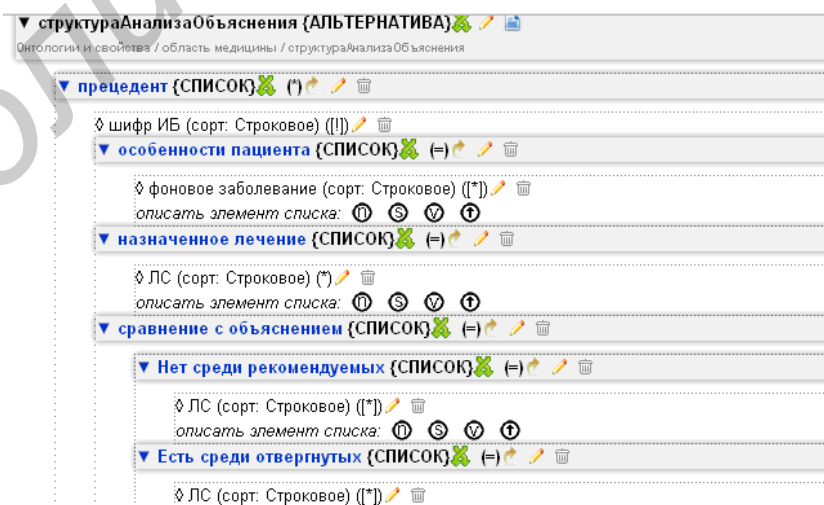


Рисунок 6 – Структура отчета о сравнении объяснения и правильного назначения

CONCEPTUAL DESIGN OF MAINTAINABLE SYSTEM OF INTELLIGENT ACTIVITY AUTOMATION

Kleschev A. , Shalfееva E.

*The Institute of Automation and Control
Processes, Vladivostok, Russia*

kleschev@iacp.dvo.ru

shalf@iacp.dvo.ru

This article is devoted to problem of quality control of knowledge bases used in decision-making process in daily intellectual activity. Technological stages of intellectual activity automation for some domain are allocated. The conceptual architecture of software system allowing to solve the set of interconnected problems of known classes using knowledge bases is proposed.

Introduction

The purpose of this research is development of the conceptual architectural design software of system of automation of medical activity - domain in which the set of the interconnected problems of known classes is solved and knowledge bases control is required.

Main Part

In the main part of the article the «place» of quality management in any intellectual activity automation is presented; identification of tasks in the domain is executed. The top-level architecture flow diagram (for medical activity automation) consisting of the subsystems connected by information streams with each other with storages of information and with an external environment is constructed.

The approach to implementation of a software of system of automation of medical activity taking into account requirements to ways of formation and use of knowledge bases and availability of all components of system for maintenance is offered.

Conclusion

The technological stages of automation of intellectual activity in concrete domain can be generalized to any area taking into account the principle of the system analysis and a modern paradigm of automation. The offered conceptual architecture of system of automation allows to solve a set of the interconnected intellectual problems of known classes where knowledge bases control is required. Feasibility of the offered approach is tested within the basic IACPaaS technology for problems of treatment of health for several medical specialization areas.

Заключение

Технологические этапы автоматизации интеллектуальной деятельности конкретной предметной области (медицинского первичного звена или отрасли) могут быть обобщены до произвольной области, поскольку выделены с учетом принципа системного анализа и современной парадигмы автоматизации. Предложенная концептуальная архитектура системы автоматизации позволяет решать при ее внедрении множество взаимосвязанных интеллектуальных задач известных классов, где требуется управление совокупностью баз знаний. Реализуемость предлагаемого подхода опробована в рамках базовой технологии IACPaaS для задач управления здоровьем по нескольким профилям (в частности, офтальмологические заболевания, заболевания ЖКТ).

Работа выполнена при частичной финансовой поддержке грантов РФФИ № 15-07-03193: «Облачные технологии обеспечения жизнеспособности и качества интеллектуальных систем» и № 14-07-00270 "Медицинские облачные сервисы"

Библиографический список

- [Гаврилова и др., 2000] Гаврилова, Т. А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб: Питер, 2000, 384 с.
- [Джексон, 2001] Джексон П. Введение в экспертные системы: Уч. пос. – М.: Издательский дом "Вильямс", - 2001. – 624 с.
- [Грибова и др., 2011] Грибова В.В., Клещев А.С., Крылов Д.А. и др. Проект IACPaaS. Комплекс для интеллектуальных систем на основе облачных вычислений. Искусственный интеллект и принятие решений. 2011. №1. С.27-35.
- [Жариков, 2008] Жариков О.Г., Литвин А.А., Ковалёв В.А. Экспертные системы в медицине. // Медицинские новости. 2008. №10. С. 15-18.
- [Клещев и др., 2015] Клещев А.С., М.Ю. Черняховская, Шалфеева Е.А. Особенности автоматизации интеллектуальной деятельности Научно-техническая информация. Сер.2., 2015, No. 1, с. 10–20.
- [Клещев и др., 2015a] Клещев А.С., Шалфеева Е.А. Онтология задач интеллектуальной деятельности // Онтология проектирования. Самара: "Новая техника", 2015. № 2 (16). с. 179-205
- [РусБИТех, 2013] НПО РусБИТех. Единое информационное пространство в здравоохранении РФ. 2011. <http://www.myshared.ru/slide/96415/> (Актуально на 30.11.2015).
- [Шалфеева и др., 2015] Шалфеева Е.А., Бочкарёв А.С. Концептуальная архитектура и подход к реализации медицинского сервиса, использующего управляемую базу знаний // Материалы Всероссийской конференции с международным участием "Знания – Онтологии – Теории" (ЗОНТ-2015). 6-8 октября 2015 г., Новосибирск, Институт математики им. С.Л. Соболева СО РАН. - Новосибирск: ООО «Технотрейд», 2015. Том 2. – С. 196-205.
- [Pressman, 2001] Pressman R.S. Software Engineering: Practitioner's Approach. Fifth edition. - McGraw-Hill Inc., 2001. - 860 p.



УДК 004.89

ТЕХНОЛОГИЯ РАЗРАБОТКИ РЕШАТЕЛЕЙ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНЫХ СЕРВИСОВ ДЛЯ ОБЛАЧНОЙ ПЛАТФОРМЫ IASPAAS НА ОСНОВЕ РАСШИРЯЕМОГО РЕДАКТОРА ОРГРАФОВ ИНФОРМАЦИИ

Москаленко Ф.М., Тимченко В.А.

Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток, Россия

philipmm@iacp.dvo.ru

vadim@dvo.ru

В работе описана специализированная технология разработки решателей задач интеллектуальных систем для облачной платформы IASPaas – на основе расширяемого редактора орграфов информации. Разработка ведётся с использованием инструментальных сервисов платформы. Технология направлена на снижение за счет ее применения трудоемкости разработки интеллектуальных систем как облачных мульти-агентных сервисов.

Ключевые слова: интеллектуальные системы; мульти-агентные системы; агентно-ориентированное программирование; облачные сервисы.

Введение

В настоящее время можно констатировать отсутствие принятой в качестве стандарта технологии, позволяющей разрабатывать жизнеспособные интеллектуальные системы (ИС), доступные широкому кругу пользователей. По-прежнему ощущается острая потребность в средствах разработки ИС, а также в повторном использовании компонентов ИС [Грибова и др., 2015a]. Для решения данных проблемных вопросов в области разработки и сопровождения жизнеспособных ИС в работах [Грибова и др., 2011, Gribova et al., 2013] предложена облачная платформа IASPaas. Она предназначена для обеспечения поддержки разработки, управления и удаленного использования прикладных и инструментальных мультиагентных облачных сервисов (прежде всего интеллектуальных) и их компонентов для различных предметных областей. Платформа предоставляет доступ:

- прикладным пользователям (специалистам в различных предметных областях) (в качестве реализации модели SaaS) – к прикладным сервисам;
- разработчикам (в том числе коллективам):
 - разработчикам прикладных и инструментальных сервисов и их компонентов (в качестве реализации модели PaaS) – к инструментальным сервисам;

- управляющим интеллектуальными сервисами (в качестве реализации модели SaaS)
- к сервисам управления.

Платформа IASPaas поддерживает:

- базовую технологию разработки прикладных и специализированных инструментальных (интеллектуальных) сервисов с использованием базовых инструментальных сервисов платформы, поддерживающих эту технологию [Грибова и др., 2015a, Тимченко и др., 2015];
- множество специализированных технологий разработки прикладных и специализированных инструментальных (интеллектуальных) сервисов, с использованием специализированных инструментальных сервисов платформы, поддерживающих эти технологии.

Специализированные технологии (по сравнению с базовой) и инструментальные средства их поддержки с одной стороны, как правило, накладывают определенные ограничения (по области применения и/или классу решаемых задач) на разрабатываемые с их использованием сервисы, а с другой, за счет учета специфики проблемной области и/или класса решаемых задач, обеспечивают более высокоуровневую поддержку разработки последних. Одной из таких специализированных технологий является технология разработки интеллектуальных сервисов на основе *расширяемого редактора орграфов*

информации [Грибова и др., 2015b, Грибова и др., 2015c]. Она позволяет создать сервис, в полной мере обладающий функциональностью и пользовательским интерфейсом редактора орграфов информации, а также дополнительными функциональными и интерфейсными возможностями. К ним относятся, например, контекстно-зависимое редактирование информационных ресурсов, текстовое, табличное или графическое представление их фрагментов и т.д.

В данной работе описывается технология разработки *интегрированных решателей задач* ИС на основе расширяемого редактора орграфов информации. *Интегрированный решатель задач* включает в себя множество взаимодействующих между собой посредством обмена сообщениями агентов, пользовательский интерфейс, обрабатываемые собственные информационные ресурсы и информационные ресурсы, представляющие метainформацию обрабатываемых входных и выходных информационных ресурсов – формальных параметров.

1. Общие сведения о технологии

Технология разработки интегрированных решателей задач интеллектуальных сервисов на основе расширяемого редактора орграфов информации в общем случае включает те же этапы, что и базовая технология интегрированных решателей задач [Грибова и др., 2015а, Тимченко и др., 2015]:

1. разработка информационных ресурсов (обрабатываемых решателем задач);
2. разработка решателя задач интеллектуального сервиса и его связывание с формальными параметрами, собственными информационными ресурсами и пользовательским интерфейсом;
3. разработка агентов решателя задач;
4. разработка шаблонов сообщений;
5. разработка пользовательского интерфейса.

Отличия описываемой специализированной технологии от базовой состоят в следующем:

- на этапе 1 дополнительно должен быть разработан информационный ресурс специального вида (лежащий в основе механизма расширения редактора орграфов информации), называемый *таблицей соответствий*;
- на этапе 2 в качестве *Корневого агента* и агента *Интерфейсный контроллер* решателя задач должен быть указан агент специального вида, а в качестве первого собственного информационного ресурса должна быть указана *таблица соответствий*;
- на этапе 3 агенты, к которым выполняется запрос непосредственно через таблицу соответствий, разрабатываются с учетом специальных требований;

- на этапе 5 создается *стартовая wiki-страница* интегрированного решателя задач, содержимое которой в обязательном порядке должно включать в себя фиксированную часть, формируемую по определенным правилам.

2. Разработка таблицы соответствий

Таблица соответствий t_G – это орграф информации, описывающий конкретное расширение редактора орграфов информации. Она сопоставляет вершинам орграфа метainформации G обращения к агентам платформы IACPaaS для формирования/модификации в процессе редактирования орграфов информации g_1, \dots, g_n ($n \geq 1$) фрагментов последних или для отображения пользовательского интерфейса, отличного от предоставляемого редактором орграфов информации. Орграф, описывающий метainформацию таблиц соответствий, представлен на рисунке 1. Здесь и далее нотация рисунков, изображающих орграфы метainформации и информации, согласована с нотацией, используемой в работах [Грибова и др., 2015b, Грибова и др., 2015c]. Начальная вершина, имеющая метку «Структура таблиц соответствий», закрашена серым цветом. Названия сортов и метки терминальных вершин, представляющих константные значения некоторого сорта, изображены на рисунке курсивным начертанием. Спецификаторы дуг имеют следующие символьные обозначения: «копия» – « \Leftarrow », «возможное отсутствие» – « \Leftarrow », «в точности один» – « \Leftarrow !», «непуста последовательность» – « \Leftarrow ^». Ограничители дуг имеют следующие символьные обозначения: «порождение» – «new», «ссылка» – «gef». Если из вершины выходит лишь одна дуга, то тип такого набора исходящих дуг – «список» – на рисунке не указывается. Вершины, представленные пунктирными прямоугольниками, принадлежат орграфу, представляющему другую метainформацию, в данном случае – орграфу метainформации, описывающему структуру агентов платформы IACPaaS. Символом Δ обозначен специальный тип вершин, семантика которого подробно описана в работе [Грибова и др., 2015c].

Для каждого обращения к агенту вершина «интерактивное» представляет логический признак, указывающий на необходимость получения информации от пользователя при работе блока продукции агента (при порождении и/или модификации значения вершины орграфа или подграфа информации g_i ($1 \leq i \leq n$), корневой вершиной которого она является). Если участие пользователя требуется, то данная вершина будет порождена при формировании таблицы соответствий t_G , в противном случае она порождаться не будет. Вершина «параметры» в t_G представляет собой корневую вершину произвольного орграфа метainформации, который описывает произвольное содержимое, которое будет передано стороннему агенту при обращении к нему.

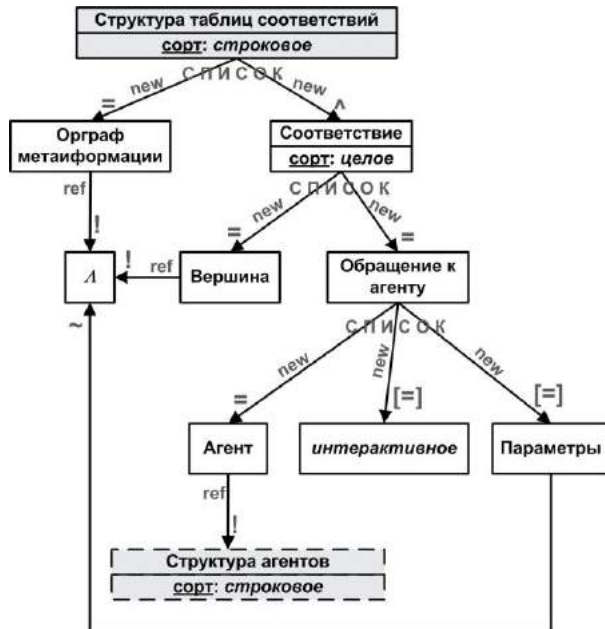


Рисунок 1 – Орграф метаинформации таблиц соответствий

Разработка информационного ресурса, как орграфа информации, представляющего конкретную таблицу соответствий, выполняется по схеме, описанной в работе [Грибова и др., 2015а]. В качестве метаинформации необходимо указать информационный ресурс «*Структура таблиц соответствий*».

3. Разработка решателя задач

Разработка решателя задач интеллектуального сервиса выполняется по схеме, описанной в работе [Грибова и др., 2015а]. Отличия, появляющиеся на этапе формирования в Фонде платформы IASaaS информационного ресурса (орграфа информации), представляющего декларативную спецификацию интегрированного (с информационными ресурсами, представляющими метаинформацию обрабатываемых информационных ресурсов – формальными параметрами, собственными информационными ресурсами и пользовательским интерфейсом) решателя задач, состоят в следующем:

- *Корневой агент* и агент *Интерфейсный контроллер* решателя задаются путем создания ссылки на информационный ресурс в Фонде платформы IASaaS, представляющий декларативную спецификацию агента «*Интерфейсный контроллер расширяемых редакторов инфоресурсов*»;

- обрабатываемые информационные ресурсы задаются путем создания ссылок на информационные ресурсы, представляющие их метаинформацию (при этом агент «*Интерфейсный контроллер расширяемых редакторов инфоресурсов*» выполняет отображение и модификацию первого (по порядку создания ссылок) информационного ресурса, в то время как остальные доступны для чтения и модификации сторонним агентам платформы IASaaS);

- в качестве первого собственного информационного ресурса необходимо указать (путем создания ссылки) информационный ресурс, представляющий конкретную таблицу соответствий;

- задать название стартовой wiki-страницы решателя задач на web-сайте платформы.

Интерфейс, формируемый агентом «*Интерфейсный контроллер расширяемых редакторов инфоресурсов*», открывает пользователю доступ ко всем функциональным возможностям редактора орграфов информации по редактированию информации в структурном виде. А посредством обработки таблицы сообщений «*Интерфейсный контроллер расширяемых редакторов инфоресурсов*» взаимодействует с описанными в ней агентами следующим образом.

При создании вершины v орграфа информации g , модификации/просмотре набора дуг, выходящих из вершины v , или модификации/просмотре ее значения, если вершина v соответствует вершине w орграфа метаинформации G , а в t_G из некоторой ее вершины u выходит дуга, входящая в вершину w , то выполняется обращение к заданному агенту (путем посылки сообщения) с соответствующими параметрами, сопоставленные вершине u таблицы соответствия t_G . «Неявными» параметрами, передаваемыми в сообщениях от агента «*Интерфейсный контроллер расширяемых редакторов инфоресурсов*» агентам, на которые сделаны ссылки из t_G , являются:

- при создании вершины v :
 - вершина v_0 орграфа информации g , являющаяся прямым предком вершины v ,
 - вершина w_0 орграфа метаинформации G , являющаяся прямым предком вершины w ,
 - дуга $w_0 \rightarrow w$ орграфа метаинформации G ;
- при модификации/просмотре набора дуг, выходящих из вершины v , или модификации/просмотре значения ее поля «значение»:
 - вершина v орграфа информации g ,
 - значение логического признака, указывающего на действие «модификация» или «просмотр».

Агент из t_G выполняет порождение или редактирование подграфа с корневой вершиной v или отображает пользовательский интерфейс. После завершения работы агента из t_G управление возвращается (также путем посылки сообщения) агенту «*Интерфейсный контроллер расширяемых редакторов инфоресурсов*». Если подграф g' орграфа информации g корневой вершиной которого является вершина, прототипу которой из G в t_G сопоставлено обращение к агенту, отображается (возможно, в отличном от структурного представлении) через интерфейс, генерируемый этим агентом, то также имеется возможность просмотреть g' через «стандартный» интерфейс, предоставляемый агентом «*Интерфейсный контроллер расширяемых редакторов инфоресурсов*».

4. Разработка агентов

Разработка каждого агента решателя задач интеллектуального сервиса выполняется по схеме, описанной в работе [Тимченко и др., 2015]. Специальные требования при разработке агентов, непосредственно взаимодействующих с агентом «Интерфейсный контроллер расширяемых редакторов инфоресурсов» через таблицу соответствий, состоят в том, что в множестве их блоков продукции в обязательном порядке должны присутствовать:

- блок продукции, выполнение которого инициируется сообщениями, сформированными по шаблону «Шаблон Сформировать подсеть»;
- блок продукции, выполнение которого инициируется сообщениями, сформированными по шаблону «Шаблон Обработать подсеть»;
- блок продукции, множество шаблонов выходных сообщений (сообщений, создаваемых в процессе выполнения данного блока продукции агента и рассылаемых адресатам после завершения его выполнения) которого должно содержать шаблон «Шаблон Результат формирования подсети»;
- блок продукции, множество шаблонов выходных сообщений (сообщений, создаваемых в процессе выполнения данного блока продукции агента и рассылаемых адресатам после завершения его выполнения) которого должно содержать шаблон «Шаблон Результат обработки подсети».

Последние два блока продукции могут быть суть одним и тем же блоком. Сообщения по шаблону «Шаблон Результат формирования подсети» («Шаблон Результат обработки подсети») должны быть посланы агенту «Интерфейсный контроллер расширяемых редакторов инфоресурсов» в ответ на сообщения по шаблону «Шаблон Сформировать подсеть» («Шаблон Обработать подсеть») от последнего. Перечисленные шаблоны сообщений присутствуют в Фонде платформы IASaaS априори.

Данные требования необходимо учесть на этапе формирования в Фонде платформы IASaaS информационного ресурса (орграфа информации), представляющего декларативную спецификацию агента, а также при написании его исходного кода.

5. Разработка пользовательского интерфейса

Разработка пользовательского выполняется по схеме, описанной в работе [Тимченко и др., 2015]. Специальное требование при разработке интерфейса, состоит в том, что при формировании содержимого *стартовой страницы* решателя, необходимо поместить в текст wiki-статьи в указанном порядке следующие элементы:

- вызов шаблона {{MaintenanceWork}};
- вызов решателя в виде *ui*-запроса к нему с единственным параметром *action*="навигация" <*ui*

solver = "предметная область/раздел/название решателя задач" *action* = «навигация» />;

- вызов шаблона {{IweCSS}} – для отображения решателя с таблицей стилей CSS расширяемого редактора орграфов информации.

6. Пример разработки интегрированного решателя задач на основе расширяемого редактора орграфов информации

В качестве примера опишем разработку решателя задач инструментального сервиса платформы IASaaS «Расширенный редактор агентов». Его функциональные возможности представляют собой объединение:

- функциональности решателя задач инструментального сервиса платформы IASaaS «Редактор агентов» [Тимченко и др., 2015] (предназначенный для формирования в Фонде платформы IASaaS информационных ресурсов (орграфов информации), представляющих декларативные спецификации агентов платформы);
- функциональности решателя задач инструментального сервиса платформы IASaaS «Генератор агентов» [Тимченко и др., 2015] (предназначенный для генерации заготовки исходного кода агента по его декларативному описанию);
- функциональности решателя задач инструментального сервиса платформы IASaaS «Загрузчик агентов» [Тимченко и др., 2015] (предназначенный для загрузки байт-кода агента (полученного в результате компиляции его исходного кода) в Фонд – в соответствующий информационный ресурс).

Орграф информации, представляющий таблицу соответствий для расширенного редактора агентов, представлен на рисунке 2. Начальная вершина, имеющая метку «Таблица соответствий для расширенного редактора агентов», закрашена серым цветом, а в квадратных скобках у вершин указаны метки соответствующих вершин из метайнформации.



Рисунок 2 – Орграф информации, описывающий таблицу соответствий расширенного редактора агентов

На рисунке 3 представлен оргграф информации, представляющий декларативную спецификацию решателя задач и его связей с другими компонентами, необходимыми для создания сервисов на основе данной спецификации.



Рисунок 3 – Оргграф информации, представляющий декларативную спецификацию интегрированного решателя задач «Расширенный редактор агентов»

В агент «Интерфейсный контроллер генератора агентов» добавляются блоки продукций для взаимодействия с агентом «Интерфейсный контроллер расширяемых редакторов инфоресурсов». Один из блоков вызывается при создании вершины «исходный код», другой – при её редактировании. При этом интерфейс и функциональность данных блоков продукций совпадает с интерфейсом и функционалом исходного блока продукций агента «Интерфейсный контроллер генератора агентов»: выводится список используемых шаблонов сообщений и кнопки для скачивания сгенерированных шаблонов исходного кода агента – см. рисунок 4.

Расширенный редактор агентов

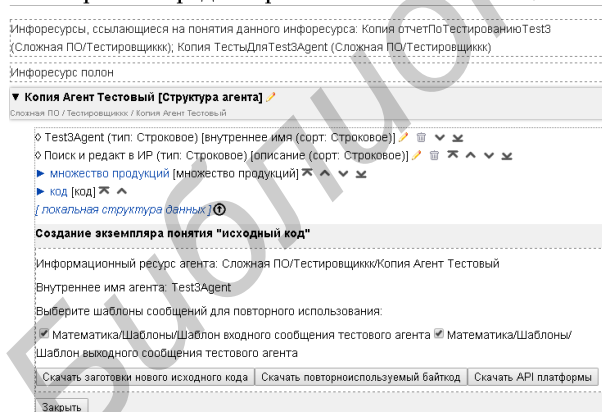


Рисунок 4 – Интерфейс генерации заготовки исходного кода агента

В агент «Интерфейсный контроллер загрузчика агентов» также добавляются блоки продукций для взаимодействия с агентом «Интерфейсный контроллер расширяемых редакторов инфоресурсов». Один из блоков вызывается при создании вершины «код», другой – при её редактировании. При этом интерфейс и функциональность данных блоков продукций

совпадает с интерфейсом и функционалом исходного блока продукций агента «Интерфейсный контроллер загрузчика агентов»: выводятся сведения об агенте, а также кнопки для начала загрузки подготовленного байт-кода агента и для проверки результата этой загрузки – см. рисунок 5.

Расширенный редактор агентов

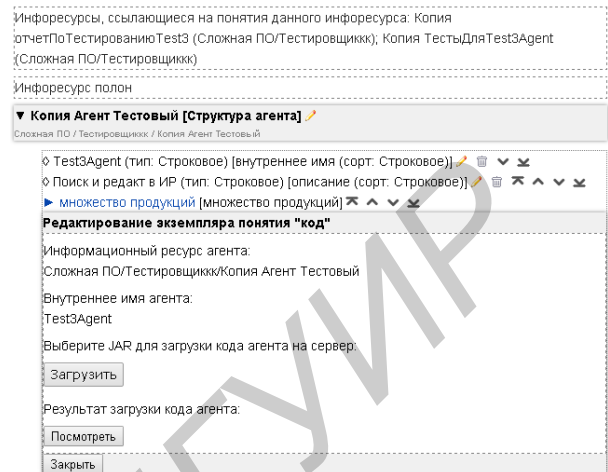


Рисунок 5 – Интерфейс загрузки байт-кода агента

Содержимое стартовой wiki-страницы интегрированного решателя задач «Расширенный редактор агентов» представлено на рисунке 6.

```

__NOCACHE__
{{MaintenanceWork}}
<ui solver="Платформа IACPaaS /
Разработка программных EX /
Расширенный редактор агентов"
action="навигация"/>
{{IweCSS}}
[[Категория:Сервисы]]

```

Рисунок 6 – Содержимое wiki-страницы интегрированного решателя задач «Расширенный редактор агентов»

На рисунках 4 и 5 продемонстрирован пример работы сервиса по редактированию агента «Копия Агент Тестовый», построенный на основе интегрированного решателя задач «Расширенный редактор агентов».

Заключение

В работе предложена специализированная технология разработки интегрированных решателей задач интеллектуальных сервисов для облачной платформы IACPaaS – на основе расширяемого редактора оргграфов информации. Использование данной технологии предпочтительнее использования базовой технологии в случаях разработки интегрированных решателей задач для сервисов, решающих задачи путем поэтапного (пошагового) формирования результирующего информационного ресурса (возможно, нескольких информационных ресурсов) – оргграфа информации в ручном, автоматизированном или автоматическом режиме. Преимущество состоит в том, что в этом случае функциональные и интерфейсные возможности редактора оргграфов информации (процесс сопровождения и развития которого выполняется независимо от разработчика решателя

задач) могут быть расширены возможностями контекстного редактирования, формирования и отображения фрагментов орграфа в форме представления, привычной для пользователя сервиса, формирования фрагментов орграфа на основе результатов вычислительных процедур и т.п.

С использованием предложенной технологии и базовых инструментальных сервисов платформы в настоящее время разрабатываются следующие интегрированные решатели (для построения облачных сервисов на их основе): структурный редактор виртуальных сред (предназначен для создания виртуальных сред по заданной в онтологии технологии в виде пошагового "мастера"), решатель для автоматизированного конструирования доказательств математических теорем и решатель в области медицинской диагностики.

Работа выполнена при частичной финансовой поддержке РФФИ (проект 14-07-00270, проект 14-07-00299 и проект 15-07-03193).

Библиографический список

[Грибова и др., 2015a] Базовая технология разработки интеллектуальных сервисов на облачной платформе IACPaaS. Часть 1. Разработка базы знаний и решателя задач / Грибова В.В. [и др.] // Программная инженерия. – №12, 2015, с. 3 - 11.

[Грибова и др., 2011] Проект IACPaaS. Комплекс для интеллектуальных систем на основе облачных вычислений / Грибова В.В. [и др.] // Искусственный интеллект и принятие решений. – 2011. – №1. – С.27-35.

[Gribova et al., 2013] A software platform for the development of intelligent multi-agent internet-services / Gribova V.V. [et al.] // Proceedings of the Distributed Intelligent Systems and Technologies Workshop (DIST'2013). – 1-4 July 2013. – St. Petersburg, Russia. – P.29-36.

[Тимченко и др., 2015] Технология разработки решателей задач интеллектуальных систем с использованием инструментальных сервисов облачной платформы IACPaaS / Тимченко В.А. [и др.] // Материалы V Международной научно-технической конференции OSTIS-2015. – Минск. – 2015. – С. 193 – 198.

[Грибова и др., 2015b] Двухуровневая модель сложноструктурированных информационных единиц, соответствующая метафоре анкетирования / Грибова В.В. [и др.] // НТИ. Сер. 2. - 2015. № 10. - С. 1-10.

[Грибова и др., 2015c] Модель порождения орграфов информации по орграфу метаинформации для двухуровневой модели сложноструктурированных информационных единиц / Грибова В.В. [и др.] // НТИ. Сер. 2. - 2015. №12. С. 26-38.

A TECHNOLOGY FOR DEVELOPMENT OF PROBLEM SOLVERS OF INTELLIGENT SYSTEMS FOR IACPAAS CLOUD PLATFORM ON THE BASIS OF EXTENSIBLE EDITOR FOR DIGRAPHS OF INFORMATION

Moskalenko Ph.M., Timchenko V.A.

*Federal State Budget Institution of Science
Institute for Automation and Control Processes
Far Eastern Branch of the Russian Academy of
Sciences, Vladivostok, Russia*

philipmm@iacp.dvo.ru

vadim@dvo.ru

The paper presents a specialized technology for development of multi-agent problem solvers of applied intelligent cloud services for the IACPaaS platform – on the basis of extensible editor for digraphs of information. The development is done with the use of instrumental services. The technology is put to reduce the labour-intensiveness of development and primarily of support for intelligent cloud services.

Introduction

Today one can find that there is no standard technology for development of viable intelligent systems which can be available for a wide range of users. There is an urgent need for tools for intelligent system development and for reuse of its components. In order to solve these tasks an IACPaaS cloud platform was introduced. It supports development, control and remote use of applied and instrumental services and their components for various domains.

The platform provides a basic technology for development of intelligent problem solvers and several specialized ones. One of the latter is a technology which uses extensible editor for digraphs of information as a basis.

Main Part

A technology for development of problem solvers of intelligent systems for the IACPaaS cloud platform on the basis of extensible editor for digraphs of information consists of the same steps as the basic technology. The differences are the following:

- the 1st step (development of information resources) includes the development of a special information resource (which is a basis of the mechanism for extending the editor for digraphs of information), which is called *table of correspondences*;
- at the 2nd step (development of the problem solver) the *Root agent* and *Interface controller* must point to a special agent (an *interface controller for extensible editors of digraphs of information*) and the first own information resource of the solver must be set to the aforementioned *table of correspondences*;
- at the 3rd step (development of agents for the problem solver) the agents which are addressed through the *table of correspondences* are developed by special rules;
- at the 5th step (development of user interface) a *starting wiki-page* for the integrated problem solver is created, whose content are also formed by special rules.

Conclusion

The described technology and basic instrumental services of the IACPaaS platform were used for creation of the following solvers: a structural editor for virtual environments (it is intended for creation of virtual environments and performs as a step-by-step master), a solver for automated construction of proofs for mathematical theorems and a solver for medical diagnostics domain.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

КОНЦЕПЦИЯ ИНСТРУМЕНТАЛЬНОГО КОМПЛЕКСА ДЛЯ ПОСТРОЕНИЯ СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИОННЫХ СИСТЕМ

Грибова В.В., Иванова А.В.

*Институт автоматизации и процессов управления
Дальневосточного отделения Российской Академии наук,
г. Владивосток, Россия*

`gribova@iacp.dvo.ru, 2395146@gmail.com`

В статье рассмотрены существующие подходы к построению комплексных систем защиты информации (СЗИ) и их недостатки. Сформированы принципы создания и концепция инструментария, обеспечивающего администратора платформы единой средой для управления разнородными СЗИ, а также рекомендациями по их выбору и настройке в соответствии с конфигурацией и особенностями конкретной информационной системы.

Ключевые слова: онтологии, информационные ресурсы, защита информации, безопасность информационных систем.

Введение

Обеспечение безопасности информационных систем (ИС) является одной из ключевых проблем в области информационных технологий. Здесь и далее под информационной системой будет пониматься любая информационная система, вне зависимости от типа платформы (локально-вычислительная сеть, автоматизированное рабочее место, распределенная платформа и т.д.). В настоящее время можно выделить два основных подхода к защите информации в ИС.

Первый подход заключается в использовании единого комплексного средства защиты, которое сочетает в себе функции каждого из отдельных узконаправленных компонентов. Такое решение обладает централизованным интерфейсом, что упрощает управление для администратора безопасности [Код Безопасности, 2015]. Второй, и наиболее популярный, подход заключается в использовании комплекса средств защиты информации (СЗИ) [ФСТЭК, 2014]. Отдельные компоненты специфичны, узконаправлены и предназначены для нейтрализации сходных угроз и могут быть разделены по типам: антивирусные средства, средства защиты от несанкционированного доступа, межсетевые экраны, криптографические средства защиты информации, системы обнаружения вторжений и др.

Однако оба подхода обладают рядом недостатков [PCMagazine, 2007]. В первом случае это:

- отсутствие совместимости с другими СЗИ;
- высокая стоимость конечного решения;
- невозможность отключения и последующей замены отдельных элементов на решения от другого разработчика (к примеру, отключение антивируса из комплекса и установка другого более репутационного антивируса или имеющего сертификат на класс выше);
- в случае невозможности эксплуатации (окончание срока действия сертификата, компрометация лицензионного ключа, сбой процедуры обновления и т.д.) нарушается работоспособность всех подсистем системы защиты информации.

Во втором случае это:

- отсутствие единого интерфейса управления;
- отсутствие механизмов оперативного контроля (необходимо напрямую вмешиваться в настройки СЗИ);
- проблемы совместимости СЗИ между собой (после установки одного СЗИ, другое может перестать функционировать, либо обнаруживаются конфликты совместного использования).

Кроме рассмотренных выше аспектов перед администратором безопасности возникает ряд новых проблем. Задача обеспечить информационную безопасность системы требует от него ряда специальных знаний и навыков для определения требований к системе, в том числе законодательных, выбора способов и средств защиты, их последующей настройки, поддержанию

непрерывного функционирования системы, разрешению конфликтных ситуаций, изучения возможностей улучшения системы и повышения ее эффективности [M. Rhodes-Ousley, 2014].

Подход к обеспечению защиты информационных систем с использованием комплекса СЗИ является наиболее популярным методом и используется в большинстве информационных систем. Исходя из вышесказанного, актуальным является создание инструментария, обеспечивающего администратора платформы единой средой для выбора и управления разнородными СЗИ, а также рекомендациями по их выбору и настройке в соответствии с конфигурацией и особенностями конкретной информационной системы. Такое решение обеспечит высокую эффективность технологий выявления, предупреждения и предотвращения атак и выполнение законодательных требований. Целью данной работы является описание общих принципов и концепции, обеспечивающих такую среду управления безопасностью.

1. Принципы построения инструментария

На основе анализа литературы, опыта работы в профилированной компании, приобретенных практических навыков, с учетом условий современной стадии развития технологий защиты информации, преимуществ и недостатков подхода, основанного на использовании набора разнопрофильных узконаправленных СЗИ, можно выделить основные принципы построения инструментального комплекса для организаций безопасности информационных систем:

1. Наличие единого интерфейса для управления разнородными СЗИ. Такой подход обеспечит администратора ИС средствами оперативного управления набором разнородных СЗИ без необходимости настройки последовательно каждого СЗИ. Единый интерфейс является необходимым условием централизованного сбора и систематизации сведений о текущем состоянии и конфигурации, полученных от различных средств защиты.

2. Инструментальный комплекс для управления безопасностью ИС должен поддерживать функцию выбора СЗИ. Выбор средств защиты информации в соответствии с конкретной конфигурацией ИС и требованиями к уровню ее защиты, настройка каждого средства, являются достаточно сложной задачей для любого администратора и часто требуют приглашения сторонних специалистов, как правило, из профилированных компаний, которые осуществляют функции консультирования, настройки системы безопасности, а также сопровождение ее в процессе эксплуатации. Однако такое решение является дорогостоящим, не всегда обеспечивает необходимый уровень эффективности и не позволяет оперативно решать конфликтные ситуации в случае их возникновения.

3. Наличие единого централизованного инструмента, обеспечивающего поддержку в актуальном состоянии информации, необходимой для принятия управленческих решений в конкретной ИС. Очевидно, что выбор набора СЗИ, их настройка в соответствии с требованиями безопасности для ИС требуют специальных знаний и навыков в области защиты компьютерных систем. Более того, это знания стремительно развиваются и изменяются. Соответственно, при их реализации в компьютерных системах необходимо учитывать эти аспекты. Во-первых, информация должна быть доступна и понятна экспертам в области информационной безопасности для анализа, исправления ошибок и неточностей, а также дальнейшего развития в связи с изменениями в предметной области. Во-вторых, необходимо создать централизованную "среду", аккумулирующую информацию о безопасности систем в едином информационном пространстве для того, чтобы обеспечить ее накопление, совместное развитие и использование для исключения "повторной разработки готовых решений". В-третьих, наличие централизованного хранилища, поддерживающего актуальность и постоянное обновление информации, позволит минимизировать человеческие ресурсы в существующих условиях кризиса высококвалифицированных специалистов в области информационной безопасности и позволит заметно сократить расходы на сопровождение комплекса средств защиты.

Для осуществления принципов предлагается:

Реализовать единую среду для хранения информационных ресурсов в области компьютерной безопасности и управления ими всем заинтересованным сообществом. Для обеспечения широкой доступности этих ресурсов, разместить их в "облаке", а также предоставить набор облачных сервисов (редакторов) для создания и управления информационными ресурсами в процессе жизненного цикла.

2. Концептуальная архитектура программного комплекса

Концептуальная архитектура инструментального комплекса для управления безопасностью ИС представлена на рисунке 1. Инструментальный комплекс состоит из: среды управления информационными ресурсами (на рисунке в верхнем блоке) и множества клиентских сред управления безопасностью информационной системы.

Среда управления информационными ресурсами состоит из интерфейса, структурного редактора онтологий, редактора информационных ресурсов, а также непосредственно информационных ресурсов и «Модуля выбора средств защиты».

Среда управления информационными ресурсами

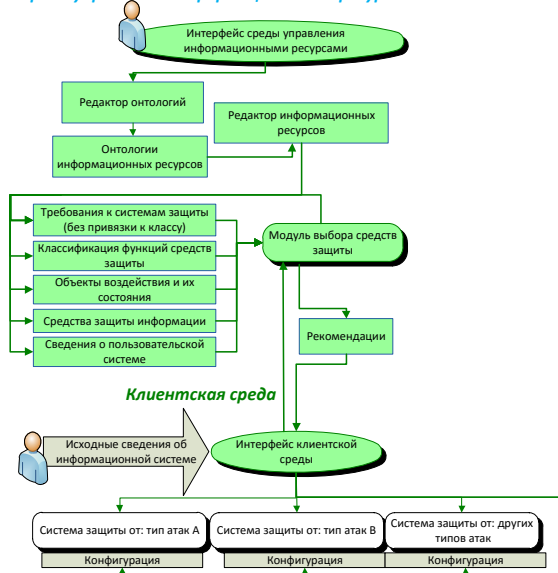


Рисунок 1 – Концептуальная архитектура инструментального комплекса для управления безопасностью информационных систем

Структурные редакторы предназначены для централизованного формирования и сопровождения (поддержки в актуальном состоянии) экспертами предметной области информации, которая используется в клиентских системах для помощи администратору в выборе и настройке СЗИ. Информационными ресурсами являются: «Требования к системам защиты (без привязки к классу)», «Классификация функций средств защиты», «Объекты воздействия и их состояния», «Средства защиты информации», «Сведения о пользовательской системе». Подробное описание информационных ресурсов дано в разделе «**Информационные ресурсы**».

В качестве средства реализации среды управления информационными ресурсами предлагается платформа IASPaas [Грибова В.В., и др. 2011]. Она представляет собой программно-информационный интернет-комплекс для обеспечения поддержки разработки, управления и удаленного использования прикладных и инструментальных (системных) мультиагентных облачных сервисов (прежде всего интеллектуальных) и их компонентов. Комплекс основан на технологии облачных вычислений и обеспечивает удаленный доступ конечным пользователям к интеллектуальным системам, а разработчикам и управляющим – к средствам создания интеллектуальных систем и управления ими. В состав платформы входит универсальный двухуровневый редактор для формирования данных и знаний. Он предназначен для описания метаинформации - онтологии информационного ресурса (первый уровень), по которой затем автоматически генерируется специализированный интерфейс для эксперта (второй уровень). Наличие универсального редактора, во-первых, не требует разработки специализированных редакторов для

необходимых информационных ресурсов, во-вторых, позволяет экспертам предметной области без инженеров знаний формировать и модифицировать информационные ресурсы в терминах онтологии. Таким образом, использование платформы IASPaas обеспечивает всю необходимую инфраструктуру для хранения и наполнения фонда информационных ресурсов.

Клиентская среда управления безопасностью информационной системы включает в себя Интерфейс управления клиентской среды, который выступает основным инструментом для взаимодействия Администратора ИС с различными СЗИ, а также Модулем выбора средств защиты. Через интерфейс клиентской среды Администратор получает сведения о состоянии подконтрольных СЗИ, определяет требования к клиентской ИС, ознакомление с информацией о минимальном наборе требуемых функций и СЗИ, реализующих их выполнение.

Рекомендации по настройке, выявление требований к набору средств защиты предусматривают централизованное управление и уведомления через Интерфейс управления клиентской среды. Функционирование этого блока основано на принципах кроссплатформенности, взаимозаменяемости компонентов выстроенной системы защиты, адаптации к таким изменениям,

«Модуль выбора средств защиты» предназначен для определения набора требований на основании входных параметрах клиентской ИС. Данный модуль также предусматривает определение набора средств защиты в случае, если в клиентской ИС уже функционируют средства защиты, которые удовлетворяют предъявленным требованиям и могли бы быть использованы.

Разрабатываемый инструментальный комплекс предусматривает определение требований к системе, определение набора допустимых средств защиты и выдачу рекомендаций по настройке конкретных программных модулей СЗИ.

3. Информационные ресурсы

Основными информационными ресурсами (ИР) являются :

1) «Сведения о пользовательской системе» содержит информацию об определяющих характеристиках клиентской информационной системы. Позволяет пользователю, описать конфигурацию о конкретной ИС в формализованном виде для дальнейшего использования. При наличии в клиентской системе предустановленных (закупленных) СЗИ, может ссылаться на ИР «Средства защиты информации» для учета этой информации и исключения избыточности рекомендаций. Фрагмент ИР изображен на рисунке 2. Взаимодействие ИР между собой будет рассмотрено ниже.

В примере указано три возможных варианта обрабатываемой информации и их цифровое обозначение: «1.1. Персональные данные», «1.2. Конфиденциальная информация», «1.3. Для служебного пользования». Также имеются поля, которые могут быть заполнены пользователем без предложенных вариантов: «Количество рабочих мест» - любая целочисленная количественная характеристика.

2) «Требования к системе защиты (без привязки к классу)» включает в себя перечень законодательных и пользовательских требований к защите информационной системы. Используется при определении требуемого к реализации набора функций защиты клиентской системы на основе конфигурации информационной системы.

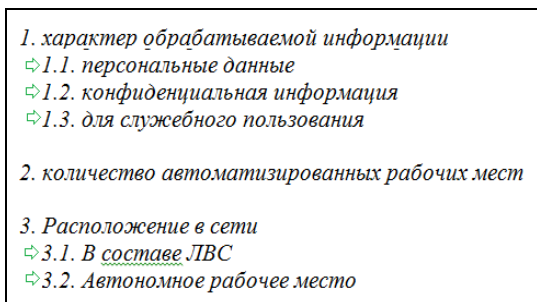


Рисунок 2 – Фрагмент ИР «Сведения о пользовательской системе»

3) «Классификация функций средств защиты», изображена на Рис. 3, представляет собой многоуровневую структуру и включает в себя классификацию функций СЗИ, определяемых нормативными документами [ФСТЭК, 2015], а также ряд дополнительных функций, которые разработчики встраивают в программное обеспечение и декларируют в «Руководствах по эксплуатации» своих программных продуктов. На рисунке 4 различными маркерами (треугольниками справа) обозначены функции средств защиты. Каждое средство защиты закодировано своим маркером. Если указанная функция имеется в средстве защиты, то напротив нее проставляется соответствующий ему маркер.

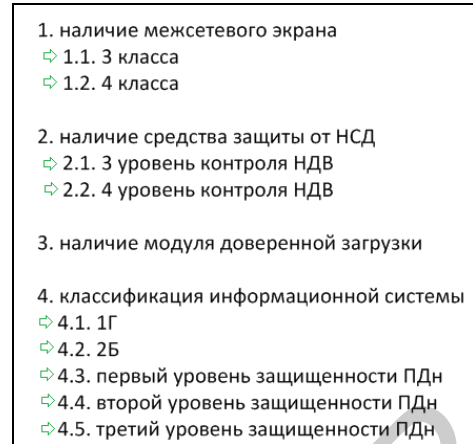


Рисунок 3 – Фрагмент ИР «Требования к системам защиты (без привязки к классу)»

Такое обозначение позволяет определить полный набор функций средства защиты, их иерархию, а также определить, являются они основными в средстве защиты или дополнительными, основываясь на глубине классификационного «дерева» конкретной функции.

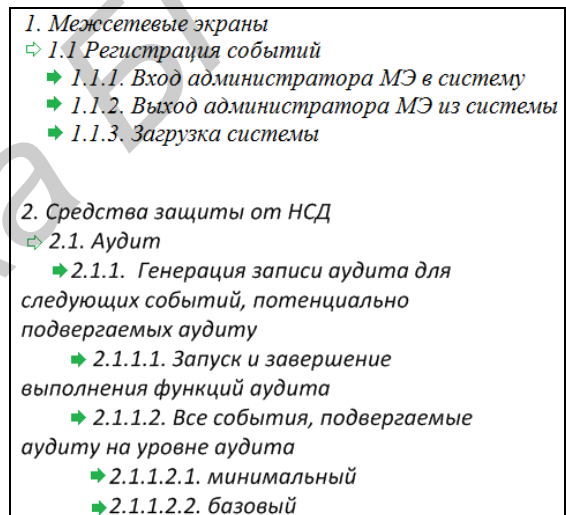


Рисунок 5 – Фрагмент ИР «Классификация функций средств защиты»

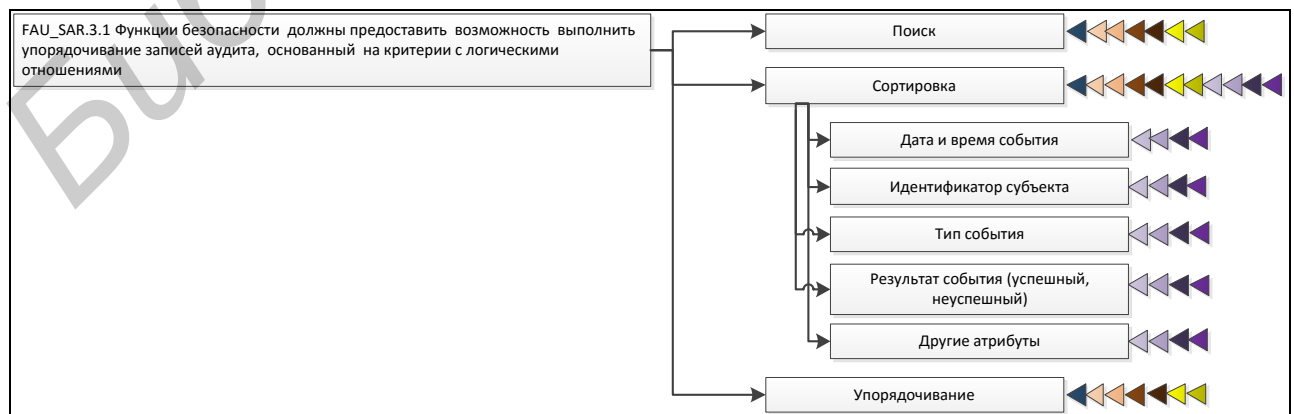


Рисунок 4 – Иерархическая классификация функций средств защиты в соответствии с нормативно-законодательными актами

ИР «Классификация функций средств защиты» содержит сводную информацию о всех функциях средств защиты без привязки к конкретному профилю (типу СЗИ). Фрагмент базы данных изображен на рисунке 5.

В качестве примера рассмотрена одна из функций межсетевых экранов: «1.1. Регистрация событий», а также возможные события регистрации: «1.1.1. Вход администратора МЭ в систему», «1.1.2. Выход администратора...», 1.1.3. «Загрузка системы». У указанных событий также могут быть параметры регистрации, такие как время, предъявленный идентификатор, а также другие атрибуты регистрации, которые в рамках примера не рассматриваются для межсетевых экранов. Вторым типом рассмотренных наборов функционала является функционал «2. Средства защиты от НСД». В примере изображена функция, которая имеет множество параметров и детализирована более глубоко, чем «1.1. Регистрация событий» для «1. Межсетевые экраны».

4) «Средства защиты информации» содержит информацию о наборе функций конкретных СЗИ, представленных на рынке сертифицированных продуктов для обеспечения информационной безопасности (Kaspersky Endpoint Security, Dr.Web, SecretNet, TrustAccess и др.). Используется для определения перечня СЗИ, необходимых к установке в клиентской системе, для определения набора функций, которые фактически будут активны в системе с привязкой к конкретным СЗИ.

Структура БД имеет порядковую нумерацию, в качестве вложений используются цифровые обозначения из других БД, в частности из БД «Классификация функций средств защиты» вместо текстового наименования функции используется ее цифровой код. Так, в п. 1.1. тип СЗИ указывается в соответствии с БД «Классификация функций средств защиты» (Пример: средство защиты от НСД; антивирусное средство или межсетевой экран), в п. 1.2. перечисляются только цифровые обозначения имеющихся функций (см. Рисунок 5).

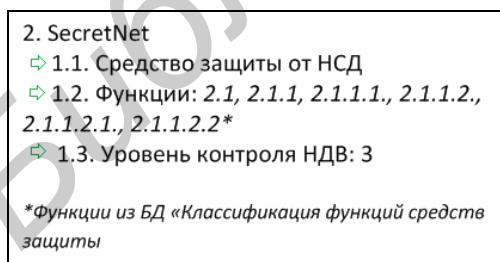


Рисунок 6 - Фрагмент ИР «Средства защиты информации»

Стоит отметить, что часть функций, встроенных в качестве обязательных в одно СЗИ, может присутствовать в качестве дополнительных функций (на усмотрение разработчика) в СЗИ другого профиля защиты.

5) «Объекты воздействия и их состояния» включает в себя перечень возможных объектов

воздействия, их атрибуты и допустимые значения функций СЗИ над этими объектами. Фрагмент ИР изображен на рисунке 7.

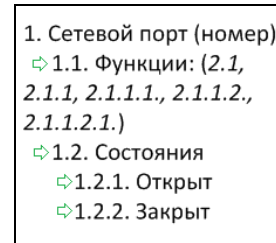


Рисунок 7 - Фрагмент базы данных «Объекты воздействия и их состояния»

ИР содержит объекты воздействия, а в качестве вложений указываются их допустимые состояния. Например, на рисунке 7, указано, что для объекта воздействия «1. Сетевой порт (<номер>»), доступны следующие управляющие им функции (см. рисунок 5): «2.1. Аудит», и далее по дереву вложенных функций можно увидеть, что доступен только «2.1.1.2.1. минимальный» уровень аудита. Это значит, что в расширенном уровне аудита указанный порт не контролируется. Исходя из того, что аудит, это лишь операция чтения, «1.2. Состояние» с детализацией «1.2.1. Открыт» можно интерпретировать как включенный аудит для данного порта. В п. 1.1. перечисляются только цифровые обозначения имеющихся функций (см. Рисунок 5).

В результате поэтапного взаимодействия всех ИР (Рис. 8) инструментального комплекса управления безопасностью будет сформировано следующее:

- Требования к системе;
- Требования к включенным функциям;
- Перечень средств защиты, которые требуется установить, с указанием возможных вариантов, а также с учетом эффективности тех средств защиты, которые уже предустановлены;
- Параметры средств защиты, которые необходимо активировать.

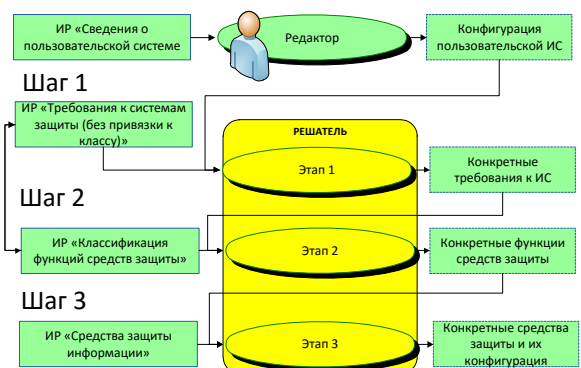


Рисунок 8 - Алгоритм работы с ИР

Заключение

В настоящей статье рассмотрены предпосылки формирования необходимости обеспечения информационной безопасности в информационных системах, существующие подходы к созданию

систем защиты (комплекс средств защиты и монолитные решения), достоинства и недостатки указанных подходов. Обосновывается необходимость создания архитектуры, объединяющей качественные средства защиты информации в единую систему. Формируются основные принципы создания такой архитектуры. Предложена концептуальная архитектура среды управления информационными ресурсами, позволяющая осуществить системное описание информационной системы, выбор средств защиты и их последующую настройку в соответствии с заданными требованиями. Описаны основные компоненты указанных сред и их функции.

На сегодняшний день в рамках решения данной задачи разработан набор описанных выше информационных ресурсов, начата реализация программных компонентов инструментального комплекса на платформе IACPaaS.

Благодарности

Работа выполнена при частичной финансовой поддержке РФФИ, грант 16 -07-00340, программы "Дальний Восток", грант 262-2015-0069.

Библиографический список

[Код Безопасности, 2015] Код Безопасности. Security Studio Endpoint Protection. Сертифицированная защита компьютера от сетевых вторжений, вредоносных программ и спама // Код безопасности - 2015. [Электронный ресурс: http://www.securitycode.ru/products/security_studio_endpoint_protection/]

[ФСТЭК, 2014] Федеральная служба по техническому и экспортному контролю (ФСТЭК России), «Меры защиты информации в государственных информационных системах», Москва, 2014, pp. 5-15.

[PCMagazine, 2007] PCMagazine. Курс лекций «Вирусы и борьба с ними» // «Лаборатория Касперского» - 2007. [Электронный ресурс: http://www.pcmag.ru/elearning/course/lesson.php?COURSE_ID=10&ID=62]

[M. Rhodes-Ousley, 2014] M. Rhodes-Ousley, Information Security. The Complete Reference, Second Edition. Silicon Valley: California, 2014, pp.578–595.

[Грибова В.В., и др. 2011] Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Смагин С.В., Тимченко В.А., Тютюник М.Б., Шалфеева Е.А. Проект IACPaaS. Комплекс для интеллектуальных систем на основе облачных вычислений // Искусственный интеллект и принятие решений. – 2011. – №1. – С.27- 35

[ФСТЭК, 2015] Федеральная служба по техническому и экспортному контролю. Документы по сертификации средств защиты информации и аттестации объектов информатизации по требованиям безопасности информации // ФСТЭК России – 2015. [Электронный ресурс] - Режим доступа. - URL: <http://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty>.

CONCEPT OF A TOOL FOR CREATING OF INFORMATION SECURITY SYSTEMS

Gribova V.V., Ivanova A.V.

*Institute of Automation and Control Processes
FEBRAS, Vladivostok, Russia*

gribova@iacp.dvo.ru,

2395146@gmail.com

The article describes the existing approaches to creation of integrated information security systems (ISS) and their shortcomings. Principles of a tool development are presented. It provides the administrator of an information system an unified interface for managing different ISS, as well as recommendations for their selection according to the configuration and characteristics of the information system.

Introduction

The security of information systems (IS) is one of the key issues in the field of information technology. There are two main approaches to information security of IS. The first approach is based on using a single tool, which combines functions of some different tools, the second and the most popular one is based on using a combination of information security tools. Two types of approaches have both advantages and disadvantages. The aim of the paper is to describe the basic principles of the tool and their components. The tool is based on the second approach to ISS.

Main Part

The tool consists of the information resources control system and a set of client systems for IS security control.

The information resources control system includes structural editors for ontologies and information resources creation, as well as the information resources and a tool for choosing ISS.

A client systems for IS security control has an interface for receiving information about state of different ISS and interaction with the choosing ISS tool.

The information resources are: Requirements to ISS, Classification of ISS functions, Impacted objects and their conditions, Information security systems, and Client system Information.

Conclusion

In our research we proposed the tool for determination the requirements to ISS, as well as the guidance for choosing a set of ISS and their settings based on configuration of the IS.

We completed the development of the information resources on the IACPaaS platform, and currently implementing the program components.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЙ ДЛЯ СОЗДАНИЯ СЦЕНАРИЕВ ОБРАБОТКИ ДАННЫХ

Глоба Л.С.

*Национальный технический университет Украины «Киевский политехнический институт»,
г. Киев, Украина*

lgloba@its.kpi.ua

Предложен метод формирования сценариев обработки данных, основанный на онтологическом описании как информационных, так и вычислительных ресурсов (web-сервисов). Поиск и принятие решения о возможности включения web-сервиса в состав сценария выполняется на основе оценки семантической близости, которая учитывает входы и выходы операций, множество типов параметров и «шаблоны сервисов». Установление порядка выполнения сервисов в рамках одного сценария выполняется с помощью упорядоченного дерева, порядок обхода вершин которого соответствует порядку вычислений. Такой метод позволяет автоматизировать процесс формирования сценариев обработки больших объемов данных, расположенных в среде Интернет.

Ключевые слова: сценарий обработки данных, web-сервис, метаграфы, онтологические модели.

Введение

Проблематика обработки значительных объемов информации, накапливаемых в различных сферах деятельности: производстве, здравоохранении, торговле, государственном управлении и т.д., приобретает все большую остроту и определяется сегодня термином Big Data (большие данные). Ожидается, что развитие и начало широкого использования всех источников больших данных инициирует проникновение технологий больших данных как в научно-исследовательскую деятельность, так и в коммерческий сектор, а также сферу государственного управления.

В качестве определяющих характеристик для больших данных отмечают «четыре V»: объём, скорость (в смыслах как скорости прироста, так и необходимости высокоскоростной обработки и получения результатов), многообразие (в смысле возможности одновременной обработки различных типов структурированных, полуструктурированных и неструктурированных данных с точки зрения двух аспектов: синтаксиса и семантики), стоимость, которая особенно связана с их коммерческой ценностью [Business Information Systems, 2014].

Современные требования к огромным объемам уже накопленных данных состоят в переходе к поддержке принципа:

ДАННЫЕ → ИНФОРМАЦИЯ → ЗНАНИЯ.

Данные рассматриваются как массивы несвязных разнородных территориально-

распределенных данных, по ним можно лишь выполнить сбор статистики или осуществить элементарный поиск.

Информация рассматривается как набор данных агрегированных по определенным характеристикам или требованиям и представляет собой предварительно обработанные и обобщенные данные, классифицированные по определенному признаку или объединённые по определённой тематике. Получение информации из данных выполняется с использованием методов аналитической обработки данных, обобщения, группирования, математических моделей и методов, методов классификации, структуризации и систематизации, аналитической обработки данных, обобщения, системного и сценарного анализа и т.д.

Знания рассматриваются как набор данных, сгенерированных на основе имеющихся данных. Получение знаний из данных и информации осуществляется на основании методов экспертных оценок, интеллектуальной обработки информации и интеллектуального анализа данных, прогнозирования, моделей представления знаний, подсистем устойчивого развития и других.

В качестве научного базиса при создании многих интеллектуальных информационных систем, призванных обрабатывать значительные объемы информации (транзакционные системы, системы поддержки принятия решений, порталы знаний и т.п.) выступают следующие математические и интеллектуальные модели и соответствующие им методы: семантические сети, онтологическая

модель, нечеткие базы знаний, метаграфы, метаописания, сценарии динамической обработки заданий пользователя. Создание таких моделей и методов является непростой задачей и требует, как знаний в определенных областях деятельности, так и понимания общесистемных взаимосвязей. Инженерная и научная сферы деятельности имеют особенно острую потребность в средствах извлечения знаний, важнейшими среди которых есть сценарии динамической обработки заданий пользователя. С точки зрения задач инженерных областей или научных исследований очевидно, что выполнение конкретных расчетов в процессе анализа требует решения ряда определенных подзадач, для чего необходимо не только описать доступные пользователю расчетные задачи, но и установить связи между конкретными сервисами обработки (набором сервисов) для реализации конкретных рабочих процессов (workflow (WF)) в рамках общих рабочих процессов и связей сервисов с информационными ресурсами, которые необходимы для выполнения расчетов. Важным фактором является оптимизация таких связей. Кроме того, в зависимости от параметров, заданных пользователем, тематикой и особенностями выполняемых расчетов, последние могут выполняться с помощью разных алгоритмов. В связи с этим при построении реальных научных и инженерных задач используют их декомпозицию на подзадачи в зависимости от направленности общей задачи и параметров, необходимых для выполнения расчета [Новогрудська, 2015]. Подрасчеты динамически komponуют в общую задачу, при этом одна подзадача может использоваться в ряде разных общих задач. Это обуславливает потребность учета и описания логики связности расчетных задач и их корректной интеграции в информационное пространство портала. Таким образом, при построении единой информационной среды в инженерных и научных сферах деятельности недостаточно использования семантических или онтологических моделей, в этом случае возникает ряд дополнительных проблем:

- определение механизма связывания информационных элементов с функциональными, а также формирование сложного расчета «на лету» на момент его выполнения в зависимости от текущих условий;
- описание характеристик функциональных и информационных элементов в соответствии с предметной областью;
- описание последовательности взаимодействия функциональных элементов для реализации глобального расчета.

Объективными трудностями при создании эффективных моделей и методов обработки больших объемов информации являются недостаточная систематизация и структуризация информации, неоднозначность метаописаний в онтологиях многих научных и инженерных порталов в случае, если информация разбросана в Интернет, невозможность автоматизированного

создания workflow для извлечения знаний, хранящихся в распределенных информационных хранилищах.

В статье предлагается с целью систематизации больших объемов данных и повышения производительности их обработки объединить три важные группы математических моделей и методов: онтологические модели как средство описания наборов данных в сети Интернет, теорию метаграфов и нечеткую логику как средства обработки больших объемов информации и извлечения знаний из них.

1. Модели и методы представления и обработки данных

Ключевая тенденция развития Интернета заключается в переходе от документов машиночитаемых к документам, которые машина понимает, а это требует наличия средств понимания смысла при обработке больших объемов данных. Процесс такой обработки требует автоматизированного выполнения ряда этапов: создания онтологий, связывающих ресурсы под конкретные задачи на основании онтологических моделей; формирования поисковых запросов «на лету» с учетом возможной неточности метаописаний и их эффективного исполнения на основании автоматизированных рабочих процессов (сценариев обработки информации, учитывающих альтернативы). Структура программных средств, построенных с использованием онтологических моделей состоит из следующих компонент:

- онтология,
- коллекционер онтологической информации о ресурсах,
- конструктор запросов,
- формирователь ответов.

Для более четкого понимания и анализа связей между элементами онтологической модели удобно использовать графовые структуры, чтобы визуализировать связи между отдельными компонентами. Это облегчает процесс структурирования данных в процессе их обработки. Кроме того, графы используют не только в качестве основы для визуализации в процессе получения вывода, но они также могут служить базисом алгебраических операций, которые позволяют более строго проводить обработку больших объемов информации. Однако, отображение нечеткости метаописаний свойств отдельных объектов сложно отобразить в виде простой графовой структуры, поэтому для этих целей удобно использовать специальный вид графов, который называют метаграфами [Amit Basu, 2008].

1.1. Онтологические модели

Онтологические модели можно использовать, для:

- моделирования доменов программных сред;
- формирования автоматизированных рассуждений и получения выводов;

- автоматического программирования и удовлетворения спецификаций;
- построения программных систем, основанных на моделях;
- технологий семантики (онтологий и семантического web).

Особенностью онтологических систем является способность поддерживать систематизацию знаний и информационных ресурсов моделируемых областей знаний, обеспечивать содержательный доступ к ним. В рамках общей онтологии выделяют специфические онтологии и проектирование общей онтологической модели, которую создают путем доопределения и развития базовых онтологий. Онтологии включают как описание предметной области (ПО), так и описание релевантных ей ресурсов. Описание предметной области включает совокупность терминов и отношений, семантически значимых для данной предметной области, а также правил, согласно которым можно строить утверждения об элементах ПО. Описание ресурса сети включает список ссылок на страницы и сайты, а также описание связей между ними.

Основные преимущества использования онтологических моделей в качестве одной из составляющих научного базиса:

- Онтологии позволяют сделать систему предметно-ориентированной, благодаря чему пользователь может задавать поисковый запрос в хорошо известных ему терминах предметной области.
- Онтологии являются новым интеллектуальным средством для поиска ресурсов в сети Интернет, новыми методами представления и обработки знаний и запросов.
- Онтологии способны точно и эффективно описать семантику данных для определенной предметной области и решить проблему несовместимости и противоречивости данных.
- Онтологии владеют собственными средствами обработки (логического вывода) и соответствующими моделями и методами семантической обработки данных.

1.2. Метаописания

Основным элементом онтологической модели предметной области знаний являются метаописания ее объектов, именно с помощью метаописаний и устанавливаются связи между объектами.

Метаописание – объект рассматриваемой ПО, который служит для определения элементов системы, информационных и вычислительных ресурсов, учитывает их смысловое значение. Метаописания позволяют однозначно идентифицировать элементы и характеризуются двойкой: $M_i = \langle A, V \rangle$, где A – атрибут, V – значение атрибута.

Синтаксис обозначения метаописаний: порядковый номер метаописания - номер элемента, который характеризуется метаописанием, название атрибута, отражающее его смысловое

значение и собственно значение атрибута. Формат записи:

$$\text{Атрибут} = \text{значение } M_i(E): \text{Atr} = \text{"Value"}.$$

Для порталов знаний выделено несколько видов метаописаний:

- Стандартные метаописания – определяются/строятся согласно стандарту дублинского ядра, и описывают основные характеристики элементов.
- Специфические метаописания – метаописания связанные с предметной областью.
- Метаописания, устанавливающие связь, – используются для определения связи элемента с другими элементами, например, информационными ресурсами.

Для установления связей между метаописаниями используют правило объединения, которое формулируется: «Если два элемента имеют одинаковые метаописания, то они связаны». Существует два вида связности: 1) элементы компонуется для объединения в общий элемент либо 2) один из элементов включается как подэлемент во второй.

Сравнение элементов происходит по значениям атрибутов, т.е. метаописаний. Вначале происходит сравнение значений, потом проверка на равенство, далее проверка самих элементов для организации их в более общий элемент. Однако, такое сравнение не всегда является точным или хотя бы числовым, поэтому для нахождения объектов по их метаописаниям и принятия решения о принадлежности элементов к тем или иным классам требуется решение задачи классификации на нечеткой области значений.

1.3. Обнаружение web-сервисов в мульти-онтологической среде

С целью извлечения знаний используют сценарии обработки заданий пользователя, называемые workflow, которые представляются как объединение web-сервисов, хранящихся в распределенных информационных хранилищах порталов знаний, в цепочку последовательно-параллельных вычислений. Для такой реализации workflow и последующей обработки полученных данных формируют семантические отношения между процессами (web-сервисами), чтобы отразить сходства и различия между запросами конечных пользователей и моделями, описывающими web-сервисы. Обнаружению web-сервисов в среде Интернет посвящены работы [Larysa Globa, 2015]. Однако, в предлагаемых подходах имеется ряд минусов. Прежде всего, это проблема неоднозначности языковых конструкций, поэтому web-службы поиска используют метаописания онтологических понятий, написанные исключительно с помощью инструментария, такого как: язык DAML (DARPA Agent Markup Language), описания в виде RDF-графов (Resource Description Framework), язык LARKS (Language for Advertisement and Request for Knowledge Sharing),

описания для целей машинного обучения LSD (Learning Source Descriptions), основанные на распределении вероятностей экземпляров различных заранее определенных онтологических моделей. Несмотря на значительное число имеющихся решений, остается существенным влияние человеческого фактора и, кроме того, не поддерживаются описания для входов и выходов web-сервисов. Для улучшения релевантности обнаружения web-сервисов в среде Интернет предлагается использовать семантическую близость, которая учитывает входы и выходы операций, например, множество типов параметров и «шаблоны сервисов».

Подход к поиску web-сервисов в нескольких онтологических средах состоит из трех последовательных этапов:

1. Создание «шаблона сервиса», базируясь на запросе пользователя;
2. Сравнение «шаблона сервиса» с несколькими web-сервисами, которые были определены как сервисы-кандидаты;
3. Возвращение web-сервисов, удовлетворяющих минимально приемлемой оценке сходства с запрошенными пользователем в упорядоченном списке.

Семантический шаблон сервиса описывает запрос пользователя. Это позволяет конечному пользователю определить набор необходимых операций, их свойств, входов и выходов. Шаблон сервиса не имеет конкретной реализации, потому что рассматривается как промежуточная абстракция - прокси web-сервис.

Шаблон сервиса может быть определен как:

$$ST = \langle N_{ST}, D_{ST}, OP_{ST}(N_{OP}, D_{OP}, O_{OP}, I_{OP}) \rangle, \quad (1)$$

где N_{ST} - имя web-сервиса,
 D_{ST} - текстовое описание web-сервиса,
 OP_{ST} - множество операций web-сервиса.

Каждая из операций web-сервиса в свою очередь определяется с помощью: N_{OP} - имени операции, D_{OP} - текстовое описание операции, O_{OP} и I_{OP} - входных и выходных параметров операции.

Шаблон сервиса сравнивается с набором сервисов-кандидатов (CS) - шаблонов сервисов, полученных с помощью анализа web-сервисов из предварительно определенного набора web-сервисов. При сравнении используют для оценки синтаксическое и семантическое сходство, определяемое как:

$$\theta(ST, CS) = \frac{w_H \cdot H(ST, CS) + w_\Phi \cdot \Phi(ST, CS)}{w_H + w_\Phi}, \quad (2)$$

где $\theta(ST, CS)$ - обобщенная оценка, $H(ST, CS)$ - синтаксическое сходство, $\Phi(ST, CS)$ - семантическое (функциональное) сходство, w_i - весовой коэффициент, соответствующий каждому типу сходства, предназначенной для более гибкого управления критерием сравнения. Подробное описание предложенного алгоритма обнаружения

web-сервисов в среде Интернет приведено в работе [Larysa Globa, 2015].

1.4. Динамическое формирование и выполнение сценариев обработки заданий пользователя

Основной задачей при проектировании последовательности элементов сценариев обработки заданий пользователя является упорядоченное объединение соответствующих частичных элементов сценария для включения их в сложный сценарий. Сложный сценарий пользователя можно идентифицировать со сложным инженерным или научным расчетом, который реализуется в программной среде в виде конкретного workflow, а каждый частичный элемент сценария идентифицируется с подрасчетом решения конкретной задачи в ПО, реализуемым в виде web-сервиса. Множество частичных элементов сценария – чСэ хранится в независимом хранилище, которое не имеет связей с хранилищем сложных сценариев – сСэ. Структура каждого из сложных и частичных сценариев Сэ включает название и параметры Сэ. Поиск частичных сценариев чСэ, которые необходимо включить в сложный, выполняется в соответствии с подходом, представленным в разделе 1.3, и базируется на сравнении параметров чСэ, поскольку в сложный сСэ включаются только те частичные, у которых есть общие параметры, или области значений соответствующих параметров пересекаются. Формальное описание сценариев Сэ:

- 1) множество сложных сСэ:

$$R^3 \ni r_k^3, r_k^3 = \langle T_k^3, p_{kj}^3 \rangle,$$

где R^3 - множество сложных сценариев сСэ,
 r_k^3 - k -й сценарий Сэ из множества сложных сценариев сСэ,
 T_k^3 - название k -ого сценария Сэ из множества сложных сценариев сСэ,
 p_{kj}^3 - p -й параметр k -ого сценария Сэ из множества сложных сценариев сСэ.

- 2) множество частичных сценариев чСэ:

$$R^4 \ni r_l^4, r_l^4 = \langle T_l^4, p_{lq}^4 \rangle,$$

где R^4 - множество частичных сценариев чСэ,
 r_l^4 - l -й сценарий Сэ из множества частичных сценариев чСэ,
 T_l^4 - название l -ого сценария Сэ из множества частичных сценариев чСэ,
 p_{lq}^4 - q -й параметр l -ого сценария Сэ из множества частичных сценариев чСэ.

Множества P_{kj}^3 и P_{lq}^4 могут пересекаться или не пересекаться, что отображает наличие у обоих Сэ одинаковых параметров или их отсутствие соответственно. Сложный сценарий Сэ формируется из частичных сценариев чСэ на базе сравнения значений параметров, которые используются в сложном сценарии Сэ. Это позволяет автоматизировать процесс формирования

сложного сценария, устанавливать порядок его формирования, выполнять формирование на «ленту». Предлагается сложный сценарий Сэ представить в виде упорядоченного дерева, порядок обхода вершин которого соответствует порядку вычислений с помощью найденных web-сервисов, что упростит процедуру его формирования и модификации. Исходя из того, что рассматриваемое дерево сценария является упорядоченным, деревом с корневым узлом и заданным порядком прохождения дочерних узлов, следует, что последовательность выполнения частичных сценариев Сэ будет соответствовать введенному порядку на дереве и может выполняться в процессе построения дерева, т.е. на «ленту». Этапы метода динамического формирования сложного сценария:

Этап 1. На первом этапе происходит отбрасывание из множества R^s сценариев, в которых ни один $p_{k,j}^s$ не равняется никакому $p_{i,q}^s$.

Этап 2. На этом этапе происходит сравнение значений $p_{i,q}^s$ сценариев, отобранных на этапе 1, для отсека подмножества сценариев, которые имеют общие параметры, но множества их значений не пересекаются.

Этап 3. На этапе 3 проводится упрощение формулы частичного сценария Сэ.

Подробное описание метода формирования сложного сценария представлено в работе [Новогрудська, 2015]. Полученный сложный сценарий для определенного инженерного, научного расчета или для процесса анализа больших объемов данных может быть представлен как шаблон для дальнейшего использования.

Для визуального анализа полученного на «ленту» сценария предлагается использовать метаграфы [Globa, 2015]. Метаграф - это двойка: порождающее множество вершин и метавершин и множество ребер. В определении метаграфа важным является рассмотрение двух составляющих: множество метавершин и множество метаребер. Так как для решения задачи визуализации метаграфа важны типы узлов и их соотношения, целесообразно выделить отдельно множество вершин и множество метавершин метаграфа. Выделять и отдельно рассматривать метаребра нет необходимости, так как при визуализации метаграфа важным является наличие ребер, но не их тип. В связи с этим множество ребер будет содержать все ребра метаграфа, независимо от того, какие типы узлов метаграфа они соединяют. Цель применения метаграфов заключается в возможности моделирования бизнес-процессов с использованием диаграмм выполнения работ (workflow), визуальном представлении и анализе аномалий в нечетких наборах данных. С точки зрения задачи моделирования бизнес-процессов (workflow) при нечеткости метаописаний web-сервисов, вершины можно представить в виде метаописаний тех web-сервисов, при поиске которых все критерии поиска выполнены, а метавершины - это метаописания таких web-сервисов, критерии поиска которых

удовлетворены частично или когда для решения одной задачи найдено множество сервисов-претендентов. В таком случае визуальный анализ аномалий в метаграфе полученного workflow может позволить избежать ошибочных решений. Подробное описание метода визуализации workflow средствами метаграфов представлено в работах [Amit Basu, 2008, Globa, 2015]. Таким образом, использование онтологий в комплексе с методами семантического поиска позволяет сформировать целостную методологию для создания сценариев (workflow) решения сложных задач в среде Интернет, а визуализация сценариев обработки информации в виде метаграфов повысит релевантность полученных решений, что упростит выполнение сложных наукоемких расчетов широкому кругу пользователей в различных сферах деятельности.

2. Комплекс инструментальных средств построения сценариев обработки данных

Комплекс инструментальных средств построения сценариев обработки данных является частью общесистемной структуры программных средств порталов знаний и включает подсистему ведения онтологий (коллекционер онтологической информации), конструктор запросов и формирователь ответов. Коллекционер онтологической информации о ресурсах использует стандартные механизмы поиска, последовательно обрабатывает ссылки на документы, страницы, сайты, определяя их тематику, структуру, степень релевантности между тематикой и информацией, которая в них содержится, сохраняет их в виде онтологий. Коллекционер анализирует новые источники и проверяет старые. Формирователь ответов обрабатывает источники, отслеживает степень их релевантности понятиям и выдает список наиболее подходящих для данного понятия ресурсов. Портал знаний имеет язык запросов, близкий к естественному языку. Это требует наличия в нем соответствующего лингвистического обеспечения, одним из основных компонентов которого является словарь терминов, органически связанный с онтологией. На основе лингвистического обеспечения решаются следующие важные задачи: задание лингвистической информации о данной предметной области, для чего используются словарь, организованный с учетом проблемно-ориентированной лексики, морфологии, словообразования, синонимии.

В виде онтологической модели выполняется описание системы знаний о предметной области как комплекса понятий, связанных между собой отношениями.

Один из возможных вариантов реализации подсистемы ведения онтологий на портале знаний представлен на рисунке 1. Такая подсистема ориентирована на участие экспертов из конкретных областей знаний. Она позволяет им собрать и

систематизировать в рамках единого информационного пространства обширные знания и данные из этих областей знаний. Для построения онтологий и управления ими служит редактор онтологий, реализованный как web-сервис. Формы для ввода конкретных информационных объектов и связей между ними автоматически генерируются на основе онтологии. Управление информационным контентом осуществляется с помощью редактора данных, который позволяет создавать, редактировать и удалять информационные объекты и связи между ними.

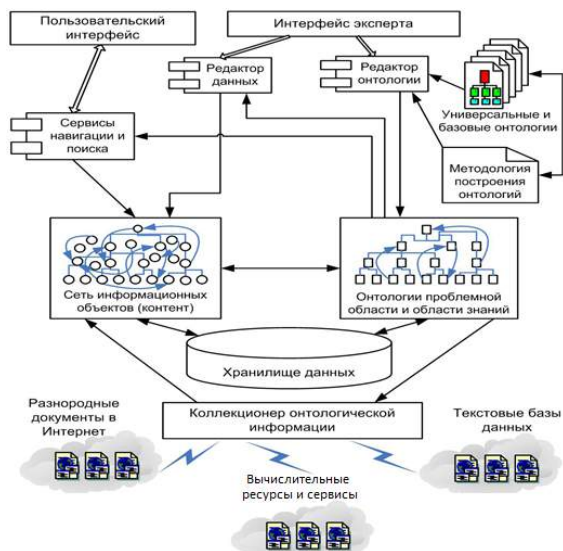


Рисунок 1 – Подсистема ведения онтологий на портале знаний

Комплекс инструментальных средств построения бизнес-процессов и обработки больших объемов данных включает набор интегрированных программных технологий и продуктов для обработки и хранения информации и позволяет:

- предоставлять пользователям доступ к данным, которые расположены в Интернет-среде;
- предоставлять данные группе конечных пользователей в форме, которая соответствует их коллективному представлению о данных;
- сокращать время ответа на запрос;
- предоставлять данные, структурированные в соответствии с требованиями доступа;
- упрощать выполнение задач очистки, загрузки, преобразования, интеграции и анализа данных;
- формировать сценарии обработки данных «на лету».

Таким образом, при обращении к portalу знаний пользователь имеет возможность получить в ответ ресурсы семантически релевантные его запросу и обработать их с помощью соответствующего запросу сценария.

Заключение

Рассмотренный в статье подход к автоматизации построения сценариев обработки данных отличается использованием онтологических моделей для описания информационных и вычислительных ресурсов, теории метаграфов для

визуализации и анализа полученных сценариев, является особенно эффективным в таких сложных сферах деятельности как инженерия и научные исследования.

Использование онтологий в комплексе с методами семантического поиска позволяет сформировать сценарии (workflow) решения сложных наукоемких задач в среде Интернет.

Визуализация сценариев обработки информации в виде метаграфов позволяет повысить релевантность полученных «на лету» сценариев, использование шаблонов упрощает выполнение сложных наукоемких расчетов широкому кругу пользователей в различных сферах деятельности.

Среди вопросов, которые необходимо дополнительно исследовать, можно выделить исследование неоднозначности построенных сценариев в случае смысловой нечеткости описания онтологий, многозначности описаний одних и тех же ресурсов в среде Интернет, а также развитие подхода к созданию сценариев обработки больших объемов данных с целью повышения производительности их обработки.

Библиографический список

- [Business Information Systems, 2014] Lecture Notes in Business Information Processing / Witold Abramowicz, Angelika Kokkinaki (Eds.) - ©Springer International Publishing Switzerland, 2014, 265 .
- [Новогрудська, 2015] Новогрудська Р.Л. Інформаційна технологія створення та підтримки порталів інженерних знань / Р.Л. Новогрудська // диссерт. на соиск. ученой степени к.т.н.; – Киев, 2015, 186 с.
- [Amit Basu, 2008] Amit Basu, Robert W. Blanning Metagraphs and their applications/ A. Basu, R. Blanning // A. Basu, R. Blanning Library of Congress Control Number: 2006930395, Springer Science+Business Media, LLC, 2008, 173 p.p.
- [Larysa Globa, 2015] L. Globa, M. Kovalskiyi, A. Stryzhak Increasing web services discovery relevancy in the multi-ontological environment/A. Wiliński, Jerzy Pejaś Advances in Intelligent Systems and Computing// ISBN 978-3-319-15147-2 (eBook): Springer International Publishing Switzerland, 2015, p. 335-344
- [Globa, 2015] L.Globa, M.Ternovoy, O.Shtogrina, O. Kryvenko Based on Force-Directed Algorithms Method for Metagraph Visualization /A. Wiliński, Jerzy Pejaś Advances in Intelligent Systems and Computing// ISBN 978-3-319-15147-2 (eBook): Springer International Publishing Switzerland, 2015 – p. .359-370.

ONTOLOGIES USING FOR DATA PROCESSING SCENARIOS DESIGNING

Globa L.S.

NTUU «KPI », Kiev, Ukraine

lgloba@its.kpi.ua

The paper deals with method of the data processing scenarios (workflow) designing based on the ontological description of the information and computing resources in conditions of the fuzzy metadata presence, evaluation of semantic similarity, the data processing scenario tree visualization as matagraph for the scenario problem control.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.896

МЕТОД ФОРМИРОВАНИЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ НА ПОРТАЛАХ ЗНАНИЙ

Глоба Л.С., Новогрудская Р.Л.

*Национальный технический университет Украины «Киевский политехнический институт»,
г. Киев, Украина*

lgloba@its.kpi.ua

rinan@ukr.net

Предложен метод формирования сложного инженерного расчета, который задает механизм объединения соответствующих частичных функциональных элементов в общий функциональный элемент. Согласно методу формирования сложного инженерного расчета поиск частичных функциональных элементов, которые необходимо включить в общий, базируется на сравнении их параметров, поскольку в общий функциональный элемент включаются только те частичные функциональные элементы, в которых есть общие параметры, или области значений параметров которых пересекаются. Процесс формирования общего функционального элемента представлено с помощью упорядоченного дерева, которое упрощает процедуру его модификации.

Ключевые слова: портал знаний, инженерные расчеты, функциональные элементы, метод формирования сложного инженерного расчета.

Введение

Среди средств, предоставляющих доступ к информации в сети Интернет, можно выделить электронные архивы, интернет-библиотеки, корпоративные сайты и порталы знаний. Наиболее эффективными среди них являются порталы знаний, поскольку они предоставляют возможность получения доступа к информации наряду с выполнением определенных пользовательских заданий. В последнее время для построения порталов знаний используются различные подходы, среди которых онтологический, алгебраический и др. [Глоба, 2012], [Андреева, 2006], [Боровикова, 2008]. Однако, если речь идет о порталах именно инженерных знаний, то возникает ряд проблем связанных с наличием большого количества инженерных расчетных задач.

Существующие программные средства, которые предоставляют специализированные сервисы решения сложных инженерных задач, зависят от узкой предметной области, для которой они разрабатывались, их трудно адаптировать для решения аналогичных задач родственных предметных областей, а также они имеют ограниченные возможности для обработки информационных и функциональных элементов, реализующих инженерную расчетную задачу.

1. Характерные особенности инженерных расчетных задач

Для решения расчетных задач предметной области доступных пользователю на портале, необходимо не только описать их элементы, но и установить связи между конкретными сервисами (набором сервисов) для реализации конкретных рабочих процессов (workflow) в рамках совместных рабочих процессов портала и связь сервисов с информационными ресурсами, которые необходимы для выполнения расчетов. Важным фактором является оптимизация этих связей. В зависимости от параметров, заданных пользователем, и тематикой расчетной задачи, последняя может решаться с помощью того или иного алгоритма [Трошенко, 2005]. Кроме того, при построении реальных научных и инженерных задач используют их декомпозицию на подзадачи в зависимости от направления общей задачи и параметров, необходимых для выполнения расчета [Зубченко, 2001]. Подрасчеты динамично komponуют в общую задачу, причем одна подзадача может использоваться в нескольких общих задачах. Это обуславливает потребность учета и описания логики связности расчетных задач и их корректной интеграции в информационную среду портала.

Рассмотрено комплексные инженерные расчетные задачи предметной области «Прочность

материалов»: «Расчет на прочность силовых элементов магнитных систем ИТЕР», «Расчет на прочность оборудования и трубопроводов атомных энергетических установок» [Нормы расчета на прочность, 1984], [Нормы расчета на прочность, 1986]. Анализ структуры этих комплексных расчетных задач позволил выделить следующие важные особенности:

- декомпозиция общей задачи на подзадачи,
- иерархическая вложенность подзадач,
- использование одинаковых подзадач в разных общих задачах,
- зависимость этапов расчетной задачи от тематик, параметров или характеристик расчета,
- различная направленность расчетных задач.

Анализ требований к представлению инженерных расчетов на порталах инженерных знаний определил необходимость задать описания элементов этих расчетов для реализации функции сочетания таких элементов в момент выполнения расчетов.

2. Основные определения

Множество частичных функциональных элементов хранится в независимом хранилище, которое не имеет связей с хранилищем общих функциональных элементов. Структура общих и частичных функциональных элементов включает название и параметры.

Рассмотрим множество общих функциональных элементов:

$$\Phi^{\exists 0} \ni \Phi_k^0, \quad \Phi_k^0 = (T_k^o, P_{ki}^o), \quad (1)$$

где $\Phi^{\exists 0}$ – множество общих функциональных элементов;

Φ_k^0 – k-й функциональный элемент из множества общих функциональных элементов;

T_k^o – название k-го функционального элемента из множества общих функциональных элементов;

P_{ki}^o – j-й параметр k-го функционального элемента из множества общих функциональных элементов.

Множество частичных функциональных элементов представлено как:

$$\Phi^{\exists q} \ni \Phi_j^q, \quad \Phi_j^q = (T_j^q, P_{ja}^q), \quad (2)$$

где $\Phi^{\exists q}$ – множество частичных функциональных элементов;

Φ_j^q – l-й функциональный элемент из множества частичных функциональных элементов;

T_j^q – название l-го функционального элемента из множества частичных функциональных элементов;

P_{ki}^q – q-й параметр l-го функционального элемента из множества частичных функциональных элементов.

Множества Φ_{ki}^o та Φ_{ki}^q могут пересекаться или не пересекаться, что означает наличие у обоих функциональных элементов одинаковых параметров или их отсутствие. Необходимо разработать метод, который обеспечит формирование общего функционального элемента из частичных на базе сравнения значений параметров функциональных элементов и даст возможность устанавливать порядок проведения вычислений общих функциональных элементов «на лету».

3. Метод формирования сложного инженерного расчета

Предлагается представить процесс формирования общего функционального элемента с помощью упорядоченного дерева, что упростит процедуру модификации функционального элемента. Учитывая, что упорядоченное дерево - это дерево с корнем, в котором определен порядок прохождения дочерних узлов, использование упорядоченных деревьев для представления общего функционального элемента позволит определить последовательность выполнения частичных функциональных элементов.

Дерево формирования общего функционального элемента, построенное с использованием описанного подхода, показано на рисунке 1. Отсеченные узлы показывают расчеты, которые были отвергнуты при переборе, корневой узел - общий функциональный элемент, листьями которого являются частичные функциональные элементы, которые будут включаться в общий.

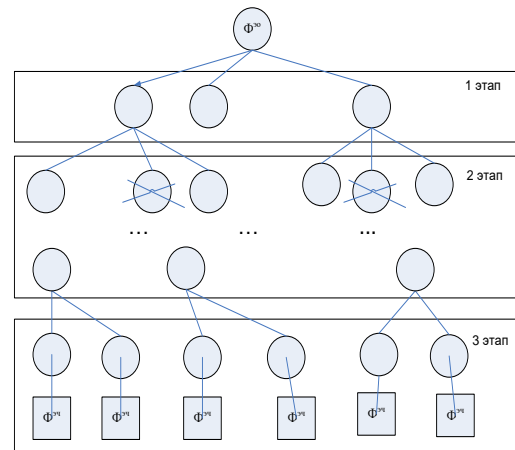


Рисунок 1 - Дерево формирования общего функционального элемента

Предлагается выделить конечные частичные функциональные элементы и промежуточные частичные функциональные элементы. Конечные частичные функциональные элементы есть неделимыми частичными функциональными элементами, они являются листьями дерева. Промежуточные частичные функциональные

элементы – это частичные функциональные элементы, которые, в свою очередь, состоят из некоторого множества частичных функциональных элементов и отражены в дереве всеми вершинами кроме корневой и листовых.

Метод формирования сложного инженерного расчета состоит из трех этапов (рисунок 2).

На первом этапе проводится изъятие из множества частичных функциональных элементов тех, в которых ни один из z $p_{i_q}^z$ не равен никакому $p_{k_j}^z$. Проводится анализ множества частичных функциональных элементов и выбираются те из них, параметры которых совпадают с параметрами общего функционального элемента по правилу $p_{k_j}^z = \bigcup_{l=1}^m p_{i_q}^z$. Причем возможно выполнение $p_{i_q}^z \cap p_{j_q}^z$, где $i, j \in l, l = \overline{1, m}$. В результате получаем подмножество множества частичных функциональных элементов, которое удовлетворяет правилу, и дерево общего функционального элемента, вершинами второго уровня которого являются элементы этого подмножества.

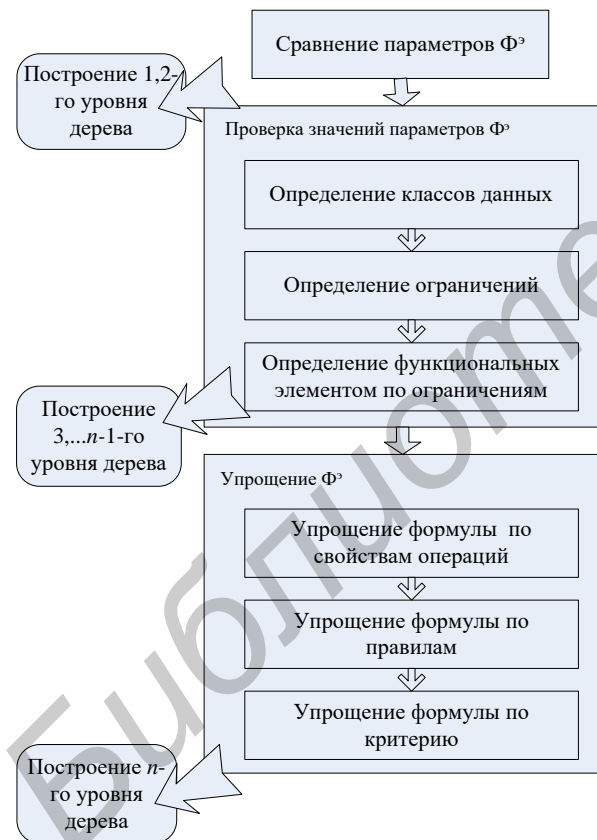


Рисунок 2 - Метод формирования сложного инженерного расчета

Второй этап - этап проверки значений параметров частичных функциональных элементов $Mn(p_{i_q}^z)$ с целью отсека подмножества функциональных элементов, которые имеют общие параметры, но множества их значений не пересекаются. На этом этапе происходит сравнение значений параметров частичных функциональных элементов из множества, которое было

сформировано после первого этапа со значениями соответствующих параметров общего функционального элемента: $Mn(p_{i_k}^z) \subset Mn(p_{j_t}^z)$, при условии, что $p_{i_k}^z = p_{j_t}^z$. Результатом проверки могут быть несколько вариантов.

1. Области совпадают – $Mn(p_{i_k}^z) = Mn(p_{j_t}^z)$. В этом случае область значений k -го параметра i -го общего функционального элемента совпадает с областью значений t -го параметра j -го частичного функционального элемента. Таким образом данный частичный функциональный элемент включается как узел в дерево общего функционального элемента.

2. Одна область включает другую:

– $Mn(p_{i_t}^z) \subset Mn(p_{i_k}^z)$. В этом случае область значений t -го параметра j -го частичного функционального элемента включает область значений k -го параметра i -го общего функционального элемента. Такой частичный функциональный элемент включается как узел в дерево общего функционального элемента;

– $Mn(p_{i_k}^z) \subset Mn(p_{j_t}^z)$. В этом случае область значений k -го параметра i -го общего функционального элемента включает область значений t -го параметра j -го частичного функционального элемента. Данный частичный функциональный элемент включается как узел в дерево общего функционального элемента, но продолжается анализ значений параметра для других частичных функциональных элементов с целью поиска того частичного элемента, область значений параметра которого в объединении с данным будет формировать область значений параметра общего функционального элемента: $Mn(p_{j_t}^z) \cup Mn(p_{i_d}^z) = Mn(p_{i_k}^z)$.

3. Области пересекаются – $Mn(p_{i_k}^z) \cap Mn(p_{j_t}^z)$.

В этом случае область значений k -го параметра i -го общего функционального элемента пересекается с областью значений t -го параметра j -го частичного функционального элемента. Такой частичный функциональный элемент включается как узел в дерево общего функционального элемента, но продолжается анализ значений параметра для других частичных функциональных элементов с целью поиска такого частичного элемента, область значений параметра которого в объединении с данным будет формировать область значений параметра общего функционального элемента: $M(p_{j_t}^z) \cup M(p_{i_d}^z) = M(p_{i_k}^z)$.

4. Области не пересекаются – $Mn(p_{i_k}^z) \neq Mn(p_{j_t}^z)$. В этом случае область значений k -го параметра i -го общего функционального элемента не пересекается с областью значений t -го параметра j -го частичного функционального элемента. В данном случае частичный функциональный элемент не включается в дерево общего функционального элемента.

На третьем этапе происходит упрощение формулы расчета по методу упрощения формул

алгебраической системы расчетов [Глоба, 2014]. Характеристики реальных инженерных расчетов являются параметрами функциональных элементов, а для алгебраической системы расчетов - элементами пространства данных. Этап упрощения формулы расчета включает следующую последовательность шагов:

- представление формул частичных расчетов в элементах алгебры,
- упрощения формул на основании свойств операций и правил упрощения,
- представление минимальной формы частичных расчетов, которые включены в общий.

Полученное дерево функционального элемента может быть сохранено в базе знаний как шаблон для периодического применения с возможностью его модификации. Такая модификация дерева может быть выполнена за счет использования стандартных операций над деревьями. Представление инженерной задачи в виде функционального элемента с древовидной структурой позволяет обрабатывать каждую вершину этого дерева, которая представляет отдельный частичный расчет, параллельно, обходя ветви дерева, которые являются независимыми друг от друга.

Метод формирования сложного инженерного расчета является основой для объединения реальных инженерных задач в комплексный общий расчет предметной области.

Заключение

Разработан метод формирования сложного инженерного расчета в процессе его выполнения, который позволяет сочетать частичные расчетные задачи портала инженерных знаний в общую расчетную задачу, которая решается по запросу конечного пользователя.

Предложен способ представления процесса формирования общего функционального элемента в виде дерева, который дает возможность устанавливать структуру общих функциональных элементов и повышает эффективность процесса их выполнения за счет параллельной обработки независимых друг от друга ветвей дерева общего функционального элемента.

Библиографический список

[Глоба, 2012] Глоба Л.С. Модель представления знаний на специализированном Интернет-портале в области сопротивления материалов / Л.С. Глоба, Р.Л. Новогрудская // Системные исследования и информационные технологии. – 2012. – №2. – С. 42–48

[Андреева, 2006] Андреева О.А. Археологический портал знаний: содержательный доступ к знаниям и информационным ресурсам по археологии / О.А. Андреева и др. // Труды 10-й национальной конференции по искусственному интеллекту с международным участием (КИИ–2006). – М.: Физматлит. – 2006. – Т. 3. – С. 832–840.

[Боровикова, 2008] Боровикова О.И. Разработка портала знаний по компьютерной лингвистике / Ю.А. Загоруйко, О.И. Боровикова и др. // Труды 11-ой национальной конференции по

искусственному интеллекту с международным участием (КИИ–2008). – М.: ЛЕНАНД. – 2008. – Т. 3. – С. 380–388.

[Трошенко, 2005] Прочность материалов и конструкций / Редкол.: В.Т. Трошенко (отв. ред.) и др. – К.: Академперіодика, 2005. – 1088 с.

[Зубченко, 2001] Зубченко А.С. Марочник сталей и сплавов / А.С. Зубченко. – М.: Машиностроение, 2001. – 663 с.

[Нормы расчета на прочность, 1984] Нормы расчета на прочность силовых элементов магнитных систем ИТЭР. – Киев: Институт проблем прочности АН Украины, 1984. – 73 с.

[Нормы расчета на прочность, 1986] Нормы расчета на прочность оборудования и трубопроводов атомных энергетических установок / под ред. О.М. Малявина — М.: Энергоатомиздат, 1989. – 525 с.

[Глоба, 2014] Глоба Л.С., Подход к построению формальной алгебраической системы порталов знаний / Л.С. Глоба, Р.Л. Новогрудская // Онтология проектирования. – 2014. – №2(11). – С. 40–59.

THE METHOD OF COMPLEX ENGINEERING CALCULATION OF KNOWLEDGE PORTALS COMPOSITION

Globa L.S., Novograduska R.L.

National technical university of Ukraine «Kyiv polytechnic institute», Kyiv, Ukraine

lgloba@its.kpi.ua

rinan@ukr.net

The method of complex engineering calculation composition is proposed, that defines a mechanism for partial functional elements combining into a general one. According to the proposed method, searching of partial functional elements to be included in the general one is based on the comparison of elements parameters as the general functional element include only those partial functional elements that have common parameters, or their parameter values are intersected.

The set of partial functional elements are stored in an independent store, which has no links with the repository of general functional elements. Structure of general and partial functional elements include the title and parameters. It is proposed to represent the process of general functional element composition using an ordered tree that will simplify the procedure of functional element modification.

The method of complex engineering calculation composition includes three stages. On the first stage the set of partial functional elements is analyzed and those elements are selected parameters of which corresponds to the general functional element parameters. On the second stage the comparison of parameter values is held for those partial functional elements that were chosen on the first step. On the third stage the simplification of functional element formula is held based on the method of algebraic system formulas simplification.

A method of complex engineering calculation composition is the basis for integration of real engineering calculation tasks in a complex global calculation of problem domain.



УДК 004.822:514

ПОДХОД К ОРГАНИЗАЦИИ КОМПЛЕКСНОЙ ПОДДЕРЖКИ ПРОЦЕССА РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СППР В СЛАБОФОРМАЛИЗОВАННЫХ ПРЕДМЕТНЫХ ОБЛАСТЯХ

Загорулько Г.Б., Загорулько Ю.А.

*Институт систем информатики им. А.П. Ершова Сибирского отделения
Российской академии наук, г. Новосибирск, Россия*

gal@iis.nsk.su

zagor@iis.nsk.su

Рассматривается подход к организации комплексной поддержки процесса разработки интеллектуальных СППР в слабоформализованных предметных областях. Такая поддержка осуществляется на трех уровнях: концептуальном, информационном и компонентном. Концептуальный уровень обеспечивается онтологией поддержки принятия решений, информационный уровень – специализированным интернет-ресурсом, содержащим систематизированную в соответствии с указанной онтологией информацию об области знаний «Поддержка принятия решений». Компонентный уровень составляет репозиторий реализованных методов поддержки принятия решений, систематизированных и описанных в соответствии с онтологией.

Ключевые слова: система поддержки принятия решений; слабоформализованная предметная область, комплексная поддержка процесса разработки, онтология.

Введение

В настоящее время остается актуальной проблема обеспечения поддержки принятия решений в слабоформализованных предметных областях. Для ее решения создаются интеллектуальные системы поддержки принятия решений (ИСППР). Однако разработка таких систем является очень трудной задачей, поскольку современные инструментальные средства разработки СППР либо не применимы в слабоформализованных предметных областях, либо малодоступны из-за высокой стоимости. Для решения этой задачи необходимо разработать технологию построения ИСППР для таких областей, включающую набор методов и средств, обеспечивающих всестороннюю поддержку процесса разработки ИСППР на всех этапах.

В разработке ИСППР участвуют не только программисты, но и инженеры знаний, владеющие методами представления знаний и поддержки принятия решений, и эксперты той предметной области (ПО), для которой создается СППР, поэтому на начальных этапах разработки очень важное значение имеет концептуальный базис, обеспечивающий указанные группы специалистов единой системой понятий.

Помимо общих представлений об области знаний «Поддержка принятия решений» и ПО, для которой разрабатывается система, разработчикам необходима информационная поддержка. Они должны иметь достаточно полную информацию о конкретных методах поддержки принятия решений (ППР), о классах задач, решаемых этими методами, о возможностях и ограничениях каждого из них. Кроме этого, разработчикам должна быть доступна информация об основных этапах принятия решений, об используемых на каждом из них методах, а также об инструментальных средствах, реализующих эти методы.

На этапе реализации ИСППР важную роль играет компонентная поддержка разработчиков. Возможность выбрать готовые программные компоненты, реализующие необходимые методы поддержки принятия решений, может существенно облегчить и ускорить процесс создания ИСППР.

В докладе описывается подход к организации комплексной поддержки процесса разработки интеллектуальных СППР в слабоформализованных предметных областях. В соответствии с ним концептуальный уровень такой поддержки представляет система онтологий. Информационная поддержка обеспечивается средствами специализированного Интернет-ресурса, в котором, в частности, систематизируются и описываются

конкретные методы поддержки принятия решений и их доступные реализации. Компонентную поддержку предоставляет репозиторий методов поддержки принятия решений, реализованных в виде сервисов, снабженных унифицированными спецификациями, на основе которых может выполняться их интеграция.

1. Концептуальный базис комплексной поддержки

В качестве концептуального базиса поддержки разработки ИСППР предлагается использовать систему взаимосвязанных онтологий – онтологии области знаний «Поддержка принятия решений», онтологии научных информационных ресурсов и онтологии задач и методов. Эти онтологии строятся на основе представительного набора базовых онтологий, составляющих знаниевый компонент технологии создания интеллектуальных научных интернет-ресурсов, разработанной в ИСИ СО РАН [Загорулько, 2014; Zagorulko et al., 2015]. Так, онтология области знаний «Поддержка принятия решений» строится на основе онтологий научного знания и научной деятельности, онтология научных информационных ресурсов – на основе базовой онтологии информационных ресурсов, а онтология задач и методов – на основе базовой онтологии задач и методов и онтологии сервисов.

При этом онтология задач и методов выделяется в отдельную онтологию ввиду особой важности для любой научной области знаний понятий «Задача» и «Метод». Эта онтология служит для детального описания решаемых в рассматриваемой ПО задач и существующих для их решения методов.

Методы решения задач реализуются с помощью соответствующих сервисов, для описания которых также используется отдельная онтология. Методы могут реализовываться как в виде локальных сервисов, так и в виде web-сервисов. Для того, чтобы web-сервисы можно было использовать из внешних приложений, необходимо иметь их стандартизированное описание. Для этих целей онтология сервисов базируется на стандарте OWL-S [OWL-S, 2004]. Это позволяет описывать в однозначной поддающейся машине обработке форме не только интерфейс сервиса (в терминах типов входных и выходных данных), но и его семантику, т.е. то, что сервис делает, указывать его связи с понятиями других онтологий, задавать ограничения на область применения и т.п. Использование стандартизированного семантического описания сервисов позволит обеспечить содержательный доступ к ним как пользователей, так и программных агентов, а также создаст возможность композиции из них новых сервисов с целью получения функциональности, требуемой для решения новых сложных задач.

Важную роль в концептуальном базисе играет понятие «Информационный ресурс». Для описания информационных ресурсов используется

специальная онтология, в которой задается типизация ресурсов и отношения, связывающие их с сущностями остальных рассмотренных онтологий. Эта онтология вводит информационные ресурсы, заданные в цифровом формате и которые пользователь может многократно использовать при решении своих задач.

2. Организация информационной поддержки

Для оказания информационной поддержки разработки ИСППР требуется построить базирующийся на описанной выше системе онтологий интеллектуальный научный интернет-ресурс (ИНИР), в котором будут собраны описания известных интеллектуальных методов принятия решений, заданы их связи с задачами и этапами принятия решений. В этом ресурсе будет также представлена информация об имеющихся инструментариях для разработки СППР, о коллективах и исследователях, занимающихся данной проблематикой. При этом на основе онтологий будет организован эффективный содержательный доступ к этой информации.

Для разработки такого ИНИР предполагается использовать упомянутую выше технологию создания интеллектуальных научных интернет-ресурсов [Загорулько, 2014]. На рисунке 1 показаны основные содержательные компоненты любого ресурса, разработанного средствами данной технологии.



Рисунок 1 – Содержательные компоненты ИНИР

Знания об области знаний ИНИР представлены онтологией, которая строится на основе базовых онтологий, приведенных на рисунке 1, а также правилами вывода, позволяющими получать знания и факты, не содержащиеся в контенте ИНИР в явном виде.

Контент ИНИР образуют информационные объекты, информационно-аналитические ресурсы и сервисы обработки информации, интегрированные в ИНИР. Информационные объекты – это конкретные экземпляры понятий онтологии. Они представляют структурированную в соответствии с онтологией информацию о рассматриваемой области знаний.

Каждый интегрированный в ИНИР информационный ресурс и сервис должен описываться соответствующим информационным объектом.

Совокупность интегрированных в ИНИР сервисов, образует репозитарий. Сервисы обработки информации предоставляют как традиционные для интернет-ресурсов средства доступа и анализа хранящейся в них информации, так и средства для решения задач рассматриваемой ПО. Добавление последних в ИНИР поднимает возможности оказания информационной поддержки на качественно новый уровень – пользователь не просто получает информацию об интересующем его методе или ссылку на его реализацию. Он может непосредственно на ресурсе просмотреть примеры использования метода, запустить его и проанализировать его работу с разными входными данными.

3. Организация компонентной поддержки

В процессе реализации ИСПП большую помощь разработчикам может оказать репозитарий – библиотека готовых к использованию методов принятия решений вместе с методикой и средствами их исполнения и композиции.

Этот репозитарий разрабатывается в соответствии с онтологией и с использованием сервис-ориентированного подхода, что упрощает включение в него имеющихся в свободном доступе готовых реализаций методов. Спецификация методов в виде сервисов позволяет снять ряд проблем, связанных с различием платформ, на которых разрабатывались методы, стандартов, форматов данных, удаленным размещением этих методов. Использование стандартизированных интерфейсов обеспечивает интеграцию методов при решении сложных задач. Сетевые протоколы, которые могут использоваться для взаимодействия между сервисами, позволяют получить доступ к информации и функциональным компонентам, размещенным на удаленных серверах.

Репозитарий будет включать представительный корпус методов ИСПП, организованный на основе онтологии. Он будет обеспечивать возможность исполнения любого интегрированного в него метода с передачей ему данных и просмотром результатов его работы. При этом сами методы могут быть физически развернуты на удаленных машинах и от пользователя будут скрыты детали их распределенной работы. Репозитарий будет также предоставлять возможность создания новых методов путем композиции нескольких, уже реализованных методов. Кроме того, предполагается сделать репозитарий расширяемым. Для этого будет реализована возможность публикации новых методов и предоставления доступа к ним.

Для продвинутых пользователей полезной возможностью будет наличие у методов программных интерфейсов (API) для обеспечения их встраивания в другие методы и/или приложения.

При работе с репозитарием пользователю не потребуется изучать дополнительное программное обеспечение, форматы описания и команды запуска методов, способы передачи данных между отдельными компонентами разрабатываемой системы и т.п. Во многих случаях ему будет достаточно свести исходную задачу к одной или нескольким задачам известных классов, для которых существуют готовые методы.

4. Близкие работы

Вопросам создания СППР посвящено множество работ, но ни в одной из них задача комплексной поддержки процесса разработки интеллектуальных СППР в слабоформализованных предметных областях не ставилась.

Так в работах [Черняховская и др., 2007; Варшавский, 2009] рассматриваются вопросы разработки интеллектуальных СППР с использованием рассуждений на основе прецедентов. Использованию ИСППР в нештатных ситуациях посвящены работы [Массель и др., 2013] и [Геловани и др., 2001].

В последнее время в отечественных и зарубежных публикациях обсуждаются проблемы и опыт использования онтологий при построении СППР в различных областях [Загорюлько и др., 2011; Черняховская и др., 2009; Sheng-Tun et al., 2003; Luigi et al., 2004], в том числе слабоформализованных [Casanovas et al., 2009]. Однако в этих работах даже не ставился вопрос создания и использования онтологии поддержки принятия решений.

Что касается информационных ресурсов, обслуживающих разработчиков СППР, то публикаций по этой теме практически нет. Описываются в основном системы, ориентированные на ЛПР (лиц, принимающих решение) или специалистов в конкретной предметной области, а не на разработчиков СППР. Например, в работе [Finkle-Perazzo et al., 2011] описан онлайн-ресурс, предоставляющий доступ специалистам к медицинским знаниям и интернет-ресурсам для принятия решений при лечении различных заболеваний. В работе [Zhang et al., 2015] представлена доступная через сеть Интернет система для поддержки принятия решений при управлении водозабором. Из отечественных разработок стоит отметить систему, описанную в [Горнов и др., 2006], осуществляющую удаленную компьютерную поддержку разработчиков, использующих математические пакеты. Однако, эта система поддерживает разработку приложений не в слабоформализованных областях, а в такой хорошо формализованной области, как вычислительная математика.

Заключение

В докладе предложена концепция комплексной поддержки процесса разработки интеллектуальных СППР в слабоформализованных предметных областях. Концептуальным базисом такой поддержки является система онтологий, ядро которой составляет онтология области знаний «Поддержка принятия решений». Для обеспечения информационной поддержки данного процесса служит интернет-ресурс, представляющий структурированное на основе системы онтологий описание области знаний «Поддержка принятия решений», включая задачи, решаемые в данной области, и методы, используемые для их решения. Для обеспечения компонентной поддержки процесса разработки ИСППР служит репозиторий методов ППР, предоставляющий непосредственный доступ к реализациям методов и позволяющий разработчикам предварительно опробовать их, чтобы лучше понять их возможности и выбрать наиболее подходящие из них.

Работа выполнена при поддержке гранта РФФИ № 16-07-00569 «Методы и средства комплексной поддержки процесса разработки интеллектуальных СППР в слабоформализованных предметных областях на основе сервис-ориентированного подхода и технологий Semantic Web».

Библиографический список

[Загорулько, 2014] Загорулько Ю. А. Технология разработки интеллектуальных научных интернет-ресурсов, ориентированная на экспертов предметной области // Инфраструктура научных информационных ресурсов и систем. Сборник избранных научных статей. Труды Четвертого Всероссийского симпозиума (С.-Петербург, 6–8 октября 2014 г.). Под ред. Е.В. Кудашева, В.А. Серебрякова. М.: ВЦ РАН, 2014 – Т.1. – С.69-86.

[Zagorulko et al., 2015] Zagorulko Y., Zagorulko G. Ontology-Based Technology for Development of Intelligent Scientific Internet Resources // Intelligent Software Methodologies, Tools and Techniques. Proceedings of 14th International Conference, SoMet 2015. Hamido Fujita, Guido Guizzi (Eds.), / Communications in Computer and Information Science, Vol. 532, Springer International Publishing, 2015. –pp. 227-241.

[OWL-S, 2004] OWL-S: Semantic Markup for Web Services. [Электронный ресурс]. URL: <http://www.w3.org/Submission/OWL-S/> (дата обращения: 15.12.2015).

[Черняховская и др., 2007] Черняховская Л.Р., Старцева Е.Б., Муксимов П.В., Макаров К.А. Поддержка принятия решений при управлении сложными производственными системами на основе онтологической базы знаний // Вестник Уфимского государственного авиационного технического университета. 2007. Т. 9. № 7. –С. 41-45.

[Варшавский и др., 2009] Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений, 2009. №2. – С. 45–57.

[Массель и др., 2013] Массель Л.В., Массель А.Г. Технологии и инструментальные средства интеллектуальной поддержки принятия решений в экстремальных ситуациях в энергетике // Вычислительные технологии. - 2013.- Т.18.- Специальный выпуск. - С. 37-44.

[Геловани и др., 2001] Геловани В.А., Башлыков А.А., Бритков В.Б., Вязилов Е.Д. Интеллектуальные системы поддержки принятия решений в нештатных ситуациях с использованием информации о состоянии природной среды. Эдиториал УРСС, 2001.

[Загорулько и др., 2011] Загорулько Ю.А., Загорулько Г.Б. Использование онтологий в экспертных системах и системах поддержки принятия решений // Труды Второго симпозиума «Онтологическое моделирование» (Казань, октябрь 2010 г.) – Москва: ИПИ РАН, 2011. –С. 321-351.

[Черняховская и др., 2009] Черняховская Л.Р., Кружков В.Н., Дикова Ф.А. Онтологический подход к разработке системы поддержки принятия решений // Электронный журнал “Информационные ресурсы России”, 2009. №1.

[Sheng-Tun, 2003] Sheng-Tun Li, Huang-Chih Hsieh, and I-Wei Sun. An Ontology-based Knowledge Management System for the Metal Industry. In Proceedings of the Twelfth International World Wide Web Conference (WWW2003), Budapest, Hungary, 2003.

[Luigi, 2004] Luigi Ceccaroni, Ulises Cortés, Miquel Sánchez-Marrè. OntoWEDSS: augmenting environmental decision-support systems with ontologies. Environmental Modelling & Software. Vol. 19, Issue 9, September 2004. 785–797.

[Casanovas et al., 2009] Casanovas P., Casellas N., Vallbe J.-J. An Ontology-Based Decision Support System for Judges. In Proceeding of the 2009 conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood, IOS Press, Amsterdam, 2009. –pp. 165–175.

[Finkle-Perazzo et al., 2011] Finkle-Perazzo D, Jetha N. Online resources to enhance decision-making in public health // Chronic diseases and injuries in Canada, 2011. –Vol.31. Is.4. –pp.172-175.

[Zhang et al., 2015] Zhang D., Chen X., Yao H. Development of a Prototype Web-Based Decision Support System for Watershed Management // Water, 2015. № 7. –pp. 780-793.

[Горнов и др., 2006] Горнов А.Ю. Разработка информационно-вычислительной системы для экспертной поддержки пользователей математических пакетов при численном решении задач оптимального управления / А.Ю. Горнов, Т.С. Зароднюк // Современные технологии. Системный анализ. Моделирование. – ИрГУПС. – 2006. – №1. – С. 114–119.

AN APPROACH TO ORGANIZATION OF INTEGRATED SUPPORT OF THE DEVELOPMENT OF INTELLIGENT DSS IN WEAKLY FORMALIZED DOMAINS

Zagorulko G.B. *, Zagorulko Yu.A. *

* *A.P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of
Sciences, Novosibirsk, Russia*

gal@iis.nsk.su

zagor@iis.nsk.su

The paper presents the approach to organization of an integrated support of developers of intelligent decision support systems in weakly formalized domains. In accordance with the approach such support includes the levels: conceptual, information and component. The conceptual level is provided by the ontology of decision support (DS). Others levels are based on this ontology. The information level is supported by online Internet resource presenting information about DS domain systematized in accordance with the ontology. The component support is provided by the library which contains implementations of DS methods systematized and described in accordance with the ontology.

This work is supported by the Russian Foundation for Basic Research, grant № 16-07-00569 “Methods and tools for integrated support the development of intelligent DSS in weakly formalized domains based on service-oriented approach and Semantic Web technologies”.



УДК 004.822:514

РАЗРАБОТКА ЯЗЫКА СИТУАЦИОННОГО УПРАВЛЕНИЯ В ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЕ СЕМИОТИЧЕСКОГО ТИПА НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО ИНЖИНИРИНГА

Массель Л.В., Массель А.Г.

*Институт систем энергетики им. Л.А. Мелентьева СО РАН,
г. Иркутск, Россия*

massel@isem.sei.irk.ru
amassel@gmail.com

В статье рассматриваются вопросы разработки языка описания знаниями и управления знаниями в интеллектуальной системе семиотического типа для ситуационного управления в условиях экстремальных ситуаций в энергетике. Определение системы семиотического типа вводится на основе трактовки Д.А. Поспелова. В качестве инструмента реализации правил изменения базовых множеств семиотической модели предлагается язык ситуационного управления. Приводится общее описание языка, разрабатываемого на основе онтологического инжиниринга, и составляющих его компонентов: для описания и манипулирования знаниями. **Ключевые слова:** семантическое моделирование, язык ситуационного управления, интеллектуальная система, семиотическая модель, онтологический инжиниринг

Введение

Авторами выполнено отображение идей Поспелова на современные информационные технологии, в частности, семантическое моделирование. Последнее рассматривается как одно из направлений семиотического моделирования, в котором преобладает графическое представление разрабатываемых моделей. Предложена архитектура Ситуационного полигона, который рассматривается как прототип интеллектуальной системы семиотического типа для ситуационного управления. Помимо инструментальных средств семантического моделирования, в него включена экспертная система *Advice*, обеспечивающая сопоставление ситуаций, требующих вмешательства, и соответствующих управляющих воздействий. Разработка языка СМЛ выполнена на основе онтологического инжиниринга проблемы ситуационного управления. Приводится метаонтология СМЛ и дается краткое описание языка СМЛ, который обеспечивает функции как создания и пополнения баз знаний, так и интеграции всех компонентов Ситуационного полигона.

1. Интеллектуальные системы семиотического типа.

Определение таких систем было дано Д.А. Поспеловым и рассматривалось впоследствии Г.С. Осиповым в [Осипов, 2002]. Описание семиотической модели приведено в статье авторов [Массель, 2015А], основные положения подхода, предлагаемого авторами, изложены в [Массель, 2015В] и в настоящем сборнике [Массель, 2016].

Д.А. Поспеловым была предложена общая схема ситуационного управления на основе семиотического подхода [Поспелов, 1986]. Основными блоками схемы являются: *Анализатор* - оценивает сообщения и определяет необходимость вмешательства системы управления в процесс, протекающий в объекте управления; *Классификатор* - используя хранящуюся в нём информацию, относит текущую ситуацию к одному или нескольким классам, которым соответствуют одношаговые решения; *Коррелятор* - определяет то логико-трансформационное правило (ЛТП), которое должно быть использовано. В случае наличия нескольких подходящих ЛТП Коррелятор прибегает к помощи *Экстраполятора*, который выбирает лучшее ЛТП из выбранных Коррелятором на основе прогноза возможного развития ситуаций. Если решение не может быть принято, то срабатывает *Блок случайного выбора* и выбирается одно из

воздействий, оказывающих не слишком большое влияние на объект, или система отказывается от какого-либо воздействия. К сожалению, в то время общая схема ситуационного управления так и не была полностью реализована.

Авторами выполнено отображение этой схемы на современные информационные технологии, а именно, блокам общей схемы ситуационного управления сопоставлены разработанные в авторском коллективе средства семантического моделирования и разрабатываемая экспертная система Advise [Массель, 2015А]. Предлагается для выполнения функций Классификатора в системе семиотического типа использовать авторские библиотеки онтологического и когнитивного моделирования OntoMap и CogMap. Функции Экстраполятора могут выполнять инструментальные средства событийного и вероятностного моделирования: авторские библиотеки EventMap и BayNet [Массель, 2012]. Для анализа ситуаций и выбора мероприятий, соответствующих конкретным ситуациям, разрабатывается экспертная система Advise. Для хранения баз знаний и семантических моделей предполагается использовать Репозиторий, первоначально разработанный в рамках ИТ-инфраструктуры исследований в энергетике и используемый в интеллектуальной ИТ-среде [Массель А., 2010], [Копайгородский, 2011].

Результаты, полученные авторами, применяются в области, связанной с энергетической безопасностью (ЭБ), поэтому ситуационное управление в данном контексте рассматривается как управление в условиях экстремальных ситуаций в энергетике (Contingency Management) [Массель, 2014]. Под экстремальными ситуациями (ЭКС) понимаются как критические, так и чрезвычайные ситуации, в соответствии со шкалой «норма – предкризис – кризис». Авторы уделяют большее внимание именно критическим ситуациям.

2. Ситуационный полигон как прототип интеллектуальной системы ситуационного управления семиотического типа.

Учитывая наличие факторов неопределенности, при ситуационном управлении зачастую не удается построить и использовать математические модели, поэтому авторы предлагают использовать методы семантического моделирования, к которым они относят онтологическое, когнитивное, событийное и вероятностное моделирование (на основе байесовских сетей доверия - БСД-моделирование) [Массель, 2013]. Семантические модели представляются, как правило, в графическом виде.

Семантическое моделирование рассматривается авторами, как одна из разновидностей семиотического моделирования, в которой преобладает графическое представление моделей. В качестве инструмента для реализации правил

изменения базовых компонентов формальной модели (что и превращает ее в семиотическую) предлагается язык ситуационного управления.

В лаборатории Информационных технологий в энергетике ИСЭМ СО РАН под руководством и при участии авторов была разработана интеллектуальная ИТ-среда, интегрирующая инструментальные средства семантического моделирования для поддержки онтологического (OntoMap), когнитивного (CogMap), событийного (EventMap) и вероятностного (BayNet) моделирования и геокомпонент для 3D-геовизуализации результатов моделирования [Массель А., 2010]. ИТ-среда рассматривается как основа для построения Ситуационного полигона – прототипа интеллектуальной системы ситуационного управления семиотического типа [Массель А., 2015].

Далее рассмотрим вопросы разработки языка описания и управления знаниями для целей ситуационного управления [Массель, 2015С].

3. Общее описание языка ситуационного управления CML.

Предлагается использовать язык ситуационного управления (Contingency Management Language – CML) для нескольких целей, как:

- 1) язык описания знаний и манипулирования знаниями;
- 2) инструмент классификации ситуаций («норма», критические ситуации, чрезвычайные ситуации);
- 3) средство инициации ЭС для установления соответствия между ситуациями и управляющими воздействиями;
- 4) средство вызова соответствующих инструментальных средств семантического моделирования и модулей отображения (для перехода от одного типа моделей к другому);
- 5) средства обращения к геокомпоненту для 3D-геовизуализации результатов моделирования.

Иными словами, CML рассматривается как надстройка над существующей версией Ситуационного полигона [Массель А., 2014] и выполняет функции как создания и пополнения баз знаний, так и функции интеграции всех компонентов Ситуационного полигона (рис. 1), что, по сути дела, и превращает его в *интеллектуальную систему управления семиотического типа*, поскольку именно с помощью CML можно описывать правила изменения компонентов классической формальной модели (T, R, A, P): множества основных символов T; множества синтаксических правил R; множества знаний о предметной области A; множества правил вывода решений (прагматических правил) P.

Ниже приводится *общее описание языка CML*, это не строгое описание, а скорее изложение идеи (с использованием элементов нотации Бэкуса-

Наура), которое предлагается для обсуждения, дополнения и развития.

CML включает словари, описания знаний и операторы манипулирования знаниями.



Рисунок 1 -- Архитектура Ситуационного полигона

Словари включают: словарь имен (понятий), словарь отношений, словарь действий, причем в первом выделяются подразделы, в которых хранятся основные понятия (концепты) предметной области, имена объектов и/или наименования программных компонентов, вызываемых с помощью CML. Словарь имен и понятий будет проблемно-ориентированным, т.е. ориентированным на конкретную предметную область. Подробно он здесь не рассматривается.

Можно привести пример из области энергетической безопасности. В общем виде можно определить понятие ситуации S как:

$$\langle S \rangle := \langle C \rangle | \langle E \rangle | \langle P \rangle | \langle G \rangle | \langle U \rangle, \quad (1)$$

где C – исходные ситуации, E – сценарии ЭКС, P – переходные ситуации, U – превентивные, оперативные или ликвидационные мероприятия (управляющие воздействия), G – целевые ситуации.

В [Массель, 2016] приведено краткое описание выполненного онтологического инжиниринга (подробное описание изложено в [Массель, 2015D]), в результате которого построена система онтологий проблемы ситуационного управления, в том числе, онтологии CML. На рис. 2 приведена одна из них – метаонтология CML.

4. Компонент описания знаний.

Для описания знаний используется простая ядерная конструкция (XYZ), предложенная в [Поспелов, 1986], где X, Z – понятия или имена, Y – отношение или действие. Авторы считают, что разработка универсального языка ситуационного управления навряд ли возможна и целесообразна. Скорее всего, универсальным может быть компонент манипулирования знаниями. В компоненте описания знаний, вероятно, будут выделены базовое ядро и проблемно-ориентированные составляющие.

Предлагаются следующие основные типы отношений (показаны на примерах из области энергетической безопасности).

1. Отношения именования:

- < объект > <имеет> <имя>
- < объект > := <физический объект> | <программный компонент> | <информационный объект>

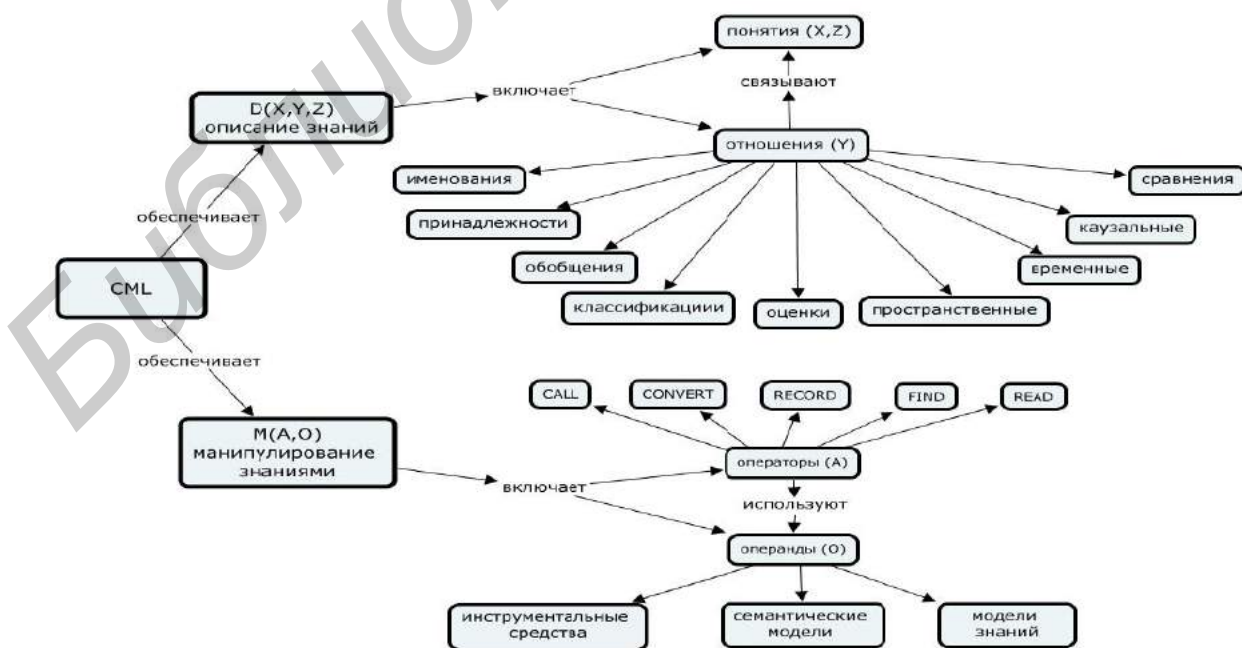


Рисунок 2 -- Метаонтология языка ситуационного управления CML

2. Отношения принадлежности:

<ситуация | объект > <относится | принадлежит>
<энергетическая система | ТЭК >
<энергетическая система> := <ЭЭС | ГСС | НСС | ТСС | ЯЭС >

3. Отношения обобщения:

Для <имя> : <объект> → <энергетическая система>
→ <ТЭК>

<объект> <является частью | входит>
<энергетическая система>
<энергетическая система> <является частью>
<ТЭК>

Для <место> :

< пункт > → <район> → <край | область > →
<страна>

<пункт> < входит | является частью> <район>
<входит > <край | область > < входит > <регион>
<входит > <страна>

4. Отношения классификации

<ситуация > <имеет тип> < исходная | переходная >
| <целевая>

<угроза ЭБ> <имеет тип> <техногенная | природная
| социально- политическая | управленческо-правовая
| внешнеэкономическая | внешнеполитическая>
< управляющее воздействие | мероприятие > <имеет
тип> <превентивное | оперативное |
ликвидационное >

5. Отношения оценки

<управляющее воздействие | мероприятие> <имеет
оценку> <R ^ C ^ T >

где R – требуемые ресурсы, C – цена мероприятия,
T – сроки реализации мероприятия (воздействия)

6. Пространственные отношения:

<ситуация > <происходит> <место>

7. Временные отношения:

<ситуация > <происходит> <время>

8. Каузальные (причинно-следственные) отношения

<ситуация S_i > <вызывает | является причиной > <
ситуация S_j >

9. Отношения сравнения

В этих отношениях используются индикаторы энергетической безопасности из множества $\{A\}$, которые принимают значения (a_1, a_2, \dots, a_n) . Если индикатор находится в диапазоне (a_i, a_j) – состояние «норма», в диапазоне (a_k, a_l) – предкризисное состояние (критическая ситуация), в диапазоне (a_m, a_n) – состояние кризисное (чрезвычайная ситуация).

Далее выполняется отношение сравнения $(a_i$ - текущее состояние индикатора ЭБ).

Если $a_i \leq a_l \leq a_j$,
то <ситуация> <является> <нормальная>

Если $a_k \leq a_l \leq a_l$,
то <ситуация> <является> <критическая>

Если $a_m \leq a_l \leq a_n$,
то <ситуация> <является> <чрезвычайная>

Заметим, что множество отношений может расширяться, дополняться и модифицироваться. В частности, отношение сравнения может быть более

сложным, если потребуется оценивать состояние не по одному индикатору ЭБ, а по их совокупности (что более соответствует реальности). Учитывая, что в настоящее время выделены 14 групп индикаторов ЭБ, каждая из которых включает от одного до 8 индикаторов, вычисляемых по специальным формулам [Сендеров, 2013], реализация отношения сравнения в интеллектуальной системе может стать отдельной, достаточно сложной задачей.

При реализации базы знаний для представления знаний используются продукционные и фреймвые модели знаний. Для хранения баз знаний используется Репозиторий Ситуационного полигона.

5. Компонент манипулирования знаниями.

При описании операторов манипулирования знаниями будем опираться на понятия авторской фрактальной стратифицированной модели [Массель, 2015А] и архитектуру Ситуационного полигона. Для краткости будем называть эти операторы *манипулирования знаниями операторами действия*. Под действием будем понимать операцию, под операндом – аргумент операции. Тогда

<оператор действия> := <действие > (<операнд>)

<действие>:= <вызвать (инициировать) | отобразить (конвертировать) | сохранить (записать, запомнить) | найти | извлечь (прочитать) >

Для уменьшения многозначности используем английский аналог:

<action> := <CALL | CONVERT | RECORD | FIND | READ >

По сути дела, операторы описывают три типа действий:

1. Отображения слоев ФС-модели $F_i^j : S_i \rightarrow S_j$ (отображение объектов i-го слоя в объекты j-го слоя ФС-модели), им соответствует оператор <CONVERT>. Например, отображение онтологической модели (из слоя онтологических моделей) в когнитивную (из слоя когнитивных моделей); когнитивной, в свою очередь, в событийную или вероятностную (из соответствующих слоев); отображение любой модели в Геокомпонент; в этом случае для отображения:

<M_i><CONVERT><M_j>

<операнд>:= <M_O> | <M_C> | <M_E> | <M_B> | <M_G>, где M_O, M_C, M_E, M_B – соответственно онтологическая, когнитивная, событийная или вероятностная (байесовская) модели, M_G – 3D-геовизуализация результатов моделирования с помощью Геокомпонента.

В свою очередь, для отображения

$\langle Ci \rangle \langle \text{CONVERT} \rangle \langle S_j \rangle$, где сценарий ЭкС (C_i) отображается на исходную ситуацию (S_j), операндами будут наименования сценариев ЭкС и исходных ситуаций (могут извлекаться из словаря имен).

2. Вызов соответствующих инструментальных средств моделирования или экспертной системы: оператор $\langle \text{CALL} \rangle$. Для него

$\langle \text{операнд} \rangle := \langle \text{NAME} \rangle$, где в общем случае

$\langle \text{NAME} \rangle := \langle \text{библиотека} \rangle \mid \langle \text{модуль} \rangle \mid \langle \text{компонент} \rangle \mid \langle \text{приложение} \rangle \mid \langle \text{сервис} \rangle \mid \langle \text{агент} \rangle$

Для Ситуационного полигона:

$\langle \text{NAME} \rangle := \langle \text{Advice} \rangle \mid \langle \text{OntoMap} \rangle \mid \langle \text{CogMap} \rangle \mid \langle \text{EventMap} \rangle \mid \langle \text{BayNet} \rangle \mid \langle \text{Geocomponent} \rangle \mid \langle \text{Repository} \rangle$

3. Взаимодействие с Репозитарием, в котором хранятся базы знаний, словари и семантические модели (последние хранятся в формате XML). Для этого вводятся операторы:

$\langle \text{RECORD} \mid \text{FIND} \mid \text{READ} \rangle$.

Для их использования применяется ядерная конструкция (XYZ), где $\langle \text{операнд} \rangle := \langle X \rangle \mid \langle Z \rangle$, Y – оператор (один из трех, приведенных выше).

$\langle X \rangle := \langle M_O \rangle \mid \langle M_C \rangle \mid \langle M_E \rangle \mid \langle M_B \rangle \mid \langle KB_F \rangle$,

где KB_F – фрагмент базы знаний экспертной системы Advice (описание знаний: ситуаций, сценариев ЭкС, управляющих воздействий U_k).

$\langle Z \rangle := \langle M_{XML} \rangle \mid \langle D_P \rangle \mid \langle D_F \rangle$, где M_{XML} – семантическая модель в формате XML, D_P и D_F – описания знаний в нотации экспертной системы, например, правила продукций или фреймы с описаниями ситуаций и/или воздействий.

Заключение

Статья посвящена вопросам разработки языка описания знаниями и управления знаниями СМЛ в интеллектуальной системе семиотического типа на примере ситуационного управления в условиях экстремальных ситуаций в энергетике. Авторы предлагают рассматривать развиваемое ими семантическое моделирование как одно из направлений семиотического моделирования, в котором преобладает графическое представление разрабатываемых моделей. Разработка выполняется на основе онтологического инжиниринга, с использованием авторской фрактальной стратифицированной модели. Предложена архитектура Ситуационного полигона, который рассматривается как прототип интеллектуальной системы семиотического типа для ситуационного управления. Язык ситуационного управления предлагается в качестве инструмента для реализации правил изменения базовых компонентов формальной модели (что и превращает ее в семиотическую). Дается краткое описание основных конструкций предлагаемого языка СМЛ, который выполняет функции как создания и

пополнения баз знаний, так и интеграции всех компонентов Ситуационного полигона.

Результаты, представленные в статье, получены при частичной финансовой поддержке гранта Программы Президиума РАН №229 и грантов РФФИ №15-07-01284, №15-07-04074 Бел_мол_а, №16-07-00474, №16-07-00569.

Библиографический список

- [Поспелов, 1986] Поспелов Д.А. Ситуационное управление. Теория и практика. – М.: Наука, 1986. – 284 с.
- [Осипов, 2002] Осипов Г.С. От ситуационного управления к прикладной семиотике / Новости искусственного интеллекта.- 2002.-№6(54).- С. 2-12.
- [Массель А., 2010] Массель А.Г. Интеллектуальная ИТ-среда для исследований проблемы энергетической безопасности /Труды Международной конференции «Информационные технологии в науке, образовании, телекоммуникации и бизнесе». Приложение к журналу «Открытое образование». – Украина, Гурзуф, 2010. – С. 306-309.
- [Копайгородский, 2011] Копайгородский А.Н., Массель Л.В. Методы, технологии и реализация хранилища данных и знаний для исследований энергетики / Вестник Южно-Уральского государственного университета, №4 (221), 2011, серия «Математическое моделирование и программирование», вып. 7. – С. 47-55.
- [Массель, 2012] Массель Л.В., Массель А.Г. Интеллектуальные вычисления в исследованиях направлений развития энергетики // Известия Томского политехнического университета. – 2012. – Т. 321. – № 5. Управление, вычислительная техника и информатика.- С. 135-141.
- [Массель, 2013] Массель Л.В., Массель А.Г. Семантические технологии на основе интеграции онтологического, когнитивного и событийного моделирования / Материалы III международной научно-технической конференции «OSTIS-2013». – Беларусь, Минск: БГУИР, 2013. – С. 247-250.
- [Массель, 2014] Массель Л.В., Массель А.Г. Ситуационное управление и семантическое моделирование в энергетике / Труды IV Международной конференции OSTIS, Беларусь, Минск, БГУИР. 2014. – С. 111–116.
- [Массель А., 2014] Массель А.Г. Иванов Р.А. Ситуационный полигон как инструмент ситуационного управления в энергетике / Труды IV Международной конференции OSTIS. – Беларусь, Минск: БГУИР. – 2014.– С. 277-280.
- [Массель, 2015А] Массель Л.В., Массель А.Г. Методы и средства ситуационного управления в энергетике на основе семантического моделирования Труды V Международной конференции OSTIS, Беларусь, Минск: БГУИР. – 2015. – С. 199-204.
- [Массель, 2015В] Массель Л.В., Массель А.Г. Семиотический подход к созданию интеллектуальных систем ситуационного управления в энергетике. Труды XLIII Международной конференции «Информационные технологии в науке, образовании и управлении», под ред. проф. Е.Л. Глориозова. - Москва, 2015. - С. 182-193.
- [Массель А., 2015] Массель А.Г., Массель Л.В. Ситуационный полигон как интеллектуальная система семиотического типа. Труды XLIII Международной конференции «Информационные технологии в науке, образовании и управлении», под ред. проф. Е.Л. Глориозова.- Москва, 2015. С. 246-255.
- [Массель, 2015С] Массель Л.В., Массель А.Г. Язык описания и управления знаниями в интеллектуальной системе семиотического типа / Информационные и математические технологии в науке и управлении // Труды XX Байкальской Всероссийской конференции. Т. 3. – Иркутск: ИСЭМ СО РАН, 2015. – с. 112 - 124.
- [Массель, 2015D] Массель Л.В., Массель А.Г., Ворожцова Т.Н., Макагонова Н.Н. Онтологический инжиниринг ситуационного управления в энергетике /Материалы Всероссийской конференции с международным участием «Знания, онтологии, теории» (ЗОНТ-2015). Т. 2, 2015.- Новосибирск: ИМ СО РАН.- С. 36-43.

[Массель, 2016] Массель А.Г., Массель Л.В. Интеграция семиотики, когнитивной графики и семантического моделирования в интеллектуальных семиотических системах ситуационного управления. В наст. сборнике.

[Сендеров, 2013] Сендеров С.М., Рабчук В.И., Славин Г.Б., Пяткова Н.И., Чельцов М.Б. Методические рекомендации по оценке состояния энергетической безопасности РФ на федеральном уровне. – Иркутск: ИСЭМ СО РАН, 2013. – 36 с.

THE DEVELOPMENT OF SITUATIONAL MANAGEMENT LANGUAGE IN INTELLECTUAL SYSTEM OF SEMANTIC TYPE BASED ON ONTOLOGICAL ENGINEERING

Massel L.V., Massel A.G.

*Melentiev Energy Systems Institute of Siberian
Branch of the Russian Academy of Sciences
Irkutsk, Russia*

massel@isem.irk.ru

amassel@gmail.com

The article discusses the development of language for knowledge description and knowledge management in the intelligent system of semiotic type for contingency management in the energy sector. The definition of the semantic type system is entered on the basis of interpretation by D.A.Pospelov. As a tool for the implementation of the changes rules in the base set of semiotic model it's proposed Contingency Management Language (CML). It's considered a general description of CML, developed on the basis of ontological engineering, and its main parts: the component of knowledge description and component of knowledge manipulation.

Key words: semantic modeling, contingency management language (CML), intelligent system, semiotic model, ontological engineering

Introduction

The authors carried out interpretation of Pospelov ideas in terms of modern information technology, in particular, semantic modeling. The last is considered as one of the areas of semiotic modeling, where a graphic representation of the developed models is dominated. Situation polygon is proposed as the prototype of the intelligent system of semiotic type for contingency management in the energy sector. In addition to semantic modeling tools, an expert system Advice is included, which provides a comparison of situations requiring intervention, and appropriate control actions. The CML development is made on the basis of ontological engineering of contingency management problems. CML metaontology is shown and brief description of the language CML is given.

Main Part

This article discusses the development of intelligent systems semantic type. The general scheme of situational management system on the basis of the semiotic approach is described, which includes blocks Analyzer, Qualifier, Correlator, Extrapolator and Block

of random selection. The author compares the blocks of this to developed copyright collective means of semantic modeling: OntoMap as ontological modeling tool, CogMap - as cognitive modeling tool, EventMap - as event modeling tool and BayNet - as probabilistic modeling tool and developed expert system Advice.

Semantic modeling tools and Geocomponent for 3D-geovisualization of simulation results form the intelligent IT-environment, which became the prototype of the Situation polygon - intelligent system of semiotic type. CML is treated as a superstructure over the existing version of the Situation polygon and performs functions such as creating and updating of knowledge bases, and function integration of all components of the Situation polygon.

A general description of XML is given. The component of knowledge description includes 9 types of relations: naming, accessories, generalization, classification, evaluation, comparison, spatial, temporal and causal (cause and effect). The component of knowledge manipulation includes manipulation operators as CALL, CONVERT, RECORD, FIND, READ (the last three - to work with the repository). At the end CML metaontology is shown.

Conclusion

The article is devoted to the development of language for description and management of knowledge (CML) in the intelligent system of semiotic type an example of contingency management in the energy sector. The authors propose to consider the semantic modeling developed by them as one of the areas of semiotic modeling, where a graphic representation of the developed models is dominated. Development of CML is based on ontological engineering with using authoring stratified fractal model. It's proposed the architecture of the Situation polygon, which is regarded as the prototype of the intelligent system of semiotic type for contingency management in the energy sector. CML is offered as a tool for the realization of the change rules of the basic components of a formal model (this addition turns it into a semiotic model). It's given a brief description of the main structures of the proposed language CML, which performs functions such as creating and updating of knowledge bases so integration of all components of the Situation polygon.

The results presented in this article were obtained with the partial financial support of the Grant Program of the Presidium of RAS №229 and grants RFFI №15-07-01284, №15-07-04074 Bel_mol_a, № 16-07-00474, № 16-07-00569.



УДК 004.822:514

ИНТЕГРАЦИЯ СЕМИОТИКИ, КОГНИТИВНОЙ ГРАФИКИ И СЕМАНТИЧЕСКОГО МОДЕЛИРОВАНИЯ В ИНТЕЛЛЕКТУАЛЬНЫХ СЕМИОТИЧЕСКИХ СИСТЕМАХ СИТУАЦИОННОГО УПРАВЛЕНИЯ

Массель Л.В., Массель А.Г.

*Институт систем энергетики им. Л.А. Мелентьева СО РАН,
г. Иркутск, Россия*

**massel@isem.irk.ru
amassel@gmail.com**

В статье рассматриваются базовые понятия, лежащие в основе предлагаемого авторами семиотического подхода к построению интеллектуальных систем ситуационного управления в критических инфраструктурах. В предыдущих работах авторов рассматривалась современная трактовка ситуационного управления, как отображение идей Д.А. Поспелова и его учеников на современные информационные технологии. В качестве основных методов ситуационного управления авторами предлагаются методы ситуационного анализа и ситуационного моделирования, реализуемые с помощью технологий и инструментальных средств семантического моделирования в энергетике, к которому отнесены онтологическое, когнитивное, событийное и вероятностное (на основе Байесовских сетей доверия) моделирование. Данная статья посвящена дальнейшему развитию предлагаемого подхода, основанного на интеграции понятий семиотики, когнитивной графики и семантического моделирования.

Ключевые слова: семиотика, ситуационное управление, семантическое моделирование, онтологическое, когнитивное, событийное и вероятностное моделирование, когнитивная графика.

Введение

Авторы развивают высказанную ими ранее идею применения концепции ситуационного управления в энергетике и предлагают расширить область ее применения не только в энергетической, но и в других критических инфраструктурах. Предлагается семиотический подход к построению интеллектуальных систем ситуационного управления в критических инфраструктурах, основанный на базовых понятиях семиотики и семиотической модели в трактовке Д.А. Поспелова и Г.С. Осипова.

Рассматривается понятие когнитивной графики, введенное в 90-х гг. прошлого века А.А. Зенкиным и обосновывается, что графический интерфейс семантических моделей, разрабатываемых под руководством и при участии авторов (онтологических, когнитивных, событийных и вероятностных), является одним из видов когнитивной графики. Предложенный подход развивается, в частности, в рамках проекта, поддержанного грантом РФФИ №16-07-00474 «Методология построения интеллектуальных систем семиотического типа для стратегического

ситуационного управления в критических инфраструктурах». Описываются результаты первого этапа реализации предложенного подхода, в частности, онтологический инжиниринг проблемы ситуационного управления.

1. Понятие критических инфраструктур

Исследования критических инфраструктур являются достаточно молодым направлением, но становятся приоритетными во многих странах мира, и в первую очередь в США [Кондратьев, 2012]. Актуальность этих исследований усугубляется угрозами кибернетической безопасности. К критическим инфраструктурам относят энергетику, транспорт, службы по чрезвычайным ситуациям, банковский и финансовый, телекоммуникационный сектора экономики и другие жизненно важные ресурсы. В исследованиях критических инфраструктуру большое внимание уделяется выявлению ключевых объектов (или их совокупности), воздействие на которые может оказать наиболее негативный эффект на отрасль экономики, ключевой ресурс или всю инфраструктуру, а также в оценке последствий подобного воздействия и разработке механизмов

снижения таких рисков. Под энергетической инфраструктурой, которую относят к критически важным инфраструктурам, понимают совокупность энергетических объектов и систем энергетики, включая энергетические транспортные магистрали. В последнее десятилетие за рубежом активно обсуждается и развивается концепция Smart Grid или, в переводе на русский, «интеллектуальных энергетических систем - ИЭС» [Кобец, 2010]. В России также анонсирована необходимость организации работ по развитию «умных энергосистем» и ведутся соответствующие исследования. Институт систем энергетики им. Л.А. Мелентьева СО РАН (ИСЭМ СО РАН) является одним из лидеров в этой области исследований. Наряду с применением в ИЭС уже ставших традиционными методов искусственного интеллекта (нейронные сети, генетические алгоритмы, нечеткая логика), все большее внимание энергетиков привлекают так называемые «системы с целеполаганием». Это интеллектуальные системы управления, которые имеют несколько целей функционирования (или умеют генерировать эти цели), выбирая самую подходящую цель в зависимости от окружающей среды, умеют прогнозировать поведение окружающей среды и свое собственное состояние. Пока такие системы отсутствуют, как в энергетике, так и в других критических инфраструктурах России. Авторы считают, что для систем с неполной информацией и высокой сложностью объекта управления, какими являются критические инфраструктуры, все более актуальным является применение методов ситуационного управления, основанных на интеллектуальных технологиях [Массель, 2015А], и развитие семиотического подхода к построению интеллектуальных систем в рамках этого направления [Массель, 2015В].

2. Семиотический подход к построению интеллектуальных систем ситуационного управления

Основы семиотического подхода к построению интеллектуальных систем для управления техническими системами в нашей стране также заложены в работах Д.А. Поспелова. Семиотика - это наука, исследующая свойства знаков, знаковых и символьных систем, используемых в процессе коммуникации. В настоящее время в мире она активно развивается. В 2014 г. состоялся 12-й Всемирный конгресс по семиотике (Болгария, София), в котором принимали участие более 600 человек, в том числе, авторы статьи [New Semiotics, 2014]. Рассматривались всевозможные приложения семиотики в самых различных предметных областях. В областях, близких к информатике, это искусственные языки, процессы обработки информации (например, языки программирования, языки для индексирования документов, записи научно-технических фактов и знаний), алгоритмы, обеспечивающие обработку текстов на

естественном языке (машинный перевод, автоматическое индексирование и реферирование, перевод с естественного языка на формальный язык), составления картографических изображений, специальных схем и планов др. Ведущими учеными-семиотиками (А. Соломоник, (Израиль), А. Володченко (Германия), К. Банков (Болгария) и др.) отмечалось недостаточное распространение семиотики в технических областях, а точнее, практическое отсутствие таких работ.

Д.А. Поспеловым было введено определение семиотической модели [Поспелов, 1986], которое потом было детализировано Г.С. Осиповым [Осипов, 2002]. Согласно последнему, семиотическую модель можно определить как восьмерку:

$$W = \langle T, R, A, P, \tau, \rho, \alpha, \pi \rangle,$$

где T - множество основных символов;
 R - множество синтаксических правил;
 A - множество знаний о предметной области;
 P - множество правил вывода решений (прагматических правил);
 τ - правила изменения множества T ;
 ρ - правила изменения множества R ;
 α - правила изменения множества A ;
 π - правила изменения множества P .

Иначе говоря, в отличие от формальных моделей, в которых элементы, образующие множество T , обладают жестким синтаксисом, жесткой семантикой и жесткой прагматикой, в семиотической модели все эти свойства элементов множества T становятся доступными для изменения; именно такой особенностью обладают знаки - элементы знаковых, или семиотических систем, изучаемых в семиотике. Такие системы тесно связаны со всей человеческой деятельностью, именно изменчивость и условность знаков делают эту деятельность эффективной.

Следует отметить, что семиотический подход в нашей стране развивается, в частности, в работах Осипова Г.С. [Осипов, 2002], Вагина В.Н., Еремеева А.П. (интеллектуальные системы реального времени семиотического типа) [Вагин, 2001], [Еремеев, 2014], Кулинича А.А. (семиотическая модель когнитивного опыта) [Кулинич, 2014] и др.

Перечисленные работы носят преимущественно теоретический характер. Сведений о развитии этого подхода в области энергетики и применительно к системам ситуационного управления авторам найти не удалось.

3. Когнитивная графика и визуальная аналитика

Термин «когнитивная графика» был введен в российской науке А.А. Зенкиным в 90-х гг. прошлого столетия и первоначально связывался с новыми возможностями визуализации результатов научных исследований, обусловленными

появлением новых графических средств персональных компьютеров [Зенкин, 1991]. Впоследствии было осознано, что графические образы могут активизировать ассоциативную логику подсознательных процессов мышления человеческого мозга, что позволяет с помощью когнитивной графики быстро находить оригинальные и зачастую неожиданные решения. По сути дела, семантические модели обладают свойствами когнитивной графики и особенно полезны для специалистов с преобладающим образным мышлением. Следует отметить, что семантические модели позволяют выполнить качественный анализ решаемой задачи. Для количественного обоснования предлагаемых решений необходимо привлекать результаты математического моделирования, которые, кроме того, представляются более убедительными для специалистов с преобладающим логическим мышлением. Тем не менее, для иллюстрации математических решений также может использоваться когнитивная графика, что и было успешно продемонстрировано А.А. Зенкиным в его работах.

Появление термина «визуальная аналитика» (Visual Analytics) датируют 2004 г., связывая его с книгой Д. Томаса «Освещающая путь, программа исследований и разработок по Visual Analytics», выпущенной как программный документ Тихоокеанской северо-западной национальной лаборатории, входящей в число шестнадцати лабораторий Министерства энергетики США. В этой книге визуальная аналитика определяется как умение мыслить аналитически, поддержанное графическим интерфейсом. Visual Analysis имеет как минимум две трактовки. Согласно первой так называют широкую круг междисциплинарных исследований, так или иначе связанных с применением интерактивных визуальных средств для анализа данных. Второе значение распространяется на собственно аналитический компонент Visual Analytics, состоящий из автоматической и интерактивной частей. Связь визуальной аналитики с другими областями применения информационных технологий (ИТ) рассматривается в [Черняк, 2013], где показано, что визуальная аналитика лежит на пересечении двух областей: автоматического анализа и визуализации данных.

Таким образом, представление семантических моделей в графическом виде позволяет отнести их к одному из видов когнитивной графики и к одной из областей визуальной аналитики – научной аналитике, которая является неотъемлемой частью интеллектуальных систем ситуационного управления.

4. Семантическое моделирование как современная трактовка семиотического моделирования

Д.А. Поспелов считал близкими понятиями метод ситуационного управления, семиотическое моделирование, семиотическое управление, логико-лингвистические методы управления, поскольку, по его трактовке, они все основаны на введении понятия ситуации, классификации ситуаций и их преобразовании.

Модель пространства знаний для ситуационного управления в энергетике [Массель, 2015D] представлена на рис. 1.

Согласно Поспелову, описание полной ситуации есть набор аксиом. Знания – это не только множество всех текущих ситуаций, но и способы перехода от одного описания объекта к другому, способы изменения компонентов формальной системы, а, следовательно, знания – это то, что воплощается в некоторой семиотической модели.

Учитывая, что наличие факторов неопределенности усложняет адекватную оценку состояния объекта и среды, авторами предложено использовать семантические технологии ситуационного анализа и моделирования, к которым отнесены онтологическое, когнитивное, событийное и вероятностное моделирование [Массель, 2013].

По сути дела, семантическое моделирование предлагается рассматривать как одно из направлений семиотического моделирования, в котором преобладает графическое представление разрабатываемых моделей, с элементами когнитивной графики. Для реализации требований семиотической модели (введения правил изменения базовых множеств) предлагается использовать язык ситуационного управления. Представляют интерес две гипотезы, выдвинутые Поспеловым, которые обосновывают необходимость разработки такого языка [Поспелов, 1986].

Гипотеза 1. Вся информацию об объекте можно выразить средствами обычного естественного языка. Следствием этой гипотезы является тот интерес к естественному языку и его изобразительным средствам, который характерен для метода ситуационного управления. Для ЭВМ возникает проблема сужения текстов на естественном языке, описывающих сам объект управления и опыт по управлению им, до такого уровня, чтобы полученные описания можно было «погрузить» в семиотическую систему.

Гипотеза 2. Всякий текст на естественном языке, относящийся к тому, о чем говорится в гипотезе 1, можно перевести на формальный язык семиотической модели

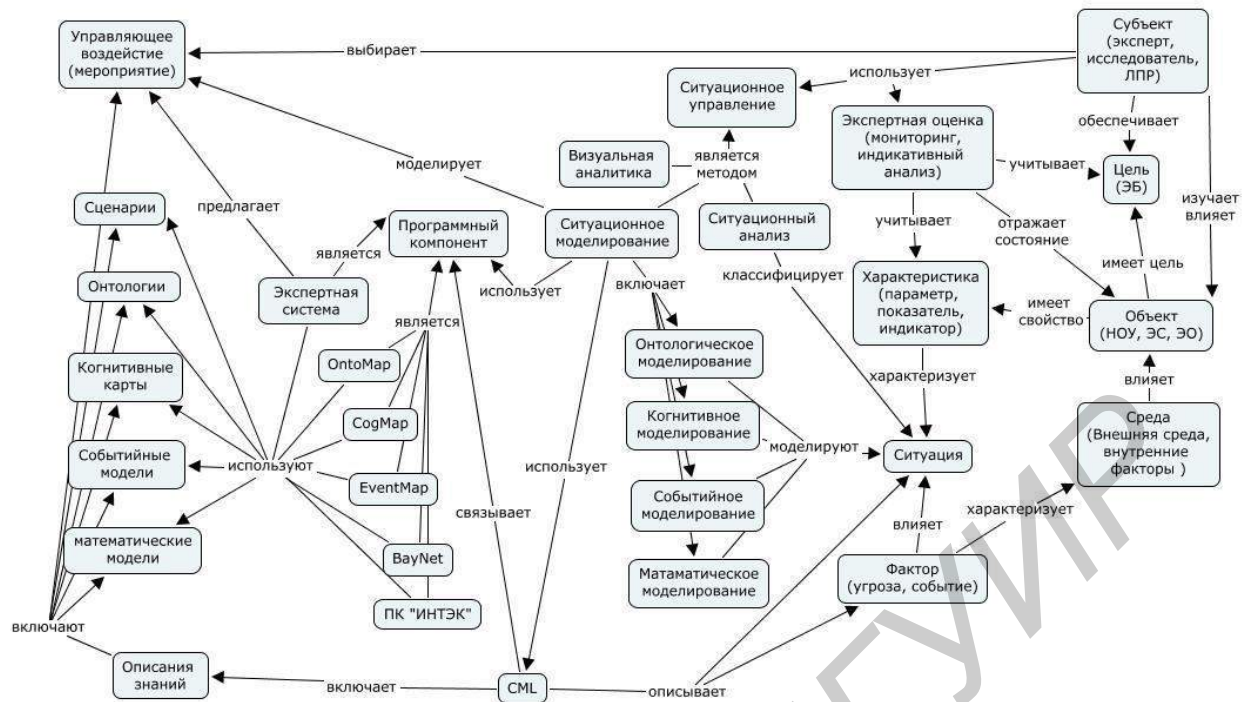


Рисунок 1 -- Модель пространства знаний для ситуационного управления в энергетике

Пока неочевидна необходимость и не проверена возможность разработки универсального языка ситуационного управления. Скорее всего, он будет включать как базовое (универсальное) ядро, так и проблемно-ориентированные компоненты.

Авторами разработана концепция такого языка, ориентированного на управление в экстремальных ситуациях в энергетике (CML – Contingency Management Language) [Массель, 2015С]. При разработке CML авторы опирались на систему онтологий, построенную в результате онтологического инжиниринга проблемы ситуационного управления в энергетике, рассмотренного ниже. Современное состояние разработки отражено в статье в настоящем сборнике [Массель, 2016].

5. Онтологический инжиниринг проблемы ситуационного управления в энергетике

Онтологический инжиниринг включает выявление: основных классов сущностей (базовых понятий) в описании реальных взаимодействующих процессов, отношений между этими классами, а также совокупности свойств, которые определяют их изменение и поведение во взаимодействии.

Целями онтологического инжиниринга [Черняховская, 2015] являются: повышение уровня интеграции информации, необходимой для принятия управленческих решений; повышение эффективности информационного поиска; предоставление возможности совместной обработки знаний на основе единого семантического описания.

В результате онтологического инжиниринга проблемы ситуационного управления построена система онтологий, которая включает метаонтологию ситуационного управления (рис. 2), метаонтологию пространства знаний в области ситуационного управления (рис. 1), онтологии ситуаций, ситуационного анализа и ситуационного моделирования, метаонтологию CML и онтологии его базовых компонентов: компонента описания знаний и компонента манипулирования знаниями [Массель, 2015D].

Заключение

В статье рассмотрен подход к построению интеллектуальных систем ситуационного управления в критических инфраструктурах. Подход основан на обобщении методов, разработанных авторами для построения таких систем ситуационного управления в энергетике. Как основу подхода, предлагается использовать интеграцию семиотики, когнитивной графики и семантического моделирования. В статье рассмотрены эти понятия, приведена семиотическая модель и обоснована необходимость разработки языка ситуационного управления, необходимого для реализации такой модели. Разработку языка предлагается выполнять на основе онтологического инжиниринга проблемы ситуационного управления, результаты которого кратко перечислены в статье.

Результаты, представленные в статье, получены при частичной финансовой поддержке гранта Программы Президиума РАН №229 и грантов РФФИ №15-07-01284, №15-07-04074 Бел_мол_а, № 16-07-00474, № 16-07-00569.

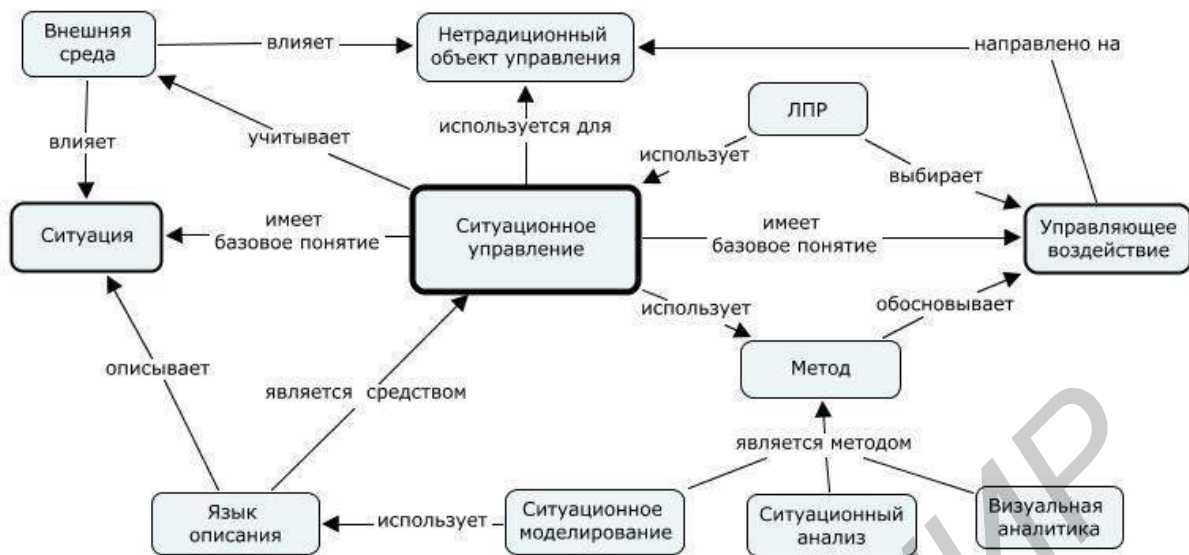


Рисунок 2 -- Метаонтология ситуационного управления

Библиографический список

[Кондратьев, 2012] Кондратьев А. Современные тенденции в исследовании критической инфраструктуры в зарубежных странах http://pentagonus.ru/publ/sovremennye_tendencii_v_issledovanii_kriticheskoi_infrastruktury_v_zarubezhnoj_stranakh_2012/19-1-0-2082 (дата обращения 7.09.2015)

[Кобец, 2010] Кобец Б.Б., Волкова И.О. Инновационное развитие электроэнергетики на базе концепции Smart Grid. – М.: ИАЦ Энергия, 2010. – 208 с.

[Массель, 2015А] Массель Л.В., Массель А.Г. Методы и средства ситуационного управления в энергетике на основе семантического моделирования//Труды V Международной конференции OSTIS, Беларусь, Минск: БГУИР. - 2015.- С.199-204.

[Массель, 2015В] Массель Л.В., Массель А.Г. Семантический подход к созданию интеллектуальных систем ситуационного управления в энергетике. Труды XLIII Международной конференции «Информационные технологии в науке, образовании и управлении», под ред. проф. Е.Л. Глорозова.- Москва, 2015. - С. 182-193.

[New Semiotics, 2014] New Semiotics: Between Tradition and Innovation / Abstracts.- Bulgaria, Sofia: New Bulgarian University. Southeast European Center For Semiotic Studies, 2014.- 218 p.

[Поспелов, 1986] Поспелов Д.А. Ситуационное управление. Теория и практика. – М.: Наука, 1986. – 284 с.

[Осипов, 2002] Осипов Г.С. От ситуационного управления к прикладной семиотике / Новости искусственного интеллекта.- 2002.-№6(54).- С. 2-12.

[Вагин, 2001] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // Изв. РАН. Теория и системы управления. – 2001. – №6. – С. 114-123

[Еремеев, 2014] Еремеев А.П. Применение обучения с подкреплением в интеллектуальных системах реального времени / Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'14». – М.: Физматлит, 2014. – Т. 1. – С. 263-271.

[Кулинич, 2014] Кулинич А.А. Семантическая когнитивная архитектура СИПР / Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'14». – М.: Физматлит, 2014. – Т.2. – С. 146-153.

[Зенкин, 1991] Зенкин А.А. Когнитивная компьютерная графика.- М.: Наука, 1991.- 192 с.

[Черняк, 2013] Черняк Л. Визуальная аналитика и обратная связь. «Открытые системы», №6, 2013. URL: <http://www.osp.ru/os/archive/> (Дата обращения 02.12.2014)

[Массель, 2013] Массель Л.В., Массель А.Г. Семантические технологии на основе интеграции онтологического,

когнитивного и событийного моделирования / Материалы III международной научно-технической конференции OSTIS-2013. – Беларусь, Минск: БГУИР, 2013. – С. 247-250.

[Массель, 2015С] Массель Л.В., Массель А.Г. Язык описания и управления знаниями в интеллектуальной системе семиотического типа / Информационные и математические технологии в науке и управлении // Труды XX Байкальской Всероссийской конференции. Т. 3. – Иркутск: ИСЭМ СО РАН, 2015. – с.112 - 124.

[Черняховская, 2015] Черняховская Л.Р., Федорова Н.И. Ситуационный подход к управлению взаимодействием сложных процессов на основе онтологического инжиниринга /Труды XX Байкальской Всероссийской конференции «Информационные и математические технологии в науке и управлении». Часть III.- Иркутск: ИСЭМ СО РАН, 2015.- 261 с.

[Массель, 2015D] Массель Л.В., Массель А.Г., Ворожцова Т.Н., Макагонова Н.Н. Онтологический инжиниринг ситуационного управления в энергетике /Материалы Всероссийской конференции с международным участием «Знания, онтологии, теории» (ЗОИТ-2015). Т. 2, 2015.- Новосибирск: ИМ СО РАН.- С. 36-43.

[Массель, 2016] Массель А.Г., Массель Л.В. Разработка языка ситуационного управления в интеллектуальной системе семиотического типа на основе онтологического инжиниринга. В наст. сборнике

INTEGRATION OF SEMIOTICS, COGNITIVE GRAPHICS AND SEMANTIC MODELING IN INTELLIGENT SEMIOTIC SYSTEM FOR SITUATIONAL MANAGEMENT

Massel L.V., Massel A.G.

*Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences
Irkutsk, Russia*

massel@isem.irk.ru

amassel@gmail.com

This article discusses the basic concepts underlying the author's proposed semiotic approach to building of intelligent systems for situational management in critical infrastructures. In previous work, the author considers the modern interpretation of situational management, as a reflection of the ideas of D.A.

Pospelov and his colleagues in the modern information technology. The main methods of situational management offered by the authors are methods of situational analysis and situational modeling, implemented with the help of technologies and the tools of semantic modeling in the energy sector, into which the ontological, cognitive, event and probability (based on Bayesian belief networks) simulation. This article focuses on the further development of the proposed approach based on the integration of semiotics, cognitive graphics and semantic modeling.

Keywords: semiotics, contingency management, semantic modeling, ontological, cognitive, event and probabilistic modeling, cognitive graphics.

Introduction

The authors develop their earlier expressed the idea of using the concept of situational management in the energy sector and offer to extend its scope, not only in energy, but also in other critical infrastructure.

It's proposed semiotic approach to building intelligent systems of situational management in critical infrastructures, based on the basic concepts of semiotics and semiotic models in the interpretation by D.A. Pospelov and G.S. Osipov.

The concept of cognitive graphics, introduced in the 90s of last century by A.A. Zenkin and it's proved that GUI semantic models developed under the guidance and with the participation of authors (ontological, cognitive, event and probability), is a type of cognitive graphics. The results of the first phase of the proposed approach are described, in particular, the problem of ontological engineering of situational management.

Main Part

The article deals with the concept of critical infrastructures. Researches of these infrastructures become a priority in many countries, and especially in the United States. The relevance of these studies is compounded by threats of cybersecurity. Critical infrastructures include energy, transport, emergency services, banking and finance, telecommunications sector and other vital resources.

The authors consider the concept of semiotics and semiotic approach to building of intelligent systems for situational management in critical infrastructures. The approach is based on the definition of the semiotic model introduced by DA Pospelov. It differs from the classical formal model that in addition to the base set (T, R, A, P) is proposed to introduce rules to change the elements of each set. The basic sets are determined as the alphabet (T), syntax (R), semantics (A) and pragmatics (P) of models.

The concepts of visual analytics and cognitive graphics are considered. It is alleged that the representation of semantic models in graphical form allows you to assign them to one of the kinds of cognitive graphics and one of the areas of visual

analytics - Research Analytics, which is an impotent part of intelligent systems for situational management.

The semantic modeling is proposed as one of the areas of semiotic modeling, where a graphic representation of the developed models is dominated.

To implement the requirements of the semiotic model (introduction of changes in the rules of basic sets) we propose to use the language of situational management. The authors have developed the concept of this language oriented on the management in extreme situations in the energy sector (CML - Contingency Management Language). In the CML development authors relied on a system of ontologies, built on the basis of ontological engineering for the problems of contingency management in the energy sector. The examples of ontologies are given in the article.

Conclusion

The article describes the approach to building of intelligent systems for situational management in critical infrastructures. The approach is based on a generalization of methods developed by the authors for the construction of such systems for situational management in the energy sector. It's proposed to use the integration of semiotics, cognitive graphics and semantic modeling as the basis of this approach. The article discusses these concepts. The semiotic model is given and it's shown the necessity to develop the language of situational management required to implement such a model. The development of language is offered to perform on the basis of ontological engineering of situational management problems, the results of which are summarized in the article.

The results presented in this article were obtained with the partial financial support of the Grant Program of the Presidium of RAS №229 and grants RFFI №15-07-01284, №15-07-04074 Bel_mol_a, № 16-07-00474, № 16-07-00569.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891

ЕДИНАЯ ОНТОЛОГИЧЕСКАЯ ПЛАТФОРМА ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Филиппов А.А., Мошкин В.С., Шалаев Д.О., Ярушкина Н.Г.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

al.filippov@ulstu.ru

PostForVadim@yandex.ru

do.shalaev@ulstu.ru

jng@ulstu.ru

В данной работе предложена методика разработки, а также основные принципы построения и архитектура единой платформы интеллектуального анализа данных на основе онтологии предметной области. Помимо этого рассмотрен процесс построения предметной онтологии с помощью встроенного в платформу универсального редактора на примере онтологии «Справочник боцмана».

Ключевые слова: онтология, платформа, data mining, семантика.

Введение

В процессе деятельности любой современной крупной организации возникает необходимость в проведении интеллектуального анализа данных. Для проведения такого анализа от специалиста требуется глубокое знание предметной области, умение использовать различные инструменты интеллектуального анализа и знание современных IT-технологий для адаптации существующих решений под особенности конкретной предметной области.

Желание максимально автоматизировать процесс обработки данных с учетом специфики предметной области вызывает потребность в едином универсальном инструментарии интеллектуального анализа данных, решающего различные задачи, основанные на накопленных знаниях и опыте экспертов в этой области:

- построение экспертной системы, обеспечивающей вывод рекомендаций в ходе принятия управленческих решений;
- моделирование и прогнозирование показателей деятельности компании по данным предыдущих периодов отчетности;
- поиск необходимой документации в объемном архиве организации с учетом ее специфики и т. д.

Современное производство нуждается в создании адекватных методов и инструментальных средств повышения эффективности и качества

выпускаемой продукции. Для этого необходимы соответствующие модели и методы, ориентированные на описание объектных структур рассматриваемой предметной области, позволяющие идентифицировать, анализировать (включая наглядное представление) и манипулировать всем многообразием объектов и отношений, имеющихся в предметной области.

Ряд научных проблем, стоящих перед современными организациями и требующих системного решения:

- необходимость разработки семантического базиса анализа содержимого электронного хранилища информации;
- отсутствие интегративных концептуальных моделей, использующих различные подходы хранения знаний о предметной области;
- необходимость универсализации процесса автоматизированной обработки хранимых знаний;
- необходимость одновременного использования разноаспектных описаний особенностей рассматриваемой предметной области;
- необходимость решения проблемы учета нечеткости в человеческих рассуждениях.

Решение данных проблем может основываться на применении интеллектуальных методов и алгоритмов анализа содержимого электронного хранилища предприятия с целью организации логического вывода рекомендаций для поддержки

принятия управленческих, экономических и технических решений с учетом модели конкретной предметной области.

В связи с этим, в настоящее время актуальной является задача обеспечения специалистов организаций различных профилей универсальным инструментарием, позволяющим решать поставленные задачи с учетом специфики конкретной предметной области.

1. Онтология – как база знаний современной компании

Современные производственные организации обладают значительными по объему электронными хранилищами различного рода информации, связанной с деятельностью данного предприятия. Фактически, такое электронное хранилище содержит в себе историю предприятия, опыт и знания большого количества высококвалифицированных специалистов.

В работах [Норенков, 2009] и [Титов, 2009] отмечается, что при увеличении объема электронного хранилища затрудняется анализ его содержимого по заранее заданным реквизитам, а от лиц, принимающих решения, требуются навыки в области семантической обработки большого объема информации, а также глубокие знания предметной области. В результате опыт и знания, зафиксированные в таких хранилищах, остаются невостребованными, что приводит к принятию неверных управленческих, экономических и технических решений.

Учет специфики предметной области приводит к необходимости формирования прикладной онтологии особой структуры, включающей в себя систему понятий предметной области, семантические отношения между ними и функции интерпретации. Ведущие исследователи в области онтологических систем отмечают актуальность исследований, основанных на онтологическом подходе [Добров и др., 2006] [Гаврилова и др., 2000]. В трудах данных исследователей отмечается важность использования онтологического инжиниринга в процессе интеллектуального анализа данных.

Наиболее универсальной и полной с точки зрения охвата специфики предметной области во всем разнообразии отношений между ее объектами является модель интеграции онтологического и продукционного подходов представления знаний, позволяющая в процессе логического вывода соответствующих рекомендаций опираться на данные, представленные в виде онтологии и наиболее полно описывающие рассматриваемую предметную область.

В широком смысле, онтологии – это модели, являющиеся формой представления знаний о предметных областях в виде семантических информационно-логических сетей взаимосвязанных

объектов, где в качестве главных элементов выступают понятия предметной области с их свойствами и отношения между объектами. Онтологии выполняют интегрирующую функцию, обеспечивая общий семантический базис в процессах принятия решений, интеллектуального анализа данных и единую платформу для объединения разнообразных информационных систем.

В данной работе предлагается подход к построению единой платформы проведения интеллектуального анализа данных, включающей в себя объектно-ориентированные, а также нечеткие модели и программные инструментальные средства работы с ними, с использованием онтологического системного анализа.

2. Общее описание платформы

2.1. Цели и задачи исследования

Базовой целью данного исследования является разработка универсальной web-ориентированной платформы, позволяющей:

1. Решать широкий круг задач на основе интеллектуального анализа данных (индексация, классификация, кластеризация, информационный поиск).
2. Адаптировать платформу к конкретной предметной области с помощью методов онтологического анализа и инженерии знаний (разработка, хранение и использование модели предметной области в виде прикладной онтологии с возможностью импорта в различные форматы).
3. Осуществлять экспертную оценку и информационную поддержку в принятии решений (использование механизмов дискрипционной логики).

На настоящий момент было разработано ядро онтологической платформы, обеспечивающее автоматизированное формирование онтологии предметной области, визуализацию ее структуры, а также процесс информационного поиска знаний. Помимо этого, были поставлены следующие задачи:

1. Расширение функционала платформы для осуществления функций контроля прав доступа, организации хранения исходных и полученных данных.
2. Предоставление конечным пользователям возможности использования платформы без специальной подготовки, разработка интуитивно-понятного интерфейса.
3. Предоставление конечным пользователям средств адаптации платформы под особенности конкретной предметной области.
4. Предоставление разработчикам возможности расширения функционала платформы без значительных временных затрат на изучение архитектуры платформы с помощью механизма плагинов.

В настоящее время не существует полных аналогов разрабатываемой платформы. Косвенным аналогом можно считать веб-ориентированную систему ClowdFlows, позволяющую пользователю настраивать этапы проведения интеллектуального анализа данных, формировать модели данных и проводить эксперименты.

Однако для расширения функционала данной системы и адаптации под конкретную предметную область необходимо использовать веб-сервисы WSDL. Данный подход является достаточно сложным для использования неподготовленными пользователями, так как требует определенных знаний в IT-сфере.

2.2. Архитектура платформы

Архитектура платформы интеллектуального анализа представлена на рисунке 1.

Как видно из схемы, серверная часть платформы реализована на языке программирования Java с использованием фреймворка Spring, который предоставляет широкие возможности для облегчения разработки корпоративных приложений.

Интерфейс платформы реализован с применением фреймворка Google Web Toolkit (GWT), который позволяет создавать RIA (Rich Internet Application) Ajax-приложения на основе Java-кода путем его компиляции в JavaScript-код.

Использование GWT позволяет использовать типобезопасный язык Java при разработке, что снижает количество потенциальных ошибок и делает возможным отладку кода интерфейсной части платформы. При этом на клиенте данный код выполняется в виде легковесного JavaScript-кода, который поддерживается всеми современными браузерами.

Платформа выполняется в контейнере сервлетов Jetty. Данный контейнер имеет модульную архитектуру, что позволяет использовать только

необходимый возможности [Wilkins, 2008], снижая нагрузку на сервер.

Также Jetty хорошо масштабируется для обслуживания многих соединений, со значительным временем простоя между запросами и позволяет обслуживать большее количество пользователей.

В основу платформы интеллектуального анализа данных положен принцип микросервисной архитектуры [Lewis et. Al, 2014].

Микросервисная архитектура – подход, при котором единое приложение является распределенным и состоит из небольших сервисов, которые тесно взаимодействуют между собой, при этом каждый такой сервис работает в собственном процессе. Для осуществления взаимодействия сервисов часто используются легковесные механизмы удаленного вызова процедур.

Каждый сервис, как правило, реализует конкретный бизнес-процесс и выполняется независимо в изолированном контейнере. Данный подход позволяет реализовывать сервисы с помощью различных языков программирования и использовать различные хранилища данных.

Такая изолированность сервисов позволяет:

1. Повысить общую отказоустойчивость системы за счет выполнения сервисов в разных адресных пространствах.
2. Повысить масштабируемость системы за счет запуска нескольких экземпляров сервисов и балансировки нагрузки между ними.
3. Предоставить возможность использования различных операционных систем, языков программирования, технологий хранения данных и т. д., что позволяет использовать наилучшее решение поставленной задачи.
4. Уменьшить время простоя системы при внесении изменений, исправлении ошибок и других сервисных задач.

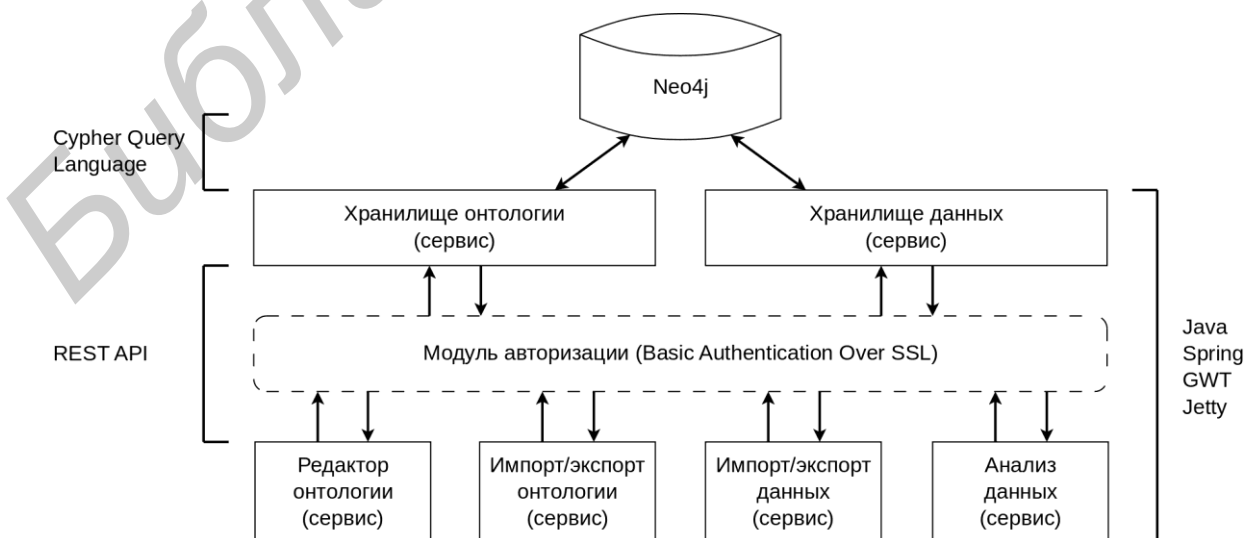


Рисунок 1 – Архитектура платформы интеллектуального анализа данных

Для взаимодействия сервисов в рамках рассматриваемой платформы используется механизм взаимодействия REST (Representational State Transfer – «передача репрезентативного состояния»), при котором вызов удаленной процедуры представляет собой обычный HTTP-запрос (GET, POST, PUT и т. д.), а необходимые данные передаются в качестве параметров запроса.

В качестве хранилища онтологий платформы интеллектуального анализа используется графовая база данных Neo4j. Данная СУБД обладает следующими преимуществами:

1. Нативный формат хранения графов.
2. Один экземпляр СУБД может обслуживать графы с миллиардами узлов и связей.
3. Может обрабатывать графы, которые полностью не помещаются в оперативной памяти. Графо-ориентированный язык запросов – Cypher.

Все перечисленные ресурсы, приложения и технологии являются бесплатными и свободно распространяемыми.

2.3. Процесс построения предметной онтологии

Рассмотрим процесс построения предметной онтологии с помощью встроенного в платформу универсального редактора на примере онтологии «Справочник боцмана».

Очень часто человек, плохо знакомый с предметной областью имеет информационную потребность, которую необходимо удовлетворить [Наместников и др, 2014].

В [Наместников и др, 2014] дано следующее определение информационной потребности пользователя: «Информационная потребность – это информационная неопределенность, которую

пользователь хочет уменьшить посредством получения информации из системы информационного поиска». В нашем случае такой системой будет являться платформа интеллектуального анализа данных.

Все необходимые для удовлетворения информационной потребности пользователя данные содержатся в онтологии предметной области, которую, как правило, формируют эксперты данной предметной области.

Рассмотрим подробно данный процесс на примере онтологии «Справочник боцмана».

Корабль – сложный объект, состоящий из большого числа составных узлов и имеющий различные характеристики. У новичков в данной предметной области часто возникают вопросы, например, о том якорную цепь какого калибра, с какой разрывной нагрузкой и массой на 1 метр цепи установить на корабль определенного водоизмещения. Ответы на данные вопросы можно узнать, используя встроенный в редактор онтологии механизм информационного поиска знаний (рисунок 2), работа которого основана на содержимом онтологии предметной области (рисунки 3 и 4).

Рисунок 2 иллюстрирует пример запроса к системе для удовлетворения информационной потребности пользователя.

Из рисунка 3 видно, что встроенный в платформу интеллектуального анализа данных редактор онтологий позволяет создавать «схему» онтологии, которая состоит из классов и отношений между ними, а также заполнять данную онтологию знаниями, которые представлены индивидуалами (объекты классов) с определенными состояниями (объекты отношений) (рисунок 4).

Запрос: Надводный_корабль_водоизмещением_4035_т Оснащен_якорной_цепью

Результат запроса: Надводный_корабль_водоизмещением_4035_т Оснащен_якорной_цепью Якорная_цепь_19_калибра;

Параметры запроса:

Найти

Корпус_корабля
Якорная_цепь
Корабль
Якорь
Якорная_цепь_19_калибра
Литой_якорь_Матросова_600_кг
Стандартный_корпус_надводного_корабля
Якорь_Холла_1500_кг
Надводный_корабль_водоизмещением_4035_т

Имеет_водоизмещение
Имеет_вспомогательный_якорь
Имеет_корпус
Оснащен_якорной_цепью
Имеет_становой_якорь

Рисунок 2 – Окно модуля информационного поиска. Пример запроса к системе.

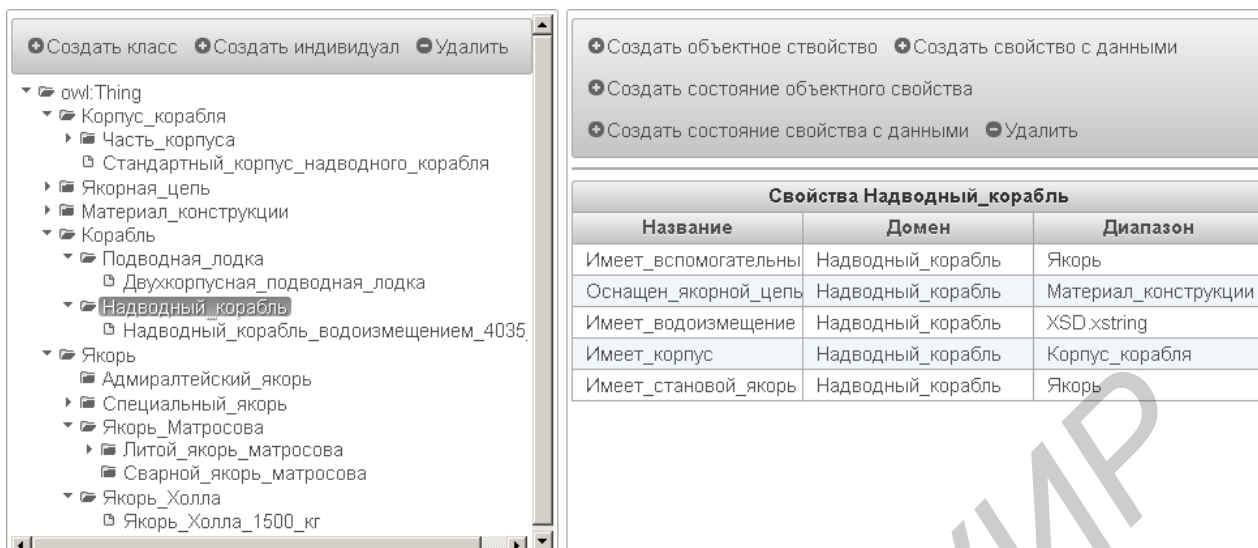


Рисунок 3 – Окно редактора онтологий. Класс «Надводный корабль» и его свойства

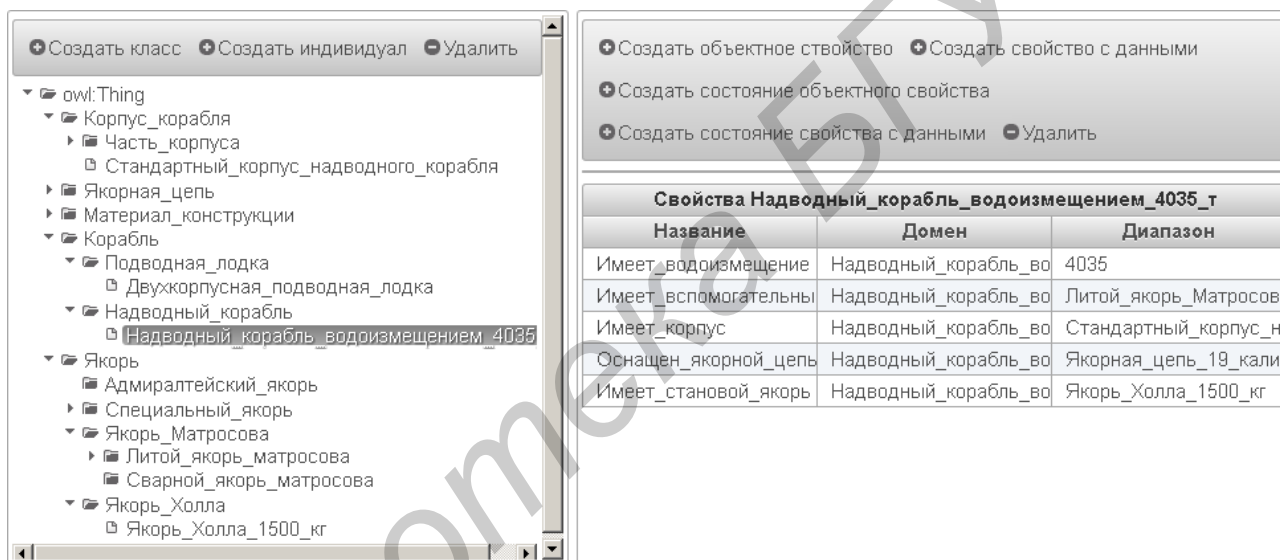


Рисунок 4 – Окно редактора онтологий. Индивидуал «Надводный корабль водоизмещением 4035 тонн» и его состояния

Используемая в данном примере предметная онтология была построена экспертом на основе специализированного справочника боцмана и включает в себя порядка 30 классов, 93 отношения и свыше 120 объектов классов.

Заключение

Таким образом, основное преимущество описанной платформы — наличие единого технологического и информационного пространства процесса принятия решения, информационного поиска с помощью интеллектуального анализа данных, обеспечивающих возможность использования накопленных экспертных знаний о специфике предметной области организации для решения самых различных задач.

Основными направлениями развития рассмотренной платформы являются:

- наращивание функционала путем добавления новых механизмов проведения интеллектуального анализа данных;
- адаптация и унификация ранее реализованных алгоритмов интеллектуального анализа под особенности разработанной платформы с целью их последующего совместного применения при решении широкого круга сложных задач;
- развитие платформы для последующего использования в различных научно-исследовательских проектах;
- интеграция с существующими средами и платформами хранения и обработки данных и знаний, используемых в производстве и других сферах деятельности.
- адаптация платформы под задачи анализа больших объемов данных (Big Data).

Библиографический список

[Добров и др., 2006] Добров Б.В., Лукашевич Н.В., Лингвистическая онтология по естественным наукам и технологиям: основные принципы разработки и текущее состояние // Десятая национальная конференция по искусственному интеллекту с международным участием (Обнинск, 25-28 сентября 2006 г.) –М.: Физматлит, 2006.

[Гаврилова и др., 2000] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. –СПб.: Питер, 2000. – 384 с.

[Норенков, 2009] Норенков И.П. Основы автоматизированного проектирования. М: МГТУ имени Баумана, 2009.

[Титов, 2009] Титов Ю. А. САПР технологических процессов. Ульяновск, 2009.

[Wilkins, 2008] Greg Wilkins Jetty vs Tomcat: A Comparative Analysis, 2008. URL – <http://www.webtide.com/choose/jetty.jsp>.

[Lewis et al, 2014] James Lewis, Martin Fowler Microservices definition of this new architectural term, 2014. URL – <http://martinfowler.com/articles/microservices.html>.

[Наместников и др., 2014] Наместников А.М., Субхангулов Р.А. Онтологически-ориентированная модель классификаций текстовых документов // Материалы IV международной научно-технической конференции OSTIS-2014, Минск. – 2014. С. 385–390.

UNIFORM ONTOLOGICAL DATA MINING PLATFORM

Filippov A.A., Moshkin V.S., Shalaev D.O.,
Yarushkina N.G.

Ulyanovsk State Technical University, Russian Federation

al.filippov@ulstu.ru

PostForVadim@yandex.ru

do.shalaev@ulstu.ru

jng@ulstu.ru

Methodology of development, basic principles of construction and architecture of a uniform data mining platform based on ontology describe in this article. Also process of building ontology using the built-in universal platform editor of the example ontology "Handbook boatswain" describes.

Introduction

The desire to automate the process of data specific to the subject area is the need for a single universal data mining tools, to solve various problems, based on the accumulated knowledge and experience of experts in the field:

- creation of the expert system providing an output of recommendations during acceptance of administrative decisions;
- simulation and prediction of indexes of activities of the company according to the previous periods of the reporting;
- search of necessary documentation in volume archive of the organization taking into account its specifics etc.

In this regard, now the task of support of specialists of the organizations of different profiles the universal tools allowing to solve objectives taking into account specifics of specific data domain is actual.

Main Part

The server part of a platform is realized in the Java programming language with use of a framework of Spring which gives ample opportunities for facilitation of development of corporate applications.

The interface of a platform is realized with application of a framework of Google Web Toolkit (GWT) which allows to create RIA (Rich Internet Application) of an Ajax-application on the basis of a Java-code by its compilation in a JavaScript-code.

Use of GWT allows to use Java language by development that reduces quantity of potential errors and does possible debugging of a code of the interface part of a platform. Thus on the client this code is executed in the form of a light-weight JavaScript-code which is supported by all modern browsers.

The principle of microservice architecture is the basis for a platform of intellectual data analysis.

The microservice architecture is a approach in case of which uniform application is distributed and consists of small services which tightly interact among themselves, thus each such service works in own process. For implementation of interaction of services light-weight mechanisms of a remote call of procedures are often used.

Each service realizes specific business process and is executed independent in the isolated container. This approach allows to realize services by means of different programming languages and to use different data stores.

All listed resources, applications and technologies are free and freely spreaded.

Conclusion

The main directions of development of the platform are:

- development of a functionality by adding of new mechanisms of carrying out intellectual data analysis;
- adaptation and unification of earlier realized algorithms of the intellectual analysis under features of the developed platform for the purpose of their subsequent combined use in case of the solution of a wide range of complex challenges;
- development of a platform for the subsequent use in different scientific исследовательских projects;
- integration with the existing environments and platforms of storage and data handling and the knowledge which are used in production and other fields of activity.
- adaptation of a platform under tasks of the analysis of large volumes of data (Big Data).



УДК 004.822:514

АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ОНТОЛОГИЧЕСКОГО АНАЛИЗА ДАННЫХ

Зубцов Р.О. *, Семенова В.А. *, Смирнов С.В. **

*Самарский государственный аэрокосмический университет им. С.П. Королёва
(научно-исследовательский университет), г. Самара, Россия

zubtsov.r@gmail.com

queenbfjr@gmail.com

**Институт проблем управления сложными системами
Российской академии наук, г. Самара, Россия

smirnov@iccs.ru

В статье рассматриваются математические модели, методы и алгоритмы онтологического анализа данных, составляющие аналитическое ядро разрабатываемого инструментария семантического моделирования и проектирования, и предназначенные для вывода формальных онтологий на основе эмпирических объектно-признаковых данных, организованных в виде соответствия «объекты-свойства». Оцениваются перспективы развития соответствующего программного обеспечения.

Ключевые слова: онтологический анализ данных; анализ формальных понятий; многозначная логика; ограничение существования свойств; веб-сервис.

Введение

Изложенный в [Смирнов, 2015а] опыт создания средств семантического моделирования и проектирования указал в целом хорошо известные задачи семантических технологий в инструментальных системах, и среди этих задач была отмечена ключевая роль *онтологического анализа* для моделирования смысла предметной области (ПрО), составляющей интерес разработчика. Компьютерная поддержка этого анализа осуществляется либо путем предоставления эксперту ПрО инструментов «приобретения» знаний [Гаврилова, 2008] (т.е. средств *автоформализации*), либо благодаря когнитивному *анализу данных* о ПрО [Загоруйко, 2013]. Поэтому аналитическим ядром развиваемого в течение ряда лет в Институте проблем управления сложными системами РАН инструментария семантического моделирования и проектирования стали методы и средства *онтологического анализа данных* (ОАД) [Семенова, 2015].

ОАД – методический комплекс для выявления *формальной онтологии* ПрО на основе эмпирических объектно-признаковых данных, структурированных согласно стандартизированному формату представления подобной информации,

известному как таблицы «объекты-свойства» (ТОС) [Барсегян, 2009].

В основу ОАД положен теоретически хорошо обоснованный метод *анализа формальных понятий* (АФП) [Ganter, 1999], в котором отражен классический подход к понятию как к фундаментальному элементу знания с объемом и содержанием. ОАД представляет собой прагматическую надстройку АФП, обладающую рядом новых способностей:

- поддерживается построение *произвольных отношений* между объектами (согласно концепции «отношение-проявление внутренних свойств объектов» [Смирнов, 2011]);
- допускается различного рода *неполнота* исходной эмпирической информации, и обрабатываются *противоречия* в извлекаемых из первичных данных *базовых семантических суждениях* (БСС) о ПрО вида b_{xy} = «объекту x присуще свойство y »;
- в анализ вовлекаются обычно имеющиеся у эксперта сведения о взаимосвязи измеряемых свойств объектов – так называемые «*ограничения существования*» свойств (ОСС) [Lammari, 2004].

Для реализации второй способности при оценке истинности БСС в отличие от *нечеткого АФП* [Yanget al., 2008] предложено использовать более

адекватную *многозначную векторную логику* [Смирнов, 2015b]. При этом и в том, и в другом варианте выявлена необходимость связывания модели неполноты исходных данных с моделью ОСС, и первоначально этот вопрос проработан применительно к *концептуальной сопряженности* свойств, возникающей в результате выполнения фундаментальной когнитивной процедуры АФП – *концептуального шкалирования* свойств [Офицеров, 2014].

В предлагаемой статье наряду с общим очерком операционной основы ОАД представлен метод и алгоритм работы с более общей моделью ОСС, а также проанализирована перспектива развития соответствующего программного обеспечения.

1. Математические модели и методы

В соответствии с общими положениями теории измерений ОАД основывается на том, что всякое измерение свойства объекта может дать специальный результат «None», свидетельствующий о нахождении значения измеряемого свойства за порогами чувствительности, вне динамического диапазона средства измерений. В АФП считается, что подобная фиксация семантического несоответствия исследуемого объекта и измерительной процедуры является, прежде всего, результатом фундаментального когнитивного приема – концептуального шкалирования [Ganter, 1999]. Т.е. исследователь сообразно субъективному видению априори «расщепляет» действительный диапазон процедуры измерения свойства, образуя набор новых свойств объектов ПрО, фактически измеряемых после этого в бинарной шкале наименований {X, None}, где X – лингвистическая константа, собирательно обозначающая любой символ шкалы динамического диапазона измерительной процедуры.

Как бы то ни было, «None-концепция» позволяет радикально изменить парадигму анализа данных и для начала естественным образом преобразовать ТОС в совокупность оценок истинности БСС_{xy}:

$$\|b_{xy}\| = \begin{cases} \text{Истина,} & \text{если результат измерения} \\ & \text{свойства у объекта } x \text{ есть } X; \\ \text{Ложь,} & \text{в противоположном случае.} \end{cases}$$

Именно на обработку таких данных ориентирован АФП, в котором используются следующие обозначения и модели:

- $K = (G^*, M, I)$ – *формальный контекст* (ФК), где $G^* = \{g_i\}_{i=1, \dots, r}$, $r = |G^*| \geq 1$ – набор объектов исследуемой ПрО, попавших в поле зрения исследователя (т.е. множество объектов обучающей выборки: $G^* \subseteq G$, где G – всё мыслимое множество объектов ПрО), $M = \{m_j\}_{j=1, \dots, s}$, $s = |M| \geq 1$ – множество измеряемых у объектов свойств, I – бинарное соответствие «объекты-свойства», т.е. совокупность оценок $\|b_{ij}\| \in \{\text{Истина, Ложь}\}$;

- операторы Галуа φ , ω (общая нотация «») для контекста K :

$\varphi(X) = X' = \{m_j \mid m_j \in M, \forall g_i \in X: (g_i, m_j) \in I\}$ – общие свойства объектов, составляющих $X \subseteq G^*$;

$\omega(Y) = Y' = \{g_i \mid g_i \in G^*, \forall m_j \in Y: (g_i, m_j) \in I\}$ – объекты, которые обладают всеми свойствами из $Y \subseteq M$;

- (X, Y) – формальное понятие, у которого $X \subseteq G^*$ – объем, $Y \subseteq M$ – содержание, причем $X = Y'$, $Y = X'$;

- $B(K)$ – множество формальных понятий контекста K ;

- $(B(K), \leq)$ – замкнутая решетка понятий, где $(X_1, Y_1) \leq (X_2, Y_2)$, если $X_1 \subseteq X_2$, или эквивалентно $Y_1 \supseteq Y_2$.

Естественную на практике многозначность оценок истинности БСС вызывает неполнота данных о ПрО (неточность, противоречивость, неопределенность и т.п.), которая вызывается реалиями накопления эмпирической информации: выполнением многократных независимых измерений свойства $m_j \in M$ у объекта $g_i \in G^*$, использованием для измерения одного и того же свойства m_j нескольких различных (но конгруэнтных) процедур, дифференциацией доверия к различным процедурам измерения. Поэтому в качестве адекватной модели исходных данных предлагается обобщенная ТОС, описываемая кортежем

$$(G^*, M, Se, Pr, A) \quad (1)$$

где:

- $Se = \bigcup_{i=1}^r Se_{(i)}$ – множество всех выполненных при зондировании ПрО серий измерений, $Se_{(i)} = \{se_{(i)k}\}_{k=1, \dots, q_{(i)}}$, $q_{(i)} \geq 1$, $i = 1, \dots, r$ – множество серий измерений, которым подвергнут объект $g_i \in G^*$;

- $Pr = \bigcup_{j=1}^s Pr_{(j)}$ – арсенал всех используемых при зондировании ПрО процедур измерения, $Pr_{(j)} = \{pr_{(j)k}\}_{k=1, \dots, p_{(j)}}$, $p_{(j)} \geq 1$, $j = 1, \dots, s$ – множество конгруэнтных процедур измерения свойства $m_j \in M$, причем всякая процедура $pr_{(j)k}$ характеризуется степенью доверия к ее результатам $t_{(j)k} \in (0, 1]$;

- $A = (a_{ij})_{i=1, \dots, m; j=1, \dots, n}$ – матрица результатов серий измерений Se свойств M у объектов из выборки G^* , выполненных с помощью процедур измерения Pr . Элементами этой матрицы могут быть константы X и None, а также еще две лингвистические константы. Константа **Failure** фиксирует отказ, сбой измерительного средства, воздержание при голосовании и т.п., т.е. тот нередко наблюдаемый на практике «результат» работы измерительной процедуры, который собирательно можно квалифицировать как «отказ от выполнения измерения». Константа **Notmeasured** (*notmeasured*) указывает, что в действительности в рассматриваемой серии измерений отдельная процедура измерения не использовалась (введение этого формального результата необходимо в том

числе и для сохранения двумерного характера обобщенной ТОС).

Задачей первичной обработки такой исходной эмпирической информации является извлечение оценок БСС $\|b_{ij}\|$. В [Смирнов, 2015b] на основе модели (1) и моделей векторной логики V^{TF} [Аршинский, 2007] дан метод построения формального контекста K с *нестрогим* соответствием I , которое образуют векторные оценки истинности $\|b_{ij}\| \in \langle b_{ij}^+, b_{ij}^- \rangle$, $b_{ij}^+, b_{ij}^- \in [0, 1]$, где компонент b_{ij}^+ - *Истина* - формируется свидетельствами, подтверждающими БСС, а компонент b_{ij}^- - *Ложь* - отрицающими БСС.

Эффективные методы вывода понятийной структуры ПрО из «мягких» ФК основаны на их предварительной α -аппроксимации при задании экспертом порога доверия к исходным данным. Затем к полученным бинарным контекстам применяются (с различными дополнениями) апробированные АФП-методы вывода понятий. Однако в [Офицеров, 2014] показано, что в общем случае этот подход оказывается некорректным, поскольку стандартное α -сечение «мягкого» соответствия «объекты-свойства» «слепо» к *зависимостям* между измеряемыми свойствами объектов ПрО.

Общие модели подобных зависимостей предложены в [Lamtagi, 2004]. Здесь достаточно учесть, что пара свойств $m_j, m_k \in M$, $j \neq k$ для любого объекта ПрО (и, следовательно, для $\forall g_i \in G^*$) может быть:

- *несовместимой*, если, обладая свойством m_j , объект g_i заведомо не обладает свойством m_k , и наоборот, т.е. $E(m_j, m_k) \leftrightarrow \forall g_i \in G^*: m_j \in \{g_i\}' \rightarrow m_k \notin \{g_i\}'$;
- *обусловленной*, если, обладая свойством m_j , объект g_i непременно обладает свойством m_k (хотя обратное может быть неверно), т.е. $C(m_j, m_k) \leftrightarrow \forall g_i \in G^*: m_j \in \{g_i\}' \rightarrow m_k \in \{g_i\}'$.

Так, в результате концептуального шкалирования в описании задачи возникают непересекающиеся подмножества попарно несовместимых свойств - *группы концептуально сопряженных свойств* (ГКСС).

Важно подчеркнуть, что рассматриваемые зависимости между свойствами известны субъекту, строящему формальное описание ПрО, *априори*, т.к. он является «владельцем» арсенала процедур измерения свойств и, обладая об этих инструментах в идеале полной информацией, может знать об определенных связях между результатами измерения одного и того же объекта.

Поэтому в качестве интегрированной модели входных данных для ОАД целесообразно использовать кортеж

$$(G^*, M, Se, Pr, A, M_{\Pi}, E_{\Pi}, C_{\Pi}, V), \quad (2)$$

где дополнением к (1) являются:

- M_{Π} - множество актуальных для исследователя «протосвойств» объектов ПрО, $1 \leq |M_{\Pi}| \leq |M|$, $M_{\Pi} = M_{\Pi 1} \cup M_{\Pi 2}$, $M_{\Pi 1} \cap M_{\Pi 2} = \emptyset$; $M_{\Pi 1}$ - подмножество *единичных* протосвойств (т.е. актуальных свойств не подвергнутых концептуальному шкалированию, формально «одиночных» ГКСС); $M_{\Pi 2}$ - подмножество «расщепленных» протосвойств (на два и более измеряемых свойства), или совокупность «множественных» ГКСС, в каждой из которых составляющие её свойства несовместимы;
- E_{Π} - множество пар несовместимых протосвойств, $E_{\Pi} \subseteq M_{\Pi} \times M_{\Pi}$;
- C_{Π} - множество пар обусловленных протосвойств, $C_{\Pi} \subseteq M_{\Pi} \times M_{\Pi}$;
- V - пары сопряженных свойств-валентностей $V \subseteq M \times M$, определяемых предикаторами актуальных бинарных отношений между объектами ПрО [Смирнов, 2011].

Идея метода построения корректной α -аппроксимации *нестрогого* ФК (НФК) предусматривает замену работы с общим порогом доверия α выбором *индивидуальных* порогов доверия к *определённым* фрагментам нестрогого соответствия I , характеризующим *каждый* объект $g_i \in G^*$. При этом допустимую α -аппроксимацию каждой такой локальной зоны нестрогого соответствия I предлагается осуществлять *без участия* человека исходя из необходимого и достаточного набора условий, начиная с требования минимального *ужесточения* порога доверия к рассматриваемому фрагменту данных и заканчивая применением различных мер *ранжирования* векторов истинности в V^{TF} -логике [Офицеров, 2014].

2. Алгоритмический базис ОАД

Основу алгоритмического обеспечения ОАД составляют результативные реализации упоминавшихся в разделе 1 методов, в том числе:

- формирование НФК на основе обобщенной ТОС;
- построение бинарного ФК путем *рационального* α -сечения НФК;
- вывод формальных понятий с выявлением отношения *обобщения* на их множестве (другие отношения в получаемой *решетке* формальных понятий оказываются установленными благодаря картине распределения валентностей в содержаниях понятий);
- трансформация решетки формальных понятий в формальную *онтологию* (с выполнением ряда прагматических требований к подобным описаниям ПрО).

Изучение и эксперименты с различными алгоритмами вывода формальных понятий позволили выбрать и адаптировать для ОАД алгоритм, описанный в [Yang et al., 2008]. Его достоинствами являются простота и наглядность при приемлемой производительности.

Другие из перечисленных алгоритмов оригинальны. Последний из указанных достаточно подробно описан в [Семенова, 2015], а наиболее важным и интересным представляется алгоритм построения рационального α -сечения НФК. Схема этого алгоритма приведена на рисунке 1.

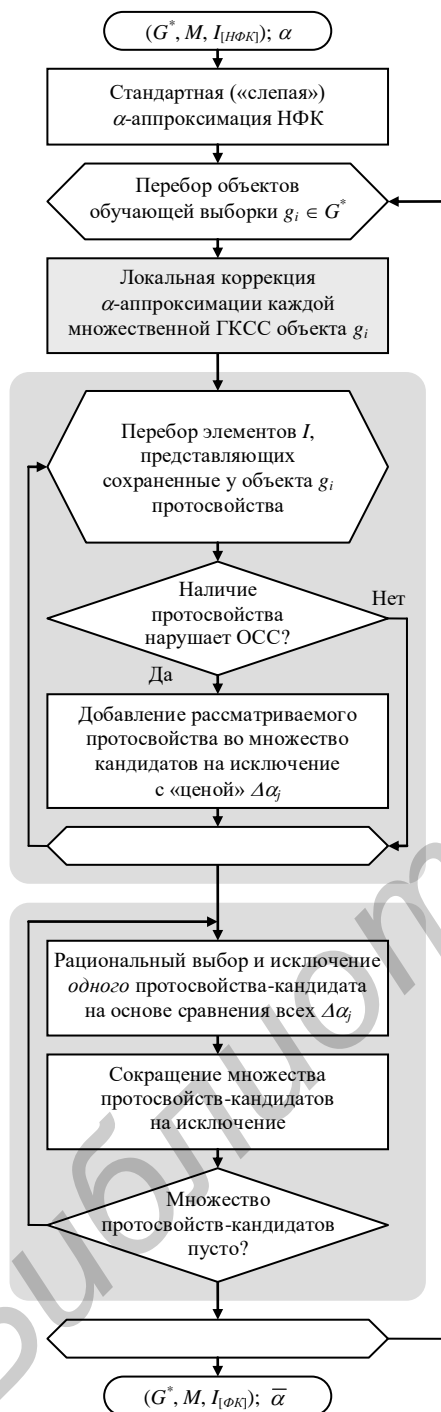
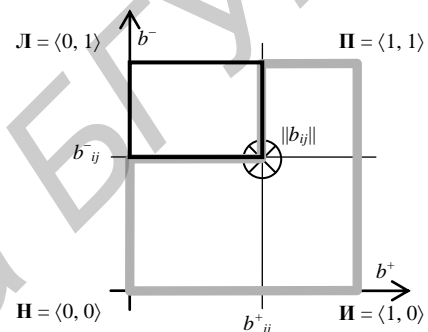


Рисунок 1 – Укрупнённая схема алгоритма построения рационального α -сечения нестроого формального контекста

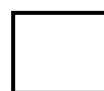
Рисунок 1 позволяет уяснить, что локальная коррекция стандартного α -сечения НФК производится на двух «уровнях»: сначала для каждой множественной ГКСС всех объектов $g_i \in G^+$ (см. выделенный заливкой блок), затем для множества протосвойств объектов обучающей выборки.

Такой подход целесообразен в силу двух обстоятельств. Во-первых, результат коррекции стандартной α -аппроксимация множественной ГКСС характеризуется существенной априорной определенностью: либо приходится признавать, что данный объект не обладает протосвойством, соответствующим рассматриваемой ГКСС, либо следует фиксировать « α -истинность» одного и только одного БСС, определяемого рассматриваемой ГКСС. А во-вторых, локальная коррекция α -аппроксимации множественной ГКСС не требует контроля соблюдения множества различных ОСС (оба вложенных цикла, выделенные на рисунке 1 фоном, включают эти акты) т.к. в такой группе эти ограничения «однородны» - всегда лишь попарно несовместимы.

Рисунки 2-4 иллюстрируют некоторые из задач, решаемых рассматриваемым алгоритмом на уровне множественных ГКСС объектов обучающей выборки.



Н, И, Л, П - истинностные константы V^{FF} -логики «Неопределенность», «Истина», «Ложь», «Противоречие».



- область, при выборе в которой порога доверия $\alpha = \langle \alpha^+, \alpha^- \rangle$, БСС с векторной оценкой истинности $\langle b^+_{ij}, b^-_{ij} \rangle$ признается истинным;



- область, при выборе в которой порога доверия $\alpha = \langle \alpha^+, \alpha^- \rangle$, БСС с векторной оценкой истинности $\langle b^+_{ij}, b^-_{ij} \rangle$ признается ложным.

Рисунок 2 – Области выбора порога доверия к исходным данным, определяющие приближенную оценку истинности в шкале {Истина, Ложь} отдельно взятого базового семантического суждения о предметной области b_{ij}

Для каждой множественной ГКСС всякого объекта обучающей выборки алгоритм выполняет следующую последовательность действий:

- выявление *граничных элементов* множества оценок истинности *локальных БСС* (т.е. БСС соответствующих рассматриваемым объекту обучающей выборки и множественной ГКСС)- *множества недоминируемых альтернатив* выбора наиболее правдоподобных оценок (рисунок 3);
- установление для каждого граничного элемента *пределов* области выбора порога доверия, в которой истинно БСС, отвечающее этому элементу, а другие локальные БСС ложны, и вычисление значения меры *ужесточения* порога доверия, необходимого для такого выбора (рисунок 4);

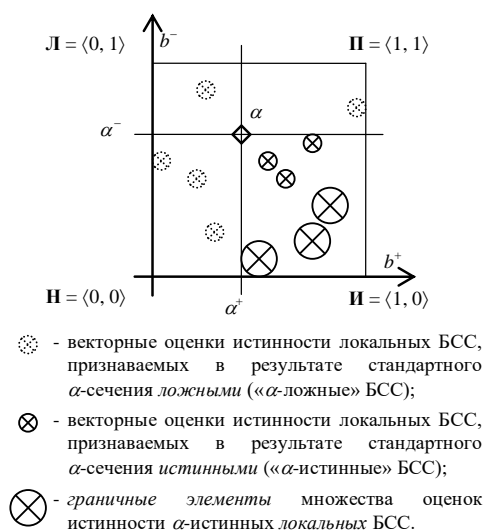


Рисунок 3 – Разделение оценок истинности локальных базовых семантических суждений

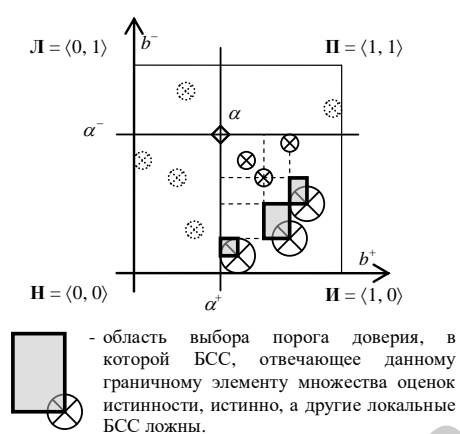


Рисунок 4 – Области корректного выбора порога доверия к данным, характеризующим множественную группу концептуально сопряженных свойств объекта обучающей выборки

- исключение из множества граничных элементов всех «дублированных» элементов (т.е. подмножеств эквивалентных оценок истинности), а также проверка существования *неустрашимого противоречия* локальных БСС – наличия среди них двух и более БСС, истинность которых оценивается вектором $I = \langle 1, 0 \rangle$;

- если во множестве граничных элементов сохранилось два или более элементов (т.е. остаётся неопределённость выбора одного «лучшего» локального БСС), то производится *гарантированное* сокращение этого множества до *одноэлементного*. Этот результат достигается благодаря использованию трех скалярных мер, значения которых вычисляются по *текущему составу* множества граничных элементов. Сначала производится сохранение элемента(ов) с *наименьшим* значением меры *жесточения порога доверия*. Если результат не достигнут, то выполняется сохранение элемента(ов) с *наибольшим* значением меры *достоверности* векторной оценки истинности [Аршинский, 2007]. Наконец (если выбор все ещё необходим), среди оставшихся граничных элементов указывается тот, который

характеризуется *наибольшим* значением меры *определённости* векторной оценки истинности [Аршинский, 2007].

3. Программное обеспечение и перспективы его развития

ОАД как аналитическое ядро *gB*-системы семантического моделирования и проектирования развивается на платформе MS Excel и языка программирования VBA [Смирнов, 2015a]. Однако очевидно, что оформленный в виде настольного приложения программный продукт удобней предоставлять конечным пользователям в качестве веб-сервиса. Для этого существует множество способов, и если приложение представляет собой отдельную самодостаточную программу, публикация его в интернете обычно не вызывает затруднений. Однако развертывание книги Excel, содержащей макросы, сопряжено с рядом проблем.

Стандартный экспорт в html позволяет сохранить определенную интерактивность приложения Excel, например сортировку данных или фильтры, однако не поддерживает макросы. Существуют специализированные сервисы, позволяющие публиковать книгу Excel как веб-страницу, например, Spreadsheet WEB или Excel Online, однако в настоящее время и они не осуществляют поддержку макросов^{1,2}.

Едва ли не единственным исключением здесь является продукт, разработанный компанией EASA Software³. Он позволяет экспортировать книгу Excel с макросами с минимальной затратой усилий – почти весь процесс преобразования происходит автоматически. Однако поддерживаются не все возможности VBA-макросов, например, работа контекстных меню. В целом данный программный продукт не пользуется особой популярностью и потому, что у него отсутствует достаточная документация.

Рассматривая перспективы развития программного обеспечения ОАД, следует отметить, что разработка *gB*-системы и все её составляющих, включая инструментарий ОАД, выполнена при последовательном соблюдении принципов объектно-ориентированного программирования и использовании соответствующих возможностей VBA. Поэтому для создания интерактивного веб-сервиса можно, переработать часть кода, использовать VBA в SharePoint⁴. Наконец, можно переписать весь функционал на Java с использованием, например, технологий JavaBeans, JSF и любого сервера приложений⁵.

¹ <https://blogs.office.com/2014/04/14/weve-updated-excel-online-whats-new-in-april-2014/>

² <http://wiki.pagos.com/display/SSWEB/Current+Restrictions>

³ <http://www.easasoftware.com/media-resources/easa-videos/>

⁴ [https://msdn.microsoft.com/en-us/library/office/aa159897\(v=office.11\).aspx](https://msdn.microsoft.com/en-us/library/office/aa159897(v=office.11).aspx)

⁵ <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

Заключение

Наличие в арсенале систем семантического моделирования и проектирования средств онтологического анализа данных не просто расширяет их функциональный потенциал. Основу таких средств по необходимости составляют фундаментальные представления о когнитивных возможностях человека и соответствующие математические модели и методы работы со знаниями, которые могут быть использованы технологической системой моделирования и проектирования в целом (использование единой платформы представления знаний для анализа «внешней» действительности и «внутреннего» устройства системы, по меньшей мере, целесообразно).

Конкурентоспособность разрабатываемых авторами инструментальных средств семантического моделирования и проектирования во многом обеспечивает именно представленный в статье методический и алгоритмический комплекс интеллектуального онтологического анализа данных, а его программное обеспечение имеет достаточные резервы развития.

Работа выполнена при проведении исследований по теме «Модели и методы формирования согласованной системы понятий о предметной области управления в процессах коллективного принятия решений» в рамках государственного задания Института проблем управления сложными системами РАН на 2016 год, а также при государственной поддержке Министерства образования и науки РФ в рамках реализации мероприятий Программы повышения конкурентоспособности Самарского государственного аэрокосмического университета среди ведущих мировых научно-образовательных центров на 2013-2020 годы.

Библиографический список

- [Аршинский, 2007] Аршинский, Л.В. Векторные логики: основания, концепции, модели / Л.В. Аршинский. - Иркутск: Иркут. гос. ун-т, 2007. - 228 с.
- [Барсегян и др.]. Анализ данных и процессов / А.А. Барсегян и др. - СПб.: БХВ-Петербург, 2009. - 512 с.
- [Гаврилова, 2008] Гаврилова, Т.А. Интеллектуальные технологии в менеджменте: инструменты и системы / Т.А. Гаврилова, Д.И. Муромцев. - СПб.: Изд-во «Высшая школа менеджмента»; Изд. дом СПбГУ, 2008. - 488 с.
- [Загоруйко, 2013] Загоруйко, Н.Г. Когнитивный анализ данных / Н.Г. Загоруйко. - Новосибирск: Академическое изд-во «Гео», 2013. - 186 с.
- [Офицеров, 2014] Офицеров, В.П. Метод альфа-сечения нестрогих формальных контекстов в анализе формальных понятий / В.П. Офицеров, В.С. Смирнов, С.В. Смирнов // Проблемы управления и моделирования в сложных системах: Труды XVI международной конф. (30 июня - 03 июля 2014 г., Самара, Россия). - Самара: СамНЦ РАН, 2014. - С. 228-244.
- [Семенова, 2015] Семенова, В.А. OntoWorker: программная лаборатория для онтологического анализа данных / В.А. Семенова, В.С. Смирнов, С.В. Смирнов // Проблемы управления и моделирования в сложных системах: Труды XVII международной конф. (22-25 июня 2015 г., Самара, Россия). - Самара: СамНЦ РАН, 2015. - С. 382-393.
- [Смирнов, 2011] Смирнов, С.В. Построение онтологий предметных областей со структурными отношениями на основе

анализа формальных понятий / С.В. Смирнов // Знания – Онтология – Теория: Материалы Всероссийской конф. с международным участием (3-5 октября 2011 г., Новосибирск, Россия). Т. 2. – Новосибирск: Институт математики СО РАН, 2011. - С. 103-112.

[Смирнов, 2015a] Смирнов, С.В. Опыт создания средств семантического моделирования и проектирования на массовой программной платформе / С.В. Смирнов // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2015): Материалы V международ. науч.-тех. конф. (19-21 февраля 2015 г. Минск, Беларусь) / Редкол.: В.В. Голенков (отв. ред.) и др. – Минск: БГУИР, 2015. – С. 413-416.

[Смирнов, 2015b] Смирнов, С.В. Нестрогий анализ формальных понятий / С.В. Смирнов // Знания – Онтология – Теория: Материалы Всероссийской конф. с международным участием (6-8 октября 2015 г., Новосибирск, Россия). Т. 2. – Новосибирск: Институт математики СО РАН, 2015. - С. 142-150.

[Ganter, 1999] Ganter, B. Formal Concept Analysis. Mathematical foundations / B. Ganter, R. Wille. - Berlin-Heidelberg: Springer-Verlag, 1999. - 290 p.

[Lammari, 2004] Lammari, N. Building and maintaining ontologies: a set of algorithms / N. Lammari, E. Metais // Data & Knowledge Engineering. - 2004. - Vol. 48(2). - P. 155-176.

[Yang et al., 2008] Yang, K. M. Fuzzy Concept Mining based on Formal Concept Analysis / K.M. Yanget al. // Int. J. of Computers. - 2008. - Issue 3. Vol. 2. - P. 279-290.

ALGORITHMS AND SOFTWARE FOR ONTOLOGICAL DATA ANALYSIS

Zubtcov R.O., Semyonova V.A.,
Smirnov S.V.

Samara State Aerospace University,
Samara, Russia

zubtcov.r@gmail.com

queenbfjr@gmail.com

Institute for the Control of Complex Systems of the
Russian Academy of Sciences, Samara, Russia

smirnov@iccs.ru

The paper describes the analytical kernel of developing semantic modeling and designing tools. This kernel is composed of mathematical models, methods and algorithms of ontological data analysis, which are designed to bring the formal ontologies based on empirical data provided in the form of matching “objects-properties”. Ontological Data Analysis is based on theoretically well-established Formal Concept Analysis method and has a number of new abilities. To be able to handle incomplete and inconsistent data was suggested to use more adequate multi-valued vector logic. Dependencies called “existence constraints” are taken into account when constructing a correct context for the Formal Concept Analysis.

In addition, perspectives of extension and development of software are estimated. Possibility and ways to create a web service are considered.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ФОРМИРОВАНИЕ МЕДИЦИНСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ

Кобринский Б.А.

*Институт системного анализа Федерального исследовательского центра
"Информатика и управление" РАН,
Российский национальный исследовательский медицинский университет им. Н.И. Пирогова
Минздрава России,
г. Москва, Российская Федерация
bak@isa.ru*

В работе рассматриваются вопросы построения и интеграции информационных медицинских систем на основе модульного принципа. Реализация «надсистемных» модулей рассматривается как основа для любых медицинских систем, «внутрисистемные» модули являются проблемно ориентированными. Модульный принцип информационного обеспечения систем является основой для построения как систем общего назначения, так и специализированных медицинских систем (регистров). Система управляющих знаний представлена как метабаза знаний, включающая суббазы по отдельным направлениям.

Ключевые слова: управляемые знаниями информационные системы; модульное построение информационных систем; метабаза знаний; суббазы знаний.

Введение

Гибридные системы, объединяющие ведение информационной системы (ИС) типа электронной медицинской карты (ЭМК) с поддержкой врачебных решений относят к клиническим информационным системам 4-го поколения [Dean, 2000]. В таких ИС-ЭМК на основании определенных данных может активизироваться индивидуальная схема обследования и лечения с выдачей, в том числе, предупреждений о потенциальном риске принимаемых решений для пациентов [Рот, Шульман, 2004; Будкевич и др., 2008]. Таким образом, под гибридными будем понимать системы, в которые инкорпорированы модули принятия решений, срабатывающие автоматически при вводе определенной информации или включающиеся по запросу пользователя для оценки состояния или выбора лечебной тактики, используя накопленные в ЭМК данные о состоянии больного [Кобринский, 2014]. В контексте систем, управляемых знаниями, данное направление может значительно повысить эффективность работы врача с ЭМК.

В связи с тем, что ИС-ЭМК являются базовыми информационными структурами, содержащими разнообразные, накапливающиеся в течение жизни, данные о здоровье людей, то именно на их

основе должны формироваться любые другие персонифицированные базы медицинских данных. Широкое распространение в настоящее время получил мониторинг состояния различных групп пациентов для чего создаются проблемно ориентированные информационные системы, часто называемых в медицинских приложениях регистрами (ИС-регистры).

1. Метабаза знаний интегрированной информационной системы, управляемой знаниями

Попытаемся в новом ракурсе, на основе Open Semantic Technology for Intelligent Systems, рассмотреть проблему отношений между первичными электронными медицинскими картами пациентов, ведущимися в медицинских организациях разного типа и разных уровней системы здравоохранения, что обеспечивает формирование территориальных федеральных интегрированных ЭМК (ИЭМК) и с использованием сведений из которых должны строиться и регулярно обновляться ИС-регистры. В этом случае, с одной стороны, должно обеспечиваться полноценное поддержание различных аспектов медико-технологического процесса ведения ЭМК, с другой стороны, автоматическое формирование специализированных регистров. Переход к

системам, управляемым знаниями позволит обеспечить выбор необходимых данных из интегрированных ЭМК или из ряда первичных ЭМК для осуществления многообразного контроля врачебных назначений.

Компьютерная система, управляемая знаниями, – система, в основе которой лежит представленная унифицированным образом база знаний, содержащая в систематизированном виде всю информацию, используемую этой системой [Голенков и др., 2015]. В рассматриваемой ситуации группы взаимодополняющих информационных систем построение базы знаний целесообразно осуществлять в виде комплекса взаимосвязанных на семантическом и программном уровнях суббаз знаний по различным направлениям (аспектам) модели Мира предметной медицинской области. Такой комплекс суббаз знаний, связанных определенными отношениями, будем называть метабазой знаний интегрированной системы, управляемой знаниями.

Семантический анализ информации имеет большое значение в медицине, как слабо структурированной области данных, где практически невозможно перейти при построении информационных систем к полной формализации данных и отказаться от текстов на естественном языке.

Разработка интеллектуальной системы сводится к разработке ее логико-семантической модели, то есть формальному описанию структуры интеллектуальной системы и всех ее компонентов. Исходя из того, что формальные описания операций, работающих над семантической памятью, также хранятся в базе знаний и являются ее частью, то, по сути, проектирование семантической модели интеллектуальной системы сводится к проектированию базы знаний этой системы [Давыденко, 2012].

Регистры, формируемые традиционно на основе специальных входных документов (учетных форм), в новой постановке предполагается формировать на основе соответствующих выборок данных из первичных ЭМК, в отдельных случаях за счет слияния содержащейся в них информации по приоритету на основе алгоритмов, построенных с использованием экспертных знаний (сохраняя в ИЭМК данные, которые можно считать более надежными), путем управления из соответствующих суббаз знаний, что ускорит процесс управления. При этом формирование любых регистров можно будет осуществлять без привычного в настоящее время дублирования входных данных. А поиск и сопоставление определенной информации в разных ЭМК для формирования интегрированных ЭМК и территориальных и федеральных регистров может быть возможен с учетом синонимии медицинских терминов.

2. Модульность и семантическая интероперабельность

В системах, управляемых знаниями, обмен информацией между решателями возможен только через общую для них память на основе унифицированного представления обрабатываемой информации. Более того, чтобы достаточно быстро можно было найти информацию, востребованную в ходе решения задачи, вся хранимая в системе информация должна быть структурирована на основе модульного принципа, что в последующем обеспечивает компонентное построение систем [Голенков и др., 2015]. Такой подход может обеспечить в медицинских системах, управляемых знаниями, поиск и передачу необходимых данных в виде отдельных модулей или семантических блоков: демографического, диагностического, факторов риска и др. Но это возможно при условии семантической интероперабельности с использованием общепринятых классификаторов, включающих стандарты для кодирования и обмена основными клиническими характеристиками состояния здоровья, примером чего может служить Systematized Nomenclature of Medicine – Clinical Terms [Price et al., 2000]. Таким образом под модулем будем понимать стандартизованную информационную структуру, включающую фиксированный перечень параметров, которая может содержать также текстовые файлы и /или присоединенные медицинские изображения, характеризующиеся смысловым содержанием. Базовые модули (анкетный и др.) должны быть едиными, далее специализация предполагается по направлениям медицины, заболеваниям, социальным и профессиональным аспектам. Из этого проистекает интеграция «надсистемная», то есть использование общих для всех информационных медицинских систем модулей (анкетный, витальных данных, профилактических прививок, перенесенных заболеваний, инвалидности и т.д.), в сочетании с «внутрисистемными» модулями для отображения разнообразной информации, специфичной для различных заболеваний и групп населения (беременные, неонатальный период жизни, профессиональная патология, течение отдельных болезней, лечение и др.) [Кобринский, 2011]. В отношении внутрисистемных специализированных модулей, обеспечивающих дифференциацию информации электронной медицинской карты необходимо отметить важность их однотипной структуризации, что позволит осуществлять быстрый обмен необходимыми модулями или блоками (фрагментами) данных по запросам к базе знаний управляющей системы. При однообразно реализованном модульном принципе построения ЭМК повысится комфортность программного продукта для медицинских работников. Одновременно это позволит оперативно предоставлять бригадам скорой медицинской помощи и врачам «Медицины катастроф» в чрезвычайных ситуациях необходимые сведения в

виде отдельных модулей. Структуризация и модульный принцип построения ЭМК являются необходимыми условиями перехода к персона-центрированной парадигме интеграции медицинских данных пациентов [Lloyd-Williams, 2004; Кобринский, 2006], в отличие от существующей «привязки» данных пациентов к картам различных типов учреждений.

Модульная реализация ИС-ЭМК позволит преодолеть «конфликт» между этими системами и специализированными регистрами по видам патологии и социальным группам. Дилемма современной ситуации заключается в независимой реализации территориальных / федеральных ИС-регистров и ИС-ЭМК. Один и тот же пациент может быть включен в несколько федеральных регистров (например, при наличии врожденной патологии у ребенка, от которого отказались родители, информация о нем должна быть занесена в регистр детей-сирот, в регистр врожденных пороков развития и в регистр по детской инвалидности, а также в регистр мониторинга диспансеризации детей). В этом случае нарушается основной постулат информатики об однократном вводе данных, что увеличивает нагрузку на медицинский персонал и противодействует снижению трудовых и финансовых затрат. В технологии модульной архитектуры электронных медицинских карт резко упрощается и ускоряется формирование специализированных регистров и многократно используются введенные в разных местах данные.

3. Управляемые знаниями информационные и справочные системы

При ИС-ЭМК, управляемой знаниями, станет возможным контроль и предоставление врачу сведений об отклонении от нормативных значений лабораторных показателей, сведения о несовместимости лекарственных препаратов, их побочных действиях и другая информация справочного характера. Поиск определенной информации в ЭМК при поступлении персональных данных мониторинга физиологических показателей, отклоняющихся от нормы для данной возрастной группы, обеспечиваемый управлением знаниями, позволит предварительно оценить уровень отклонения при обращении к справочной системе, а затем уточнить персональную ситуацию при анализе информации, накапливаемой в персона-центрированной ЭМК и обеспечить решение на основе логического вывода.

К области ситуаций, требующих получения справочной информации, несомненно относятся предоставление врачу-пользователю возможности навигации по семантическому пространству предметной области и интерпретация любых вопросов, поиск необходимой информации и представление ее в удобной для пользователя форме [Давыденко. 2012]. Также можно говорить об информационной поддержке обследования пациентов в зависимости от выдвигаемых врачом диагностических гипотез. Одновременно управляемая знаниями информационная система может контролировать стоимость лечения в целом, расходы на медикаменты, стоимость проведенных исследований и сравнивать их с медико-экономическими стандартами, с объемом бюджетного финансирования, со страховыми взносами и т.д.

В целом интегрированная информационная медицинская система, управляемая знаниями (рис.1), на любом уровне будет представлять собой мультифункциональную систему, обеспечивающую выполнение различных задач при минимизации затрат на ввод данных о состоянии здоровья пациентов и одновременном получении врачом максимально полной информации независимо от места и времени ее ввода в ИС-ЭМК.

Заключение

В настоящее время в медицине все большее значение приобретают задачи интеграции персональных данных пациентов как для полномасштабного представления состояния организма, так и для мониторинга определенных данных на основе проблемно-электронных медицинских карт и формирования на их основе проблемно ориентированных информационных систем (регистров). Это определяет интерес к информационным системам, управляемым знаниями, которые характеризуются модульным построением, для чего применяются библиотеки многократно используемых компонентов. В то же время база знаний, которую в этом случае целесообразно рассматривать как метабазу, следует строить как комплекс суббаз.

Модульное построение интегрированных электронных медицинских карт и регистров требует создания, постоянного пополнения и сопровождения специальных библиотек повторно используемых модулей, которые должны обеспечивать формирование и модификацию информационных систем.

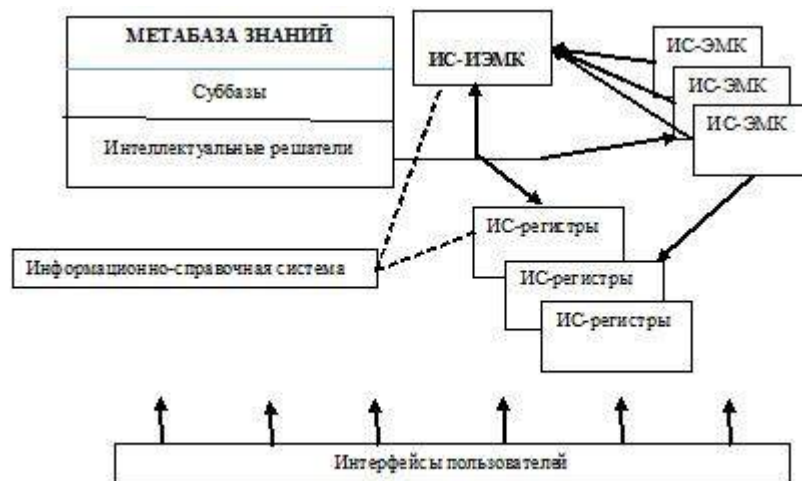


Рисунок 1 –Интегрированная информационная медицинская система, управляемая знаниями

Библиографический список

[Будкевич и др., 2008] Электронная история болезни с поддержкой врачебных решений при ожоговой травме у детей /Л.И.Будкевич, Б.А.Кобринский, М.А.Подольная, В.М.Розинов, О.И. Старостин // Вестник новых медицинских технологий. – 2008. – №2. – С.232-233.

[Голенков и др., 2015] Семантическая технология компонентного проектирования систем, управляемых знаниями / В.В.Голенков, Н.А. Гулякина // Материалы Умеждународной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (Минск, 19-21 февр. 2015 г.). – Минск:БГУИР, 2015.– С.57-78.

[Давыденко, 2012]Комплекснаяметодикапроектированиясемантическихмоделейинтеллектуальныхсправочныхсистем / И.Т.Давыденко // Материалы II международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (Минск, 16-18 февр. 2012 г.). – Минск:БГУИР, 2012.– С. 457-466.

[Кобринский, 2006] Информационные медицинские системы: конвергенция и интеграция на основе персонализированной парадигмы / Б.А.Кобринский // Международный форум «Информационные технологии и общество 2006 (18-25 сент. 2006, Каорли (Венеция), Италия): Матер. форума. – М.: ООО «Форсикон», 2006. – С.68-74.

[Кобринский, 2011] Единая концепция построения персональных электронных медицинских карт, информационных систем разных уровней и специализированных регистров / Б.А.Кобринский // Врач и информационные технологии. – 2011. – №3. – С.15-21.

[Рот и др., 2004] Эволюция клинических информационных систем / Г.З.Рот, Е.Н. Шульман // Здоровоохранение. – 2004. - №1. – С.169-174.

[Dean, 2000] Computerizing public health surveillance systems / A.G. Dean // Principles and Practice of Public Health Surveillance / S.M. Teutsch, S.M. Churchill (eds.). Second ed. – Oxford Univ. Press, Inc., 2000. – P.229-252.

[Lloyd-Williams, 2004]Ehealth: Adilemma for Europe / D.Lloyd-Williams // British Journal Healthcare Computing & Information Management. – 2004. – Vol.21, No.10. – P.20-23.

[Price et al., 2000]SNOMED Clinical Terms / C. Price, K. Spackman // British Journal Healthcare Computing & Information Management. – 2000. –Vol.17, No.3. – P.27-31.

FORMING OF MEDICAL INFORMATION SYSTEMS, MANAGED BY KNOWLEDGES

Kobrinskii B.A.

Institute for Systems Analysis, Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Russian National Pirogov Research Medical University, Moscow, Russia

bak@isa.ru

The paper considers the issues of building and integrating health information systems based on a modular principle. Implementation of the "super-system" modules regarded as the basis for any medical systems, "intra" modules are problem-oriented. The modular concept of information support systems is the basis for the construction as a general-purpose and specialized medical systems (registers). The system of control of knowledge is presented as a metabase knowledge, including subbases in separate directions.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СРЕДСТВА СТРУКТУРИЗАЦИИ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ БАЗ ЗНАНИЙ

Давыденко И.Т.* , Гракова Н.В.* , Сергиенко Е.С.** , Федотова А.В.**

* *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ir.davydenko@gmail.com
natalia.grakova@gmail.com

** *Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Россия*

deav@inbox.ru

*** *Московский государственный технический университет им. Н.Э.Баумана,
г. Москва, Россия*

afedotova.bmstu@gmail.com

В работе рассмотрены средства структуризации семантических моделей баз знаний, основанные на унифицированных семантических сетях. Основное внимание уделяется рассмотрению понятий структура, знание, рассматривается их классификация. Отдельный раздел посвящен описанию таких видов знаний, как предметная область и онтологии.

Ключевые слова: база знаний, структуризация базы знаний, предметная область

Введение

При накоплении больших объемов знаний появляется огромное количество самых различных *понятий* (концептов) и возникает проблема структуризации и систематизации этого многообразия понятий.

Одним из базовых принципов *Технологии OSTIS* (Open Semantic Technology for Intelligent Systems) является принцип унификации структуризации баз знаний [PMS, 2016]. В рамках данного принципа предполагается трактовать семантическую структуру базы знаний компьютерной системы как отражение иерархической системы взаимосвязанных друг с другом *предметных областей*, представляемых в базе знаний. Это предполагает:

- (1) уточнение понятия предметной области;
- (2) разработку языковых средств описания структуры предметных областей с помощью унифицированных семантических сетей;
- (3) разработку языковых средств описания типологии предметных областей и различных видов связей между ними.

Структуризация базы знаний, выделение в ней различных связанных между собой подструктур необходимы по целому ряду причин [Гаврилова и др., 2001]. В частности, это необходимо для дидактических целей (человеку усваивающему некоторые знания, желательно иметь, своего рода оглавление этих знаний, что позволяет планировать их усвоение и рассматривать их с различной степенью детализации), а также для организации распределения работ по проектированию баз знаний (когда разным исполнителям поручается разработка разных фрагментов базы знаний, имеющих достаточно четкие границы) [Голенков, 2012].

Таким образом, в данной работе вводится и уточняется одно из базовых с точки зрения представления знаний в *Технологии OSTIS*, понятие *структуры*, на основе которого далее уточняется понятие *знания*, одним из видов которого является понятие *предметной области*. Для спецификации предметных областей вводится понятие *онтологии* и рассматривается их типология [Tarassov, 2015], [Кудрявцев, 2010], [Федотова, 2015].

В качестве формальной основы для представления знаний в рамках *Технологии OSTIS* используются графодинамические модели

специального вида – семантические модели представления и обработки знаний, в основе которых лежат унифицированные семантические сети с базовой теоретико-множественной интерпретацией. Основным способом кодирования информации для таких сетей является SC-код (Semantic Code) [Голенков, 2012]. Элементы таких сетей, представленных в SC-коде будем называть *sc-элементами*, в свою очередь, узлы таких сетей будем называть *sc-узлами*, связи между ними – *sc-коннекторами*. [IMS, 2016]

Некоторые из рассматриваемых в данной работе понятий будут сопровождаться фрагментами их спецификации на формальном языке SCn (Semantic Code natural). [IMS, 2016]

1. Предметная область структур

В данном разделе вводится понятие *структуры*, рассматривается классификация структур, типология ролей элементов в рамках структур.

1.1. Понятие структуры и типология структур

Одним из наиболее общих (с точки зрения уточнения семантики) понятий при описании свойств какого-либо объекта является понятие *структуры*.

Каждая *структура* представляет собой множество *sc-элементов*, удаление одного из которых может привести к нарушению целостности этого множества.

Пример изображения структуры на языке SCg [IMS, 2016] приведен на рисунке 1.

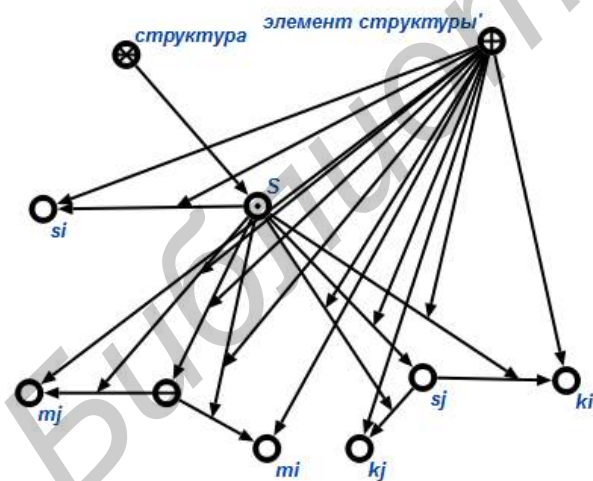


Рисунок 1 - Пример структуры на языке SCg

Выделяются следующие классы структур:

структура

= *sc-структура*

= *структура, представленная в виде текста SC-кода*

<= *разбиение**:

{

• *связная структура*

• *несвязная структура*

}

<= *разбиение**:

{

• *тривиальная структура*

• *нетривиальная структура*

}

Структуре, представленной в SC-коде, поставим в соответствие орграф, вершинами которого являются *sc-элементы*, а дугами – пары инцидентности, связывающие *sc-коннекторы* с инцидентными им *sc-элементами*, которые являются компонентами указанных *sc-коннекторов*.

Если полученный таким способом орграф является связным орграфом, то исходную структуру будем считать *связной структурой*.

Структуре, представленной в SC-коде, поставим в соответствие орграф, вершинами которого являются *sc-элементы*, а дугами – пары инцидентности, связывающие *sc-коннекторы* с инцидентными им *sc-элементами*, которые являются компонентами указанных *sc-коннекторов*. Если полученный таким способом орграф не является связным орграфом, то исходную структуру будем считать *несвязной структурой*.

Под *тривиальной структурой* понимается *структура*, не содержащая в качестве элементов *связок*.

В свою очередь, под *нетривиальной структурой* понимается *структура*, среди элементов которой есть хотя бы одна *связка*.

1.2. Типология элементов структур

В рамках заданной структуры ее элементы можно классифицировать по следующим признакам:

• насколько полно в рамках заданной структуры представлено множество, обозначаемое заданным sc-элементом вместе с соответствующими дугами принадлежности;

• существуют ли в рамках заданной структуры *sc-элементы*, обозначающие множества, являющиеся надмножествами того множества, которое обозначается заданным sc-элементом;

• уровень («этаж») иерархии перехода от знаков к метазнакам для заданного sc-элемента в рамках заданной *структуры*.

Для формального представления структур были введены понятия, описывающие роли элементов в рамках структуры. Рассмотрим их подробнее.

элемент структуры' – *ролевое отношение*, указывающее на все элементы каждой структуры.

элемент структуры'

<= *разбиение**:

{

• *непредставленное множество'*

- *полностью представленное множество'*
 - *частично представленное множество'*
 - *элемент структуры, не являющийся множеством'*
- }
 <= разбиение*:
 {
 • *максимальное множество'*
 • *немаксимальное множество'*
 }

непредставленное множество' – *ролевое отношение*, связывающее структуру со знаком множества, все элементы которого не являются элементами данной структуры.

непредставленное множество'
 = *множество, не представленное в рамках данной структуры'*
 = *быть знаком множества, элементы которого не являются элементами данной структуры'*
 ∈ *ролевое отношение*

полностью представленное множество' – *ролевое отношение*, связывающее структуру со знаком множества (любого семантического типа – класса, связки или структуры), все элементы которого являются элементами данной структуры.

полностью представленное множество'
 = *множество, полностью представленное в рамках данной структуры'*
 = *множество, все элементы которого являются элементами данной структуры'*
 = *полностью представленный класс'*
 ∈ *ролевое отношение*

частично представленное множество' – *ролевое отношение*, связывающее структуру со знаком множества, не все элементы которого являются элементами данной структуры.

частично представленное множество'
 = *множество, частично представленное в рамках данной структуры'*
 = *множество, некоторые элементы которого являются элементами данной структуры'*
 = *быть знаком множества, некоторые элементы которого являются элементами данной структуры'*
 ∈ *ролевое отношение*

элемент структуры, не являющийся множеством' - *ролевое отношение*, связывающее структуру с знаком некоторой сущности, не являющейся множеством.

максимальное множество' – *ролевое отношение*, связывающее структуру со знаком множества, для которого не существует множества, которое было бы надмножеством указанного множества и знак которого был бы элементом этой же структуры.

немаксимальное множество' – *ролевое отношение*, связывающее структуру со знаком

множества, для которого в рамках данной структуры существует множество, являющееся надмножеством указанного множества.

первичный элемент' – *ролевое отношение*, указывающее на элемент структуры, являющийся либо терминальным элементом, либо знаком множества, такого что не существует другого элемента этой же структуры, который был бы элементом множества, обозначаемого первым из указанных элементов структуры. При этом соответствующая пара принадлежности может существовать, но в состав данной структуры не входит.

первичный элемент'
 = *первичный элемент данной структуры'*
 = *sc-элемент первого уровня в рамках данной структуры'*
 ∈ *ролевое отношение*
 ∈ *семантический уровень структурного элемента*
 ⊂ *элемент структуры'*

вторичный элемент' – *ролевое отношение*, указывающее на элемент структуры, обозначающий множество, все или некоторые элементы которого являются элементами указанной структуры.

вторичный элемент'
 = *вторичный элемент данной структуры'*
 = *элемент данной структуры, имеющий семантический уровень более 2'*
 = *непервичный элемент'*
 ∈ *ролевое отношение*
 ⊂ *элемент структуры'*
 ⊃ *элемент второго уровня'*

элементом второго уровня' в рамках заданной структуры может быть связка первичных элементов, тривиальная структура из первичных элементов или класс первичных элементов.

элемент второго уровня'
 ∈ *ролевое отношение*
 ∈ *семантический уровень структурного элемента*

В зависимости от того, элементы каких уровней представлены в рамках заданной структуры выделяются дополнительные классы структур, в частности, **структура второго уровня** – структура, среди элементов которой есть хотя бы один элемент второго уровня'.

Для того, чтобы описать, элементы каких уровней представлены в рамках структуры, вводится понятие **семантический уровень структурного элемента**, который представляет собой параметр, каждый элемент которого является классом *sc-дуг принадлежности*, связывающих некоторую структуру с теми ее элементами, которые имеют одинаковый семантический уровень в рамках данной структуры. Значением данного параметра является число, обозначающее указанный семантический уровень.

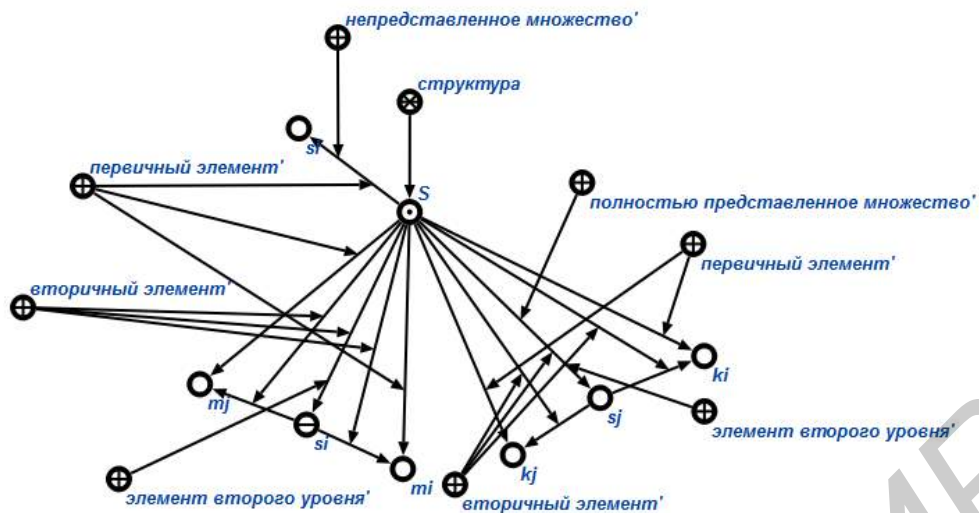


Рисунок 2 – Пример описания элементов структуры на SCg

семантический уровень структурного элемента вычисляется следующим образом:

- элементы структуры, входящие в нее с атрибутом *первичный элемент'* имеют семантический уровень 1;
- уровень элемента, не являющегося *первичным элементом'* структуры, вычисляется путем прибавления 1 к максимальному из уровней элементов этого элемента (множества), входящих в эту же структуру. Например, *sc-дуга*, соединяющая два *первичных элемента'* структуры будет иметь семантический уровень 2, а *sc-элемент*, обозначающий отношение, которому принадлежит указанная *sc-дуга* – семантический уровень 3.



Рисунок 3 – Семантический уровень структурного элемента

В свою очередь, **количество семантических уровней элементов структуры** – параметр, каждый элемент которого представляет собой класс *структур*, у которых совпадает максимальный среди семантических уровней элементов этих структур. Значением данного параметра является число, совпадающее с указанным максимальным семантическим уровнем элементов.

1.3. Соответствия, заданные на структурах

На структурах может быть задан ряд *соответствий**, таких как *гомоморфизм**, *полиморфизм**, *автоморфизм**, *изоморфизм** и т.д.

Отдельное внимание стоит уделить соответствию *аналогичность структур**, которое фиксирует факт наличия некоторой аналогии на подструктурах (подмножествах) указанных структур. Каждой ориентированной паре, принадлежащей *аналогичности структур** может быть поставлено в соответствие множество пар, задающих *сходства** некоторых подструктур и *различия** некоторых подструктур исходных структур.

Пример отношения *аналогичности структур** приведен на рисунке 4.

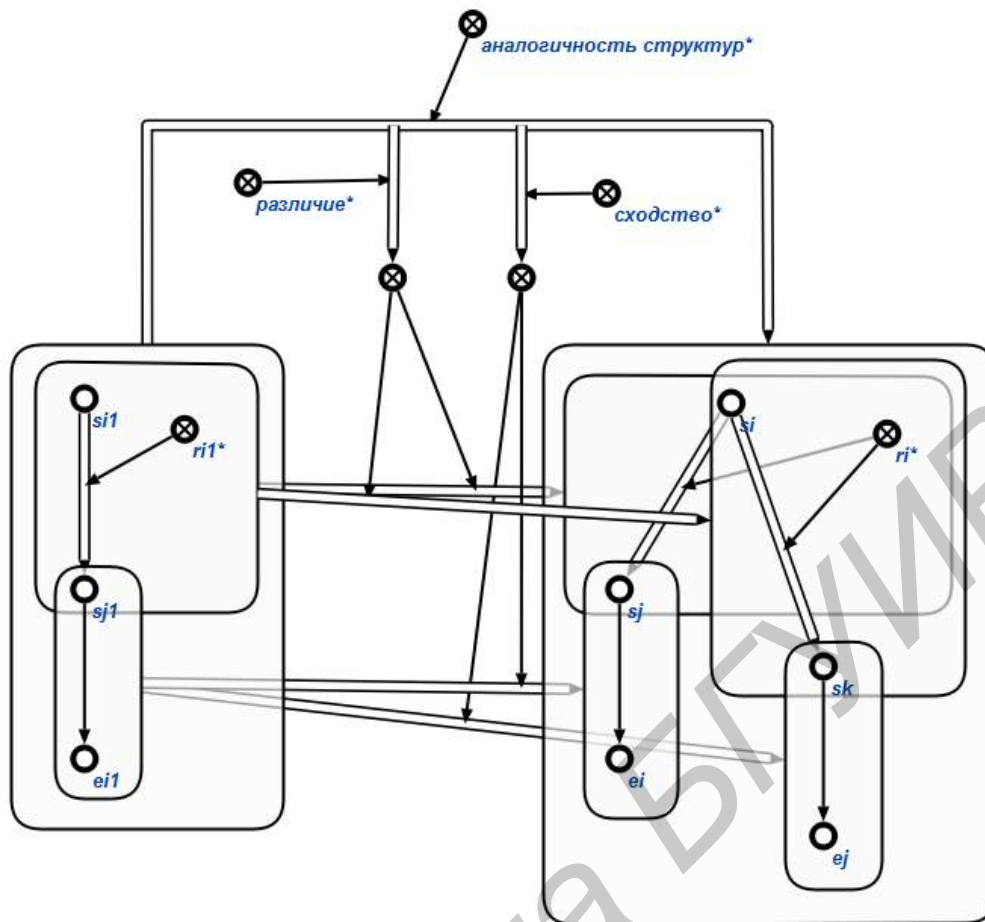


Рисунок 4 - Пример отношения аналогичности структур

2. Предметная область знаний

В рамках базы знаний будем выделять семантически осмысленные *структуры*, обладающие некоторой семантической целостностью. Такие структуры будем называть *знаниями*.

Рассмотрим типологию знаний:

знание

- = *sc-знание*
- = Множество всевозможных знаний
- = *sc-знание* или целостный фрагмент *sc-знания*
- ⊂ структура
- ⊃ семантическая окрестность
- ⊃ сравнение
- ⊃ фактографическое знание
- ⊃ раздел
- ⊃ предметная область
- ⊃ онтология
- ⊃ задача
- ⊃ программа
- ⊃ план
- ⊃ решение
- ⊃ утверждение
- ⊃ определение

Важнейшим отношением, заданным на множестве знаний, является отношение *быть метазнанием**, описывающее переход от знаний к описывающим их метазнаниям [Гаврилова и др., 2001].

метазнание*

- = *быть метазнанием**
- = Бинарное ориентированное отношение, связки которого связывают некоторое исходное знание со знанием, которое является метаотношением исходного знания, его спецификацией, описанием его структуры*

Примером связи между знанием и соответствующим ему *метазнанием** является переход от некоторого исходного знания к описанию его декомпозиции (сегментации) на некоторые части с указанием связей между этими частями.

В *Предметной области знаний* исследуется ряд ролевых отношений, уточняющих семантику *sc-элементов* структур, являющихся *знанием*, в частности, *ключевой sc-элемент* и *главный ключевой sc-элемент*.

ключевой sc-элемент' – ролевое отношение, связывающее знак каждой структуры (текста, знания) со специально выделяемыми (ключевыми) элементами этой структуры.

Каждая структура может иметь либо несколько ключевых элементов, либо один ключевой элемент, либо ни одного.

ключевой sc-элемент'

- = быть ключевым sc-элементом заданного sc-знания'
- = ключевой знак'
- = ключевой элемент sc-знания'
- = быть ключевым элементом заданной sc-структуры
- = быть ключевым элементом заданной sc-знания
- ⊃ основной sc-элемент'
- ∈ ролевое отношение

главный ключевой sc-элемент' – ролевое отношение, связывающее знак некоторой структуры (текста, знания) со специально выделяемыми главными элементами этой структуры. Главный элемент некоторой структуры является ее **ключевым sc-элементом'**, но имеющим особый статус по каким-либо причинам, зависящим в каждом случае от семантики данного элемента и типа структуры.

главный ключевой sc-элемент'

- = быть ключевым sc-элементом заданного sc-знания'
- ⊂ ключевой sc-элемент'

Рассмотрим более подробно виды знаний, необходимые для описания структуры базы знаний, такие как *раздел*, *семантическая окрестность*, *предметная область* и *онтология*.

2.1. Понятие раздела базы знаний и их типология

База знаний каждой *ostis-системы* декомпозируется на *разделы*.

Понятие *раздела* представляет собой знак множества всевозможных разделов, входящих в состав различных баз знаний. Каждый раздел является условно дидактически выделяемым фрагментом базы знаний, обладающим логической целостностью и завершенностью. Вся база знаний конкретной *ostis-системы* также может рассматриваться как один неатомарный *раздел*.

На понятии раздела заданы отношения *декомпозиция раздела** и *базовый порядок разделов**.

*декомпозиция раздела** – это *квазибинарное отношение* между разделом и множеством его подразделов. Данное отношение задает дидактическую структуру раздела. В отличие от отношения *базовая декомпозиция**, которое связывает некоторую сущность с другими сущностями, являющимися её частями, отношение *декомпозиция раздела** связывает раздел с его подразделами, т.е. сужается область определения отношения *базовая декомпозиция**.

*базовый порядок разделов** – это бинарное отношение между разделами, определяющее

порядок их следования в рамках декомпозиции раздела.

Данное отношение задает дидактический порядок следования подразделов в рамках декомпозиции более общего раздела.

Выделяется следующая типология разделов базы знаний:

раздел

- = раздел базы знаний
- = sc-модель раздела базы знаний
- <= разбиение*:

- {
- атомарный раздел
- неатомарный раздел
- }

- ⊃ раздел-обоснование
- ⊃ раздел-описание принципов
- ⊃ раздел-документация
- ⊃ раздел-описание предметной области
- ⊃ раздел-описание семантической окрестности
- ⊃ раздел-теория предметной области

Под **неатомарным разделом** понимается знак множества всевозможных неатомарных разделов, входящих в состав различных документаций, то есть разделов, которые декомпозируются на более частные разделы.

Соответственно, под **атомарным разделом** понимается знак множества всевозможных атомарных разделов, входящих в состав различных документаций, то есть разделов, не декомпозируемых на более частные разделы.

Любое понятие, входящее в состав атомарного раздела и не являющееся ключевым в рамках этого раздела должно стать ключевым в рамках какого-либо другого атомарного раздела рассматриваемой базы знаний.

Для каждого раздела необходимо явно указать принадлежность к множеству атомарных или неатомарных разделов.

С точки зрения описываемой информации можно выделить следующие классы разделов:

- **раздел-обоснование** представляет собой формальный текст, обосновывающий разработку чего-либо, то есть постановку проблемы, указание недостатков существующих решений такого рода, достоинства предлагаемых решений, предполагаемый эффект от применения разработанных моделей, методов и средств и т.д.

- **раздел-документация** содержит документацию к чему-либо, как правило – к *ostis-системе* или ее компонентам. Документация описывает назначение системы или подсистемы, принципы работы с той или иной системой или подсистемой, ее структуру и состав. В случае многократно используемого компонента документация также содержит руководство по доработке и использованию такого компонента.

- **раздел-описание предметной области** содержит знак самой этой *предметной области*, знаки всех ее элементов, а также всю спецификацию этой *предметной области*, включая все ее онтологии, в том числе – логическую онтологию, содержащую описание формальной теории, соответствующей данной *предметной области*.

- **раздел-описание семантической окрестности** содержит знак *структуры*, которая является семантической окрестностью некоторого *sc-элемента*, знаки всех ее элементов, а так же спецификацию этой *структуры*, например, указание принадлежности более частному классу семантических окрестностей.

- **раздел-описание принципов** содержит описание основополагающих принципов устройства либо функционирования какой-либо технологии, подсистемы, компонента и т.д.

2.2. Отношения, задающие дополнительную спецификацию разделов базы знаний

Любой *раздел* базы знаний может дополнительно быть специфицирован с использованием следующих отношений:

- **комментарий*** – бинарное отношение, связывающее некоторую сущность (*sc-текст* или знак файла) со знаком *sc-текста*, являющегося комментарием к этой сущности.

- **аннотация*** – бинарное отношение, связывающее некоторый *раздел* и *sc-текст*, являющийся аннотацией к данному *разделу*. Как правило, аннотация содержит краткое (недетализированное) описание того, чему посвящен данный *раздел*.

- **введение*** – бинарное отношение, связывающее некоторый *раздел* и *sc-текст*, являющийся введением к данному *разделу*. Как правило, введение содержит информацию о том, как сущности, рассматриваемые в данном *разделе*, связаны с сущностями, описанными в других *разделах*.

- **заключение*** – это бинарное отношение, связывающее некоторый *раздел* и *sc-текст*, являющийся заключением к данному *разделу*. В заключении, как правило, подводятся итоги и указываются основные положения, описанные в *разделе*, а также рассматриваются направления развития самого *раздела* и связанных с ним.

- **предисловие*** – это бинарное отношение, связывающее некоторый *раздел* и *sc-текст*, являющийся предисловием к данному *разделу*.

- **послесловие*** – это бинарное отношение, связывающее некоторый *раздел* и *sc-текст*, являющийся послесловием к данному *разделу*.

- **эпиграф*** – это бинарное отношение, связывающее некоторый *раздел* и *sc-текст*, являющийся эпиграфом к данному *разделу*.

2.3. Предметная область семантических окрестностей

Для спецификации отдельных *сущностей* в рамках базы знаний вводится понятие **семантической окрестности**.

семантическая окрестность – это знание, являющееся спецификацией (описанием) некоторой *сущности*, знак которой является **ключевым элементом** указанного знания. Заметим, что каждая **семантическая окрестность** в отличие от знаний других видов имеет только один ключевой элемент (ключевой знак, знак описываемой сущности). Также отметим, что многообразие видов семантических окрестностей свидетельствует о многообразии семантических видов описаний различных сущностей.

семантическая окрестность

= *sc-окрестность*

= *семантическая окрестность, представленная в виде sc-текста*

= *sc-текст, являющийся семантической окрестностью некоторого sc-элемента*

= *спецификация заданной сущности, знак которой указывается как ключевой элемент этой спецификации*

= *описание заданной сущности, знак которой указывается как ключевой элемент этой спецификации*

⊆ *знание*

⊃ *семантическая окрестность по инцидентным коннекторам*

⊃ *полная семантическая окрестность*

⊃ *базовая семантическая окрестность*

⊃ *специализированная семантическая окрестность*

Перечислим основные виды *семантических окрестностей*.

семантическая окрестность по инцидентным коннекторам – это вид *семантической окрестности*, в которую входят знаки всех коннекторов, инцидентных заданному элементу, а также знаки всех элементов, инцидентных указанным коннекторам.

семантическая окрестность по инцидентным коннекторам

⊃ *семантическая окрестность по выходящим дугам*

⊃ *семантическая окрестность по входящим дугам*

семантическая окрестность по выходящим дугам – это вид *семантической окрестности*, в которую входят знаки всех дуг, выходящих из заданного *sc-элемента*, а также знаки их вторых компонентов, также указывается факт принадлежности этих дуг каким-либо отношениям.

семантическая окрестность по выходящим дугам

⊃ *семантическая окрестность по выходящим дугам принадлежности*

семантическая окрестность по выходящим дугам принадлежности – это вид *семантической окрестности*, в которую входят знаки всех дуг принадлежности, выходящих из заданного *sc-*

элемента, а также знаки их вторых компонентов. При необходимости может указываться факт принадлежности этих дуг каким-либо ролевым отношениям.

семантическая окрестность по входящим дугам – это вид *семантической окрестности*, в которую входят знаки всех дуг, входящих в заданный *sc-элемент*, а также знаки их первых компонентов, также указывается факт принадлежности этих дуг каким-либо отношениям.

семантическая окрестность по входящим дугам
⊃ *семантическая окрестность по входящим дугам принадлежности*

семантическая окрестность по входящим дугам принадлежности – это вид *семантической окрестности*, в которую входят знаки всех дуг принадлежности, входящих в заданный *sc-элемент*, а также знаки их первых компонентов. При необходимости может указываться факт принадлежности этих дуг каким-либо ролевым отношениям.

полная семантическая окрестность – это вид *семантической окрестности*, включающий описание всех связей описываемой сущности.

Структура *полной семантической окрестности* определяется прежде всего семантической типологией описываемой сущности.

Так, например, для *понятия* в полную семантическую окрестность необходимо включить следующую информацию (при наличии):

- варианты идентификации на различных внешних языках;
- принадлежность некоторой предметной области с указанием роли, выполняемой в рамках этой предметной области;
- теоретико-множественные связи заданного понятия с другими *sc-элементами*;
- определение или пояснение;
- высказывания, описывающие свойства указанного понятия;
- задачи и их классы, в которых данное понятие является ключевым
- описание типичного примера использования указанного понятия;
- экземпляры описываемого понятия.

Для *понятия*, являющегося отношением дополнительно указываются:

- домены;
- область определения;
- схема отношения;
- классы отношений, которым принадлежит описываемое отношение.

базовая семантическая окрестность – это вид *семантической окрестности*, содержащий минимальную (краткую) информацию об описываемой сущности

Структура *базовой семантической окрестности* определяется прежде всего семантической типологией описываемой сущности.

Так, например, для *понятия* в базовую семантическую окрестность необходимо включить следующую информацию (при наличии):

- варианты идентификации на различных внешних языках;
- принадлежность некоторой предметной области с указанием роли, выполняемой в рамках этой предметной области;
- определение или пояснение.

Для *понятия*, являющегося отношением дополнительно указываются:

- домены;
- область определения;
- описание типичного примера использования указанного отношения.

базовая семантическая окрестность

= минимально достаточная семантическая окрестность

= минимальная спецификация описываемой сущности

= сокращенная спецификация описываемой сущности

= основная семантическая окрестность

специализированная семантическая окрестность – это вид *семантической окрестности*, набор связей для которой уточняется отдельно для каждого класса такой окрестности.

специализированная семантическая окрестность

⊃ *пояснение*

⊃ *примечание*

⊃ *правило идентификации экземпляров*

⊃ *терминологическая семантическая окрестность*

⊃ *теоретико-множественная семантическая окрестность*

⊃ *логическая семантическая окрестность*

⊃ *описание типичного экземпляра*

⊃ *описание декомпозиции*

пояснение – знак *sc-текста*, поясняющего описываемую сущность.

примечание – знак *sc-текста*, являющегося примечанием к описываемой сущности. В примечании обычно описываются особые свойства и исключения из правил для описываемой сущности.

правило идентификации экземпляров – это *sc-текст* являющийся описанием правил построения идентификаторов элементов заданного класса.

терминологическая семантическая окрестность – семантическая окрестность, описывающая идентификацию указанной сущности.

теоретико-множественная семантическая окрестность – описание связи описываемого

понятия с другими понятиями с помощью теоретико-множественных отношений.

описание декомпозиции – семантическая окрестность, описывающая декомпозицию некоторой сущности на частные сущности.

логическая семантическая окрестность – семантическая окрестность, описывающая семейство высказываний, описывающих свойства данного понятия.

описание типичного экземпляра – это *sc-текст* являющийся описанием типичного примера использования рассматриваемого класса.

сравнительный анализ – описание сравнительного анализа некоторой сущности с другими сущностями.

3. Предметная область предметных областей

Предметная область – это важнейший вид знаний, входящих в состав смыслового пространства [Голенков, 2015]. Каждая предметная область фокусирует внимание на описание связей соответствующего класса объектов исследования. Каждый знак, входящий в состав базы знаний, должен принадлежать (входить в состав) хотя бы одной предметной области, выполняя в ней ту или иную роль. Каждой предметной области можно поставить в соответствие:

- множество семантических окрестностей, описывающих объекты исследования этой предметной области;
- семейство различного вида онтологий, описывающих свойства понятий этой предметной области.

Предметные области являются основой структуризации смыслового пространства, средством локализации, фокусирования внимания на свойства наиболее важных классов описываемых сущностей, которые становятся классами объектов исследования в *предметных областях*.

Понятие базы знаний тесно связано с понятием *предметной области*. Соотношение между базой знаний и описываемой ею предметной областью задает семантику базы знаний компьютерной системы.

Рассмотрение структуры базы знаний во взаимосвязи с предметной областью позволяет рассматривать исследуемые объекты на разных уровнях детализации. Детализацию рассмотрения исследуемых объектов можно осуществлять как в рамках исходной предметной области, так и в системе самостоятельных, но связанных между собой предметных областей.

При переходе от предметной области к ее модели, представленной в виде семантической сети, выполняются следующие условия:

- каждому элементу предметной области взаимно однозначно соответствует обозначающий его элемент семантической сети;

- каждому сигнатурному элементу предметной области взаимно однозначно соответствует либо обозначающий его ключевой узел семантической сети, либо обозначающий элемент алфавита семантической сети.

В состав **Предметной области предметных областей** входят структурные спецификации всех *предметных областей*, входящих в состав базы знаний *ostis-системы*, в том числе, самой **Предметной области предметных областей**. Таким образом, **Предметная область предметных областей** является, во-первых, *рефлексивным множеством*, во-вторых, рефлексивной предметной областью, то есть *предметной областью*, одним из объектов исследования которой является она сама.

Предметная область – это результат интеграции (объединения) частичных семантических окрестностей, описывающих все исследуемые сущности заданного класса и имеющих одинаковый (общий) предмет исследования (то есть один и тот же набор отношений, которым должны принадлежать связи, входящие в состав интегрируемых семантических окрестностей).

Предметная область представляет собой *структуру*, в состав которой входят:

- (1) основные исследуемые (описываемые) объекты – первичные и вторичные;
- (2) различные классы исследуемых объектов;
- (3) различные связи, компонентами которых являются исследуемые объекты (как первичные, так и вторичные), а также, возможно, другие такие связи – то есть связи (как и объекты исследования) могут иметь различный структурный уровень;
- (4) различные классы указанных выше связей (то есть отношения);
- (5) различные классы объектов, не являющихся ни объектами исследования, ни указанными выше связями, но являющихся компонентами этих связей.

При этом все классы, объявленные исследуемыми понятиями, должны быть полностью представлены в рамках данной предметной области вместе со своими элементами, элементами элементов и т.д. вплоть до терминальных элементов.

Понятие **предметной области** является важнейшим методологическим приемом, позволяющим выделить из всего многообразия исследуемого Мира только определенный класс исследуемых сущностей и только определенное семейство отношений, заданных на указанном классе. То есть осуществляется локализация, фокусирование внимания только на этом, абстрагируясь от всего остального исследуемого Мира. [Голенков, 2012]

Во всем многообразии **предметных областей** особое место занимают:

(1) **Предметная область предметных областей**, объектами исследования которой являются всевозможные **предметные области**, а предметом исследования – всевозможные **ролевые отношения**, связывающие предметные области с их элементами, отношения, связывающие предметные области между собой, отношения, связывающие предметные области с их онтологиями

(2) **Предметная область сущностей**, являющаяся предметной областью самого высокого уровня и задающая базовую семантическую типологию *sc-элементов* (знаков, входящих в тексты *SC-кода*)

(3) Семейство **предметных областей**, каждая из которых задает семантику и синтаксис некоторого *sc-языка*, обеспечивающего представление онтологий соответствующего вида (например, *теоретико-множественных онтологий*, *логических онтологий*, *терминологических онтологий*, *онтологий задач и способов их решения* и т.д.)

(4) Семейство **предметных областей** верхнего уровня, в которых классами объектов исследования являются весьма «крупные» классы сущностей. К таким классам, в частности, относятся:

- класс всевозможных *материальных сущностей*,
- класс всевозможных *множеств*,
- класс всевозможных *связей*,
- класс всевозможных *отношений*,
- класс всевозможных *структур*,
- класс всевозможных *временных (нестационарных) сущностей*,
- класс всевозможных *действий* (акций),
- класс всевозможных *параметров* (характеристик),
- класс *знаний* всевозможного вида
- и т.п.

Каждой **предметной области** можно поставить в соответствие:

(1) семейство соответствующих ей *онтологий* разного вида;

(2) некий *язык* (в нашем случае – язык, построенный на основе *SC-кода*), тексты которого представляют различные фрагменты соответствующей предметной области.

Указанные языки будем называть *sc-языками*. Их синтаксис и семантика полностью задается *SC-кодом* и *онтологией* соответствующей **предметной области**. Очевидно, что в первую очередь нас должны интересовать те *sc-языки*, которые соответствуют **предметным областям**, имеющим общий (условно говоря, предметно независимый) характер. К таким предметным областям, в частности, относятся:

- **Предметная область множеств**, описывающая множества и различные связи между ними
- **Предметная область графовых структур**
- **Предметная область чисел** и числовых структур
- и т.д.

Каждому типу знаний можно поставить в соответствие предметную область, которая является результатом интеграции всех знаний данного типа. Эти знания и становятся объектами исследования в рамках указанной предметной области

Понятие **предметной области** может рассматриваться как обобщение понятия алгебраической системы. При этом семантическая структура базы знаний может рассматриваться как иерархическая система различных **предметных областей**.

3.1. Типология предметных областей

Можно говорить о типологии **предметных областей** по разным структурным признакам:

- наличие метасвязей;
- наличие исследуемых структур, входящих в состав предметной области;
- наличие исследуемых (смежных, дополнительных) объектов, которых исследуются в других предметных областях.

Одним из основных критериев классификации **предметных областей** является зависимость от времени связей между сущностями, входящими в состав рассматриваемой **предметной области**.

предметная область

= *sc-модель предметной области*

= *sc-текст предметной области*

= *sc-граф предметной области*

= *представление предметной области в SC-коде*

⊆ *знание*

⊆ *бесконечное множество*

⊆ *разбиение**:

- {
- *стационарная предметная область*
- *нестационарная предметная область*
- }

стационарная предметная область - это **предметная область**, в которой связи между сущностями, входящими в ее состав, не зависят от времени (не меняются во времени). При этом некоторые из указанных сущностей могут иметь конечное время "жизни" (конечное время существования).

Таким образом, элементами **стационарной предметной области** не могут быть *временные сущности*.

нестационарная предметная область - это **предметная область**, в которой некоторые связи между сущностями, входящими в ее состав,

меняются со временем (то есть носят ситуационный, нестационарный характер, другими словами, являются *временными сущностями*).

3.2. Отношения, задающие роли элементов предметной области

Любая *предметная область* с формальной точки зрения является *структурой*. Уточним роли элементов, входящих в состав *предметной области*, которые являются частным случаем ролей элементов структур, рассмотренных выше.

понятие предметной области' – это *ролевое отношение*, указывающее в рамках *предметной области* на знак множества, являющегося классом некоторых объектов.

понятие предметной области'

\leq разбиение*:

- {
- *исследуемое понятие'*
- *понятие, исследуемое в частной предметной области'*
- *понятие, исследуемое в более общей предметной области'*
- *понятие, исследуемое в неродственной предметной области'*
- }

\in *неосновное понятие*

понятие – класс, являющийся исследуемым понятием хотя бы для одной *предметной области*.

понятие

\leq *строгое включение**:

класс

\leq *второй домен**:

понятие предметной области'

исследуемое понятие' – это *ролевое отношение*, указывающее в рамках *предметной области* на знак множества, являющегося классом объектов исследования данной *предметной области*, то есть такого множества, все элементы которого являются элементами данной *предметной области*.

исследуемое понятие' может быть:

(1) классом *первичных элементов'* этой *предметной области*;

(2) отношением (классов связей), связывающих элементы этой *предметной области* – уровень иерархии этих элементов может быть различным;

(3) классом *структур*, все элементы которых являются элементами заданной *предметной области*, т.е. классом подструктур заданной *предметной области*.

исследуемое понятие'

= *класс объектов исследования данной предметной области'*

= *понятие, исследуемое в данной предметной области'*

\in *ролевое отношение*

\leq *разбиение**:

- {
- *максимальный класс объектов исследования'*
- *немаксимальный класс объектов исследования'*
- }

\leq *разбиение**:

- {
- *исследуемый класс первичных элементов'*
- *исследуемое отношение'*
- *класс исследуемых структур'*
- }

\leq *включение**:

полностью представленное множество'

максимальный класс объектов исследования'

это *ролевое отношение*, указывающее в рамках *предметной области* на множество, являющееся максимальным классом объектов исследования данной *предметной области*, то есть на такое *исследуемое понятие'*, для которого в рамках данной *предметной области* не существует другого *исследуемого понятия'*, которое бы являлось надмножеством для данного.

немаксимальный класс объектов исследования'

это *ролевое отношение*, указывающее в рамках *предметной области* на такое *исследуемое понятие'*, для которого в рамках данной *предметной области* существует другое *исследуемое понятие'*, являющееся надмножеством первого.

исследуемый класс первичных элементов'

такое *исследуемое понятие'* для данной *предметной области*, что все его элементы являются ее первичными элементами'.

исследуемое отношение'

это *ролевое отношение*, указывающее в рамках *предметной области* на множество связей, являющееся исследуемым отношением данной *предметной области*, то есть таким отношением, все связки которого являются элементами этой *предметной области*.

При этом элементы таких связей также входят в данную *предметную область*, но в общем случае могут не являться элементами *исследуемых понятий'* данной *предметной области*.

класс исследуемых структур' – это *ролевое отношение*, указывающее в рамках *предметной области* на множество *структур*, знак каждой из которых принадлежит данной *предметной области*.

В общем случае в данную *предметную область*, могут входить не все элементы таких *структур*, а только некоторые из них (хотя бы один для каждой *структуры*).

понятие, исследуемое в частной предметной области'

это *понятие предметной области'* данной *предметной области*, которое является *исследуемым понятием'* для какой-либо из *частных предметных областей** относительно данной.

понятие, исследуемое в предметной подобласти'
= класс исследуемых объектов, который детально исследуется в предметной области, которая является частной по отношению к заданной предметной области'

∈ ролевое отношение

≤ включение*:

частично представленное множество'

понятие, исследуемое в более общей предметной области' – это понятие предметной области', которое является исследуемым понятием' для какой-либо из предметных областей, для которых данная предметная область является частной предметной областью*.

понятие, исследуемое в более общей предметной области'

≤ включение*:

частично представленное множество'

понятие, исследуемое в неродственной предметной области' – это понятие предметной области', которое является исследуемым понятием' для какой-либо из предметных областей, являющихся неродственными предметными областями* для данной.

понятие, исследуемое в неродственной предметной области'

≤ включение*:

частично представленное множество'

3.3. Отношения, заданные на предметных областях

На множестве предметных областей могут быть заданы теоретико-множественные отношения: *включение**, *объединение**, *пересечение**, *декомпозиция**, *гомоморфизм**, *изоморфизм**, а также специальные отношения, область определения которых является множеством предметных областей. Рассмотрим некоторые из них.

частная предметная область* – бинарное ориентированное отношение, с помощью которого задается иерархия предметных областей путем перехода от менее детального к более детальному рассмотрению соответствующих исследуемых понятий.

частная предметная область*

= дочерняя предметная область*

= быть частной предметной областью*

= предметная область, детализирующая описание одного из классов объектов исследования другой (более общей) предметной области*

∈ бинарное отношение

∈ ориентированное отношение

∈ неролевое отношение

⊃ частная предметная область по классу первичных элементов*

⊃ частная предметная область по исследуемым отношениям*

частная предметная область по классу первичных элементов*

= сужение предметной области по классу первичных элементов*

частная предметная область по исследуемым отношениям*

= частная предметная область по предмету исследования*

= сужение предметной области по предмету исследования*

Связки отношения **неродственные предметные области*** связывают две предметные области, имеющие общие элементы, однако не связанные отношением частная предметная область*.

неродственные предметные области*

∈ бинарное отношение

∈ неориентированное отношение

Рассмотренные отношения являются частным случаем отношения **метазнание***.

Каждая предметная область задает соответствующий специализированный язык (**sc-язык**), представляющий собой множество всевозможных фрагментов (подструктур), включаемых в состав этой предметной области.

sc-язык – это подязык (подмножество) **SC-кода**, ориентированный на представление **sc-текстов**, являющихся фрагментами некоторой предметной области. Таким образом, каждому **sc-языку** взаимно однозначно соответствует некоторая предметная область (точнее, **sc-модель** некоторой предметной области).

Отношение **sc-язык и соответствующая предметная область*** – это бинарное ориентированное отношение, каждая связка которого связывает знак некоторого **sc-языка** (под первым атрибутом) и знак соответствующей этому **sc-языку** предметной области.

4. Предметная область онтологий

Для спецификации (описания свойств) соответствующей предметной области, ориентированной на описание свойств и взаимосвязей понятий, входящих в состав указанной предметной области используется такой вид знаний, как **онтология** [Кудрявцев, 2010], [Федотова, 2015].

онтология

= система понятий соответствующей предметной области

= концептуальный каркас (скелет) описания некоторой предметной области

= концептуальная (семантическая) основа различных языков, обеспечивающих описание объектов исследования, принадлежащих заданной предметной области

= семантический интерфейс для интеграции знаний по заданной предметной области и для согласованного понимания различными

- субъектами этих знаний
- = онтология соответствующей предметной области
- = описание концептов и отношений заданной предметной области
- <= включение*:
знание
- <= разбиение*:
{
 - интегрированная онтология
 - структурная спецификация
 - теоретико-множественная онтология
 - логическая иерархия понятий
 - логическая онтология
 - логическая иерархия высказываний
 - терминологическая онтология
 - онтология задач и решений задач
 - онтология классов задач и способов решения задач

Связь между предметной областью и ее онтологией задается отношением *онтология** (*быть онтологией**), которое является частным видом отношения *метазнание** (*быть метазнанием**).

онтология* – это бинарное отношение, связывающее некоторую предметную область с ее онтологией (спецификацией).

- онтология***
- = *sc-онтология**
- = *быть онтологией предметной области**
- = *sc-онтология, специфицирующая заданную предметную область**
- => *первый домен**:
предметная область
- => *второй домен**:
онтология

В зависимости от рассматриваемых свойств понятий предметной области, которые описываются в онтологии, выделяют следующие типы онтологий:

- **структурная спецификация** – это онтология, в которой описываются роли понятий, входящих в состав предметной области, а также связи специфицируемых предметных областей с другими предметными областями.
- **теоретико-множественная онтология** – это онтология, описывающая теоретико-множественные связи между понятиями заданной предметной области (включение, разбиение, объединение, пересечение, разность множеств, область определения, домен, функция)
- **логическая онтология** – это онтология, описание системы высказываний заданной предметной области.
- **логическая иерархия понятий** – это онтология, являющаяся надстройкой над логической онтологией, включающая описание системы определений понятий заданной предметной области с указанием набора понятий,

через которые определяется каждое определяемое понятие рассматриваемой предметной области.

- **используемые константы*** – это отношение, связывающее понятие со множеством понятий, на основании которых оно определяется в рамках рассматриваемой предметной области.

- **логическая иерархия высказываний** – это онтология, являющаяся надстройкой над логической онтологией и включающая описание системы утверждений рассматриваемой предметной области с указанием набора утверждений, через которые доказывается каждое утверждение.

- **используемые утверждения*** – это отношение, связывающее утверждение со множеством утверждений, на основании которых оно доказывается в рамках рассматриваемой предметной области

- **терминологическая онтология** – это онтология, описывающая систему основных и неосновных терминов (имен, внешних обозначений), соответствующих концептам и отношениям заданной предметной области, а также описание правил построения терминов для сущностей, являющихся элементами (экземплярами) указанных концептов и отношений.

- **онтология задач и решений задач** – это онтология, описывающая задачи и их классы, решаемые в рассматриваемой предметной области

- **онтология классов задач и способов решения задач** – это онтология, описывающая способы решения задач и их классов в рамках предметной области. Является надстройкой над онтологией задач и классов задач.

- **интегрированная онтология** – это онтология, объединяющая все онтологии различного вида некоторой предметной области.

Заключение

Рассмотренные в работе средства позволяют осуществлять семантическую структуризацию баз знаний любой системы, построенной по *Технологии OSTIS* унифицированным образом.

Семантическая структура базы знаний представляет собой иерархическую систему описываемых ею предметных областей, надстраиваемых над заданной основной предметной областью.

Построение семантической структуры базы знаний требует не только явного представления спецификации каждой описываемой предметной области в виде формального текста, но и явного описания всевозможных связей между этими предметными областями.

В работе представлены результаты, полученные в научных группах МГТУ им. Н.Э.Баумана (кафедра «Компьютерные системы автоматизации производства») и БГУИР (кафедра интеллектуальных информационных технологий) совместно.

Работа выполнена при поддержке гранта БРФФИ-РФФИ-М «Методы и средства онтологического моделирования для семантических технологий проектирования интеллектуальных систем» (Ф15PM-074, 15-57-04047).

Библиографический список

[Гаврилова и др., 2001] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник / Гаврилова Т.А. [и др.]; – СПб.: Изд-во «Питер», 2001.

[Голенков, 2012] Голенков, В.В., Гулякина Н.А. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования. – В кн Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2012). Материалы конф. [Минск, 16-18 февр. 2012 г.]. – Минск: БГУИР, 2012.

[Голенков, 2015] Голенков, В.В. Семантическая технология компонентного проектирования систем, управляемых знаниями. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар. научн.-техн. конф. / В. В. Голенков, Н.А Гулякина // Мн.: БГУИР, 2015

[Кудрявцев, 2010] Кудрявцев Д.В. Системы управления знаниями и применение онтологий: Учеб. пособие / Д.В. Кудрявцев. - СПб.: Изд-во Политехн. ун-та, 2010.

[Тузовский и др., 2005] Тузовский А. Ф., Чириков С. В., Ямпольский В. З. (2005) Системы управления знаниями. — Томск: Изд-во науч.-техн. литературы.

[Федотова, 2015] Федотова А.В., Давыденко И.Т. Онтологическое моделирование технического обслуживания // Материалы V Международной научно-технической конференции OSTIS-2015 (Минск, 19–21 февраля 2015 г.) – Минск: БГУИР, 2015. – С. 429-438.

[Tarassov, 2015] Tarassov V.B., Fedotova A.V., Stark R., Karabekov B.S. Granular Meta-Ontology and Extended Allen's logic: Some Theoretical Background and Application to Intelligent Product Lifecycle Management Systems // Proceedings of the 4th International Conference on Intelligent Systems and Applications (INTELLI'2015, St. Julians, Malta, October 11-16, 2015). – Copenhagen: IARIA XPS Press, 2015. – P.86-93. ISBN: 978-1-61208-437-4.

[IMS, 2016] Метасистема IMS.OSTIS [Электронный ресурс]. Минск, 2016. – Режим доступа: <http://ims.ostis.net/>. – Дата доступа: 15.01.2016.

THE TOOLS OF STRUCTURING THE SEMANTIC MODELS OF KNOWLEG BASES

Davydenko I.T. *, Grakova N.V. *,
Sergienko E.S. **, Fedotova A.V.***

*Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus

ir.davydenko@gmail.com

natalia.grakova@gmail.com

** National Research Nuclear University MEPHI
(Moscow Engineering Physics Institute), Moscow,
Russia Federation

deav@inbox.ru

*** Bauman Moscow State Technical University,
Moscow, Russia

afedotova.bmstu@gmail.com

The paper presents the tools of structuring the semantic models of knowleg bases, based on unified semantic networks. The main attention is paid for such concepts as structure, knowledge, considered their classification. Separately section describes the types of knowledge such as ontologies and subject domain.

Key words: knowledge base, knowledge base structuration, subject domain.

Introduction

Structuring of knowledge bases, highlighted in it various interrelated substructures needed for a variety of reasons. It is necessary for didactic aims, and for organizing the distribution of work on the knowledge bases design.

Main part

In this work introduced and refined one of the basic terms of knowledge representation OSTIS technology, the concept of structure, which is the basis for the concept knowledge, one type of which is the concept of subject domain. For the subject domain specification introduces the concept of ontology and considered their topology.

Conclusion

The tools, considered in this work, allows realised the semantic structuring of knowledge base of each system, based on OSTIS technology unified way. The semantic structure of knowledge base is hierarchical system of subject domains, describe by it.

Construction of the semantic structure of the knowledge base requires not only the explicit representation of the specifications described each subject domain in the form of a formal text, but the explicit description of all possible connections between these domains.



УДК 004.822:514

СЕМАНТИЧЕСКАЯ МОДЕЛЬ КОЛЛЕКТИВНОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ

Давыденко И.Т.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

ir.davydenko@gmail.com

В работе рассматривается унифицированная модель коллективного проектирования баз знаний. Описывается унифицированная структура базы знаний системы, управляемой знаниями. Приведена классификация пользователей систем, управляемых знаниями, а также типология действий разработчиков баз знаний.

Ключевые слова: база знаний, система, управляемая знаниями, проектирование баз знаний.

Введение

В настоящее время все более актуальной становится задача эффективного информационного обеспечения научной и производственной деятельности, связанная с бурным ростом объемов информации в различных отраслях знаний. Данная задача, как правило, рассматривается в контексте создания хранилищ знаний и их систематизации и структуризации с целью облегчения их обработки.

Ускорение и упрощение процесса разработки интеллектуальных систем различного назначения является одной из важнейших задач *Проекта OSTIS*, направленного на разработку открытой технологии проектирования *компьютерных систем, управляемых знаниями*.

Компьютерная система, управляемая знаниями – система, в основе которой лежит представленная унифицированным образом база знаний, содержащая в систематизированном виде всю информацию, используемую этой системой. [Голенков, 2015]. В рамках данной работы будем рассматривать понятие системы, управляемой знаниями, построенной по *Технологии OSTIS*, которую будем называть *ostis-системой*.

База знаний является одним из ключевых компонентов компьютерных систем, управляемых знаниями различного назначения [Гаврилова и др., 2001], [Гаврилова, 2008], [Хорошевский, 2008], [Кудрявцев, 2010], [Тузовский и др., 2005]. Разработка этого компонента является трудоемким и продолжительным процессом.

Под *базой знаний ostis-системы* будем понимать конечную информационную конструкцию,

являющуюся формальным представлением всех знаний, достаточных для функционирования некоторой компьютерной системы и хранимых в памяти этой системы. [Голенков, 2015].

В рамках данной работы рассмотрим унифицированную модель базы знаний любой *ostis-системы*, а также модель коллективной разработки *баз знаний ostis-систем*.

В качестве формальной основы проектируемых логико-семантических моделей баз знаний интеллектуальных систем используются графодинамические модели специального вида – семантические модели представления и обработки знаний, в основе которых лежат унифицированные семантические сети с базовой теоретико-множественной интерпретацией. Основным способом кодирования информации для таких сетей является SC-код (Semantic Code) [Голенков, 2012].

Некоторые из рассматриваемых в данной работе понятий будут сопровождаться фрагментами их спецификации на формальном языке SCn. [IMS, 2016]

Каждая *ostis-система* состоит из *sc-модели компьютерной системы*, не зависящей от платформы реализации этой модели, и *платформы интерпретации sc-моделей*, обеспечивающей интерпретацию какого-либо класса *sc-моделей*.

ostis-система

\leq базовая декомпозиция*:

- {
- платформа интерпретации *sc-моделей*
- *sc-модель компьютерной системы*
- }

В свою очередь, *sc-модель компьютерной системы* декомпозируется следующим образом:

sc-модель компьютерной системы

<= базовая декомпозиция*:

- {
- *sc-модель базы знаний*
- *sc-модель машины обработки знаний*
- *sc-модель интерфейса компьютерной системы*
- }

В свою *sc-модель базы знаний* любой компьютерной системы, построенной по *Технологии OSTIS* имеет унифицированную структуру, которая будет подробнее рассмотрена далее.

Стоит отметить, что описываемые ниже разделы базы знаний отражают структуру базы знаний с точки зрения технологии разработки базы знаний.

1. Структура базы знаний систем, управляемых знаниями

Рассмотрим структуру *sc-модели базы знаний ostis-системы*:

sc-модель базы знаний

= унифицированная семантическая модель базы знаний

∈ раздел базы знаний

<= базовая декомпозиция*:

- {
- *предметная часть базы знаний*
- *контекст предметной части базы знаний в рамках Глобальной базы знаний*
- *документация компьютерной системы*
- *история и текущие процессы эксплуатации компьютерной системы*
- *история, текущие процессы и план развития компьютерной системы*
- }

предметная часть базы знаний содержит всю информацию о *предметной области* (или нескольких взаимосвязанных предметных областях в рамках этой же базы знаний) [Давыденко, 2013], для работы с которой предназначена та или иная система, управляемая знаниями (например - справочная система). Примерами разделов такого рода могут быть *Документация. Технология OSTIS* или *Документация. Геометрия Евклида*.

контекст предметной части базы знаний в рамках Глобальной базы знаний содержит спецификацию объектов, которые не исследуются непосредственно в предметной части базы знаний данной системы, но имеют к ней отношение, то есть, упоминаются при описании каких-либо понятий, исследуемых в предметной части базы знаний. Например, для системы *IMS* это могут быть такие понятия, как *искусственный интеллект* или *интеллектуальная система*, для системы по *Геометрии Евклида* – историческая справка о жизни Евклида, математики и т.д.

Раздел *документация компьютерной системы* содержит документацию самой *ostis-системы*, как минимум, спецификацию ее базы знаний, машины обработки знаний и интерфейса, а также все необходимые руководства, обеспечивающие возможность обучения работе с системой.

история и текущие процессы эксплуатации компьютерной системы

<= базовая декомпозиция*:

- {
- *история эксплуатации компьютерной системы*
- *текущие процессы эксплуатации компьютерной системы*
- }

В разделе *история эксплуатации компьютерной системы* хранится история диалога системы с ее пользователями, то есть спецификация всех действий, выполненных (уже ставших *прошлыми сущностями*) *системой* с целью удовлетворения информационных потребностей пользователей, в том числе, ответов на вопросы и осуществления указанных ими преобразований в базе знаний, в том числе – последовательность выполнения этих действий и полученный результат. При необходимости, часть таких описаний может удаляться, например, по истечении какого-либо установленного срока.

В разделе *текущие процессы эксплуатации компьютерной системы* хранятся спецификации всех действий, выполняемых *ostis-системой* в данный момент (входящих во множество *настоящих сущностей*), а также все временные вспомогательные конструкции, сгенерированные *sc-агентами* в процессе работы и пока еще не удаленные. После выполнения указанных действий их знаки и спецификации переносятся в раздел *история эксплуатации компьютерной системы*.

история, текущие процессы и план развития компьютерной системы

<= базовая декомпозиция*:

- {
- *структура и организация проекта компьютерной системы*
- *история развития компьютерной системы*
- *текущие процессы развития компьютерной системы*
- *план развития компьютерной системы*
- }

В разделе *структура и организация проекта компьютерной системы* описываются структура проекта, направленного на развитие *ostis-системы*, в том числе, указываются его подпроекты и роли разработчиков, ответственных за каждый проект.

В разделе *история развития компьютерной системы* находятся спецификации проектных действий, выполненных в процессе разработки системы (*прошлых сущностей*), с обязательным указанием исполнителей, последовательности и

результата выполнения.

В разделе *текущие процессы развития компьютерной системы* находятся спецификации утвержденных и инициированных проектных действий, выполняемых разработчиками системы в данный момент времени (*настоящих сущностей*), с обязательным указанием исполнителей, последовательности и цели выполнения, а также вся информация, описывающая предложения по редактированию *предметной части базы знаний и истории эксплуатации компьютерной системы* и их обсуждение администраторами, менеджерами и экспертами.

В разделе *план развития компьютерной системы* находятся спецификации проектных действий, которые утверждены к выполнению, но пока еще не выполняются по каким-либо причинам, а также вся информация, описывающая предложения по редактированию раздела *история, текущие процессы и план развития компьютерной системы* и их обсуждение администраторами, менеджерами и экспертами.

В соответствии с указанной структурой построена база знаний как самой системы IMS [IMS, 2016], так и всех прикладных систем, построенных по *Технологии OSTIS*.

2. Типология пользователей систем, управляемых знаниями

Процесс разработки *sc-модели базы знаний ostis-системы* по сути сводится к формированию разработчиками предложений по редактированию того или иного раздела базы знаний и последующему рассмотрению этих предложений администраторами, при необходимости - экспертами, а также в отдельных случаях - менеджерами соответствующих проектов.

Рассмотрим типологию пользователей *ostis-системы*.

Все пользователи любой *ostis-системы* делятся на *зарегистрированных пользователей** и *незарегистрированных пользователей**.

*пользователь**

= *пользователь ostis-системы**

∈ *бинарное отношение*

∈ *неролевое отношение*

≤ *разбиение**:

- {
- *незарегистрированный пользователь**
- *зарегистрированный пользователь**
- }

*незарегистрированный пользователь** - это *бинарное отношение*, связывающее *ostis-систему* и *sc-элемент*, обозначающий *персону*, не прошедшую процедуру регистрации в системе.

Незарегистрированный пользователь имеет доступ на чтение *предметной части базы знаний*

ostis-системы. Данный тип пользователей может работать с *ostis-системой* в режиме эксплуатации, т.е. может только задавать запросы, адресуемые *предметной части базы знаний* (т.е. предметные задачи).

*зарегистрированный пользователь** - это *бинарное отношение*, связывающее *ostis-систему* и *sc-элемент*, обозначающий *персону*, прошедшую процедуру регистрации в системе.

Зарегистрированный пользователь имеет доступ на чтение всей базы знаний и внесение предложений ко всей базе знаний, может выполнять роль конечного пользователя *ostis-системы*, т.е. работать в режиме эксплуатации, а также роль ее разработчика. При этом независимо от роли, которую выполняет тот или иной пользователь, он может делать предложения по редактированию любой из частей базы знаний, которые в зависимости от его уровня будут либо приняты автоматически, либо будут отдельно рассматриваться.

Среди зарегистрированных пользователей выделен отдельный тип пользователей - *разработчик**.

*разработчик** - это *бинарное отношение*, связывающее какой-либо *раздел базы знаний ostis-системы* (в пределе - всю базу знаний) и *sc-элемент*, обозначающий *персону*, которая может быть разработчиком данного *раздела базы знаний*, т.е. выполнять проектные задачи в рамках данного раздела.

Помимо эксплуатации *ostis-системы разработчик** может вносить предложения по изменению какой-либо части базы знаний, оставлять комментарии к предложениям по редактированию базы знаний. Среди разработчиков выделены такие роли, как *администратор**, *менеджер** и *эксперт**.

*разработчик**

=> *включение**:

- *администратор**
- *менеджер**
- *эксперт**

*администратор** - это *бинарное отношение*, связывающее какой-либо *раздел базы знаний ostis-системы* (в общем случае - всю базу знаний) и *sc-элемент*, обозначающий *персону*, которая является администратором данного *раздела базы знаний*.

Задачами *администратора** являются:

- контроль целостности и непротиворечивости всей базы знаний;
- определение уровней доступа других пользователей;
- принятие решения относительно принятия или отклонения предложений в различные части базы знаний, в том числе при необходимости отправка их на экспертизу;

• самостоятельное внесение изменений в различные части базы знаний, путем использования соответствующих команд редактирования (при этом изменения автоматически оформляются как предложения и заносятся в *историю развития ostis-системы*).

менеджер* - это бинарное отношение, связывающее какой-либо раздел базы знаний *ostis-системы* (в общем случае – всю базу знаний) и *sc-элемент*, обозначающий *персону*, которая является менеджером данного раздела базы знаний.

Задачами **менеджера*** являются:

- планирование объемов работ по проектированию базы знаний;
- детализация проектных задач на подзадачи, непосредственно формулирование проектных задач, назначение исполнителей проектных задач;
- установка приоритетов и сроков выполнения работ по проектированию базы знаний;
- контроль сроков выполнения проектных задач;

менеджер* вносит изменения в часть соответствующего раздела, описывающую проектные задачи, используя соответствующие команды редактирования (при этом изменения автоматически оформляются как предложения и заносятся в раздел *проект ostis-системы. История, текущие процессы и план развития ostis-системы*). Таким образом, **менеджер*** является **администратором*** раздела *история, текущие процессы и план развития компьютерной системы*.

эксперт* - это бинарное отношение, связывающее какой-либо раздел базы знаний *ostis-системы* (в общем случае – всю базу знаний) и *sc-элемент*, обозначающий *персону*, которая является экспертом данного раздела базы знаний.

Задачами **эксперта*** являются:

- верификация и тестирование результатов выполнения проектных задач;
- при необходимости эксперт может оставлять комментарии к любому фрагменту базы знаний относительно его корректности. Все комментарии попадают в *план развития компьютерной системы*.

3. Типология действий разработчиков баз знаний

В процессе разработки *sc-модели базы знаний ostis-системы* каждый из пользователей, участвующий в разработке использует определенный набор команд. Каждой такой команде соответствует некоторый класс *действий в sc-памяти* [IMS, 2016]. Все такие действия объединены в общий класс **действие разработчика унифицированных семантических моделей баз знаний**.

Рассмотрим типологию таких действий:

действие разработчика унифицированных семантических моделей баз знаний

\leq включение*:

действие в sc-памяти

\Rightarrow включение*:

- *действие эксперта базы знаний*
- *действие администратора базы знаний*
- *действие менеджера базы знаний*
- *действие разработчика базы знаний*

3.1. Действия разработчика базы знаний

действие разработчика базы знаний

\Rightarrow включение*:

- *действие. доработать предложение по редактированию базы знаний*
- *действие. построить новый фрагмент для включения в базу знаний*

действие разработчика базы знаний* может выполняться любым зарегистрированным пользователем* *ostis-системы*, в том числе, при необходимости, ее администратором или менеджером.

действие. сформировать предложение по редактированию базы знаний

\Rightarrow включение*:

- *действие. сформировать предложение проектного задания*
- *действие. сформировать предложение исполнителя проектного задания*

\leq включение*:

действие разработчика базы знаний

Единственным аргументом **действия. сформировать предложение по редактированию базы знаний** является знак *структуры*, описывающей предлагаемое изменение в базе знаний (чаще всего – дополнение к имеющейся базе). Предполагается, что в случае утверждения данного предложения, данная структура будет интегрирована в соответствующий раздел базы знаний, и, при необходимости, будут инициированы *действия*, знаки которых входят в ее состав. Процесс формирования данной *структуры* может быть автоматизирован при помощи соответствующих команд.

Выполнение **действия. сформировать предложение по редактированию базы знаний** предполагает следующие изменения в базе знаний:

- генерируется связка отношения *автор**, связывающая знак *структуры*, обозначающей предложение и знак пользователя, делающего предложение;
- генерируется и инициируется знак *действия. рассмотреть предложение по редактированию базы знаний, первым аргументом' (предложением*)* которого становится знак *структуры*, описывающей предложение по редактированию базы знаний;
- если в базе знаний существует знак какого-либо *действия разработчика базы знаний* (или

более частного), *аргументом* которого является структура, описывающая вносимое предложение, то генерируется *связка отношения последовательность действий**, связывающая знак этого *действия разработчика базы знаний* и знак *действия. рассмотреть предложение по редактированию базы знаний*. Факт наличия такой связи может облегчить задачу администратора по рассмотрению предложения. В противном случае считается, что задача на формирование указанной структуры поставлена не была, и разработчик принял решение о необходимости ее разработки самостоятельно. При этом указанное *действие разработчика базы знаний* становится *прошлой сущностью*.

Среди действий рядовых разработчиков выделен ряд классов действий, которые выполняются только разработчиками, имеющими дополнительные уровни ответственности, т.е. *администратор**, *менеджер** и *эксперт базы знаний**.

действие. утвердить предложение по редактированию базы знаний
= *действие. одобрить предложение по редактированию базы знаний*
<= включение*:

- *действие эксперта базы знаний*
- *действие менеджера базы знаний*
- *действие администратора базы знаний*

Единственным аргументом **действия. утвердить предложение по редактированию базы знаний** является знак действия. рассмотреть предложение по редактированию базы знаний.

Изменения в базе знаний могут быть различными, в зависимости от более частного типа аргумента **действия. утвердить предложение по редактированию базы знаний** и типа пользователя, выполняющего это действие.

Если пользователь является *администратором** проекта соответствующего раздела базы знаний, то:

- генерируется *связка отношения утверждено**, связывающая указанный аргумент и знак пользователя, утвердившего предложение;
- *структура*, описывающая предложение по редактированию базы знаний, интегрируется в базу знаний, при этом иницируются знаки *действий*, входящих в состав этой *структуры*;
- знак этой *структуры* с необходимой спецификацией заносится в раздел, описывающий историю эволюции базы знаний;
- аргумент **действия. утвердить предложение по редактированию базы знаний** становится *прошлой сущностью*.

Если пользователь является менеджером проекта соответствующего раздела базы знаний, и аргумент **действия. утвердить предложение по редактированию базы знаний** является действием. рассмотреть предлагаемого исполнителя проектной задачи то:

- генерируется *связка отношения утверждено**, связывающая указанный аргумент и знак пользователя, утвердившего предложение;
- структура, описывающая предложение по редактированию базы знаний, интегрируется в базу знаний, при этом иницируются знаки действий, входящих в состав этой структуры;
- аргумент **действия. утвердить предложение по редактированию базы знаний** становится *прошлой сущностью*.

Во всех остальных случаях выполнение **действия. утвердить предложение по редактированию базы знаний** считается неправомерным (например, когда пользователь является рядовым разработчиком базы знаний).

действие. отклонить предложение по редактированию базы знаний
<= включение*:

- *действие эксперта базы знаний*
- *действие менеджера базы знаний*
- *действие администратора базы знаний*

Единственным аргументом **действия. отклонить предложение по редактированию базы знаний** является знак действия. рассмотреть предложение по редактированию базы знаний.

В зависимости от более частного типа аргумента **действия. отклонить предложение по редактированию базы знаний** и типа пользователя, выполняющего это действие, изменения в базе знаний могут быть различными.

Если пользователь является администратором проекта соответствующего раздела базы знаний, то:

- генерируется *связка отношения отклонено**, связывающая указанный аргумент и знак пользователя, отклонившего предложение;
- знак этой *структуры* с необходимой спецификацией заносится в раздел, описывающий историю эволюции базы знаний;
- аргумент **действия. отклонить предложение по редактированию базы знаний** становится *прошлой сущностью*.

Если пользователь является менеджером проекта соответствующего раздела базы знаний, и аргумент **действия. отклонить предложение по редактированию базы знаний** является действием. рассмотреть предлагаемого исполнителя проектной задачи то:

- генерируется *связка отношения отклонено**, связывающая указанный аргумент и знак пользователя, отклонившего предложение;
- аргумент **действия. отклонить предложение по редактированию базы знаний** становится *прошлой сущностью*.

Во всех остальных случаях выполнение **действия. отклонить предложение по редактированию базы знаний** считается неправомерным (например, когда пользователь является рядовым разработчиком базы знаний).

3.2. Действия администратора базы знаний

действие администратора базы знаний

=> включение*:

- *действие. рассмотреть предложение по редактированию базы знаний*
=> включение*:
 - *действие. рассмотреть новое проектное задание*
 - *действие. рассмотреть результат верификации предложения*
- *действие. сформировать задание на верификацию предложения*
- *действие. утвердить результат верификации предложения*
- *действие. отклонить результат верификации предложения*

Первым аргументом *действия. сформировать задание на верификацию предложения* является знак *действия. рассмотреть предложение по редактированию базы знаний*. При необходимости может быть указан второй аргумент, являющийся знаком пользователя, который является экспертом, который должен, по мнению администратора, произвести верификацию заданного предложения.

Выполнение *действия. сформировать задание на верификацию предложения* предполагает следующие изменения в базе знаний:

- генерируется и иницируется знак *действия. верифицировать заданную структуру*, в качестве объекта* для данного *действия* указывается знак того предложения, которое должно быть верифицировано;
- если был явно указан эксперт, который должен произвести верификацию, генерируется связка отношения *исполнитель**, связывающая указанный знак *действия. верифицировать заданную структуру* и знак этого эксперта;
- генерируется связка отношения *поддействие**, связывающая знак *действия. рассмотреть предложение по редактированию базы знаний* и указанный знак *действия. верифицировать заданную структуру*.

Единственным аргументом *действия. утвердить результат верификации предложения* является знак *действия. рассмотреть результат верификации предложения*.

Выполнение *действия. утвердить результат верификации предложения* предполагает следующие изменения в базе знаний:

- генерируется и иницируется знак *действия. доработать предложение по редактированию базы знаний, аргументом'* (целью*) которого становится структура, описывающая новую версию предложения, которые было верифицировано экспертом. В качестве инициатора* данного *действия* указывается администратор, от имени которого осуществляется выполнение *действия. утвердить результат верификации предложения*, в качестве исполнителя* -

разработчик предыдущей версии этого предложения по редактированию базы знаний;

- генерируется связка отношения *утверждено**, связывающая аргумент *действия. утвердить результат верификации предложения* и знак, обозначающий администратора, утвердившего результат верификации предложения;
- генерируется связка отношения *последовательность действий**, связывающая аргумент *действия. утвердить результат верификации предложения* и упомянутый знак *действия. доработать предложение по редактированию базы знаний*;

- аргумент *действия. утвердить результат верификации предложения* становится *прошлой сущностью*;

- исходное *действие. рассмотреть предложение по редактированию базы знаний, аргументом'* (целью*) которого была предыдущая версия предложения по редактированию базы знаний также становится *прошлой сущностью*.

Единственным аргументом *действия. отклонить результат верификации предложения* является знак *действия. рассмотреть результат верификации предложения*.

Выполнение *действия. отклонить результат верификации предложения* предполагает следующие изменения в базе знаний:

- генерируется связка отношения *отклонено**, связывающая аргумент *действия. отклонить результат верификации предложения* и знак, обозначающий администратора, отклонившего результат верификации предложения;
- аргумент *действия. отклонить результат верификации предложения* становится *прошлой сущностью*.

3.3. Действия менеджера базы знаний

действие менеджера базы знаний

=> включение*:

- *действие. сформировать предложение проектного задания*
- *действие. рассмотреть предлагаемого исполнителя проектной задачи*

<= включение*:

действие. рассмотреть предложение по редактированию базы знаний

Единственным аргументом *действия. сформировать предложение проектного задания* является знак *структуры*, которую необходимо разработать в процессе выполнения данного проектного задания.

Выполнение *действия. сформировать предложение проектного задания* предполагает следующие изменения в базе знаний:

- генерируется знак *действия. построить новый фрагмент для включения в базу знаний, первым аргументом'* (целью*) которого становится знак *структуры, структуры*, которую необходимо

разработать в процессе выполнения данного проектного задания;

- генерируется *структура*, содержащая весь фрагмент базы знаний, сформированный в предыдущем пункте, далее выполняются те же шаги, как если бы указанная структура была аргументом *действия*. *сформировать предложение по редактированию базы знаний*, с учетом того, что вместо класса *действие*. *рассмотреть предложение по редактированию базы знаний* используется более частный класс *действие*. *рассмотреть новое проектное задание*.

Каждое *действие*. *сформировать предложение исполнителя проектного задания* имеет два аргумента. Первый аргумент' является знаком действия, соответствующего проектному заданию, второй аргумент' – знаком зарегистрированного пользователя*, который предлагается в качестве исполнителя.

Выполнение *действия*. *сформировать предложение исполнителя проектного задания* предполагает следующие изменения в базе знаний:

- генерируется *структура*, описывающая связь *действия*, соответствующего проектному заданию и исполнителя посредством отношения *исполнитель**;

- далее выполняются те же шаги, как если бы данная структура была аргументом *действия*. *сформировать предложение по редактированию базы знаний*, с учетом того, что вместо класса *действие*. *рассмотреть предложение по редактированию базы знаний* используется более частный класс *действие*. *рассмотреть предлагаемого исполнителя проектной задачи*.

3.4. Действия эксперта базы знаний

действие эксперта базы знаний

=> включение*:

- *действие*. *верифицировать заданную структуру*
- *действие*. *отклонить верифицируемое предложение*
- *действие*. *утвердить верифицируемое предложение*
- *действие*. *сформировать задание на рассмотрение результата верификации предложения*

Единственным аргументом *действия*. *утвердить верифицируемое предложение* является знак действия. верифицировать заданную структуру.

Если пользователь является экспертом проекта соответствующего раздела базы знаний, то генерируется связка отношения *утверждено**, связывающая указанный аргумент и знак пользователя, утвердившего предложение, что говорит о том, что указанный эксперт не возражает против включения данного предложения в базу знаний.

Единственным аргументом *действия*.

отклонить верифицируемое предложение является знак действия. верифицировать заданную структуру.

Если пользователь является экспертом проекта соответствующего раздела базы знаний, то генерируется связка отношения *отклонено**, связывающая указанный аргумент и знак пользователя, утвердившего предложение, что говорит о том, что указанный эксперт возражает против включения данного предложения в базу знаний (без дополнительных комментариев).

Единственным аргументом *действия*. *сформировать задание на рассмотрение результата верификации* является знак структуры, описывающей результат верификации экспертом некоторого предложения по редактированию базы знаний. Такая структура обязательно включается в себя:

- знак самого этого предложения, а также знак новой версии данного предложения связанный со знаком исходного предложения связкой отношения *новая версия**;

- комментарий, *ключевым sc-элементом'* которого является знак этой связки, поясняющий, в чем собственно состоит разница между версиями предложения по редактированию базы знаний (что конкретно необходимо доработать).

Выполнение *действия*. *сформировать задание на рассмотрение результата верификации предложения* предполагает следующие изменения в базе знаний:

- генерируется и иницируется знак *действия*. *рассмотреть результат верификации предложения*, *объектом** которого становится *структура*, описывающей результат верификации экспертом соответствующего предложения;

- генерируется связка отношения *последовательность действий**, связывающая знак *действия*. *верифицировать заданную структуру* и упомянутый знак *действия*. *рассмотреть результат верификации предложения*.

4. Средства спецификации предложений по редактированию базы знаний

Рассмотрим ряд отношений, используемых для спецификации *действий разработчиков sc-моделей баз знаний* и *структур*, описывающих предложение по редактированию базы знаний.

*предложение** - это *бинарное отношение*, связывающее знак *действия*. *рассмотреть предложение по редактированию базы знаний* и знак той *структуры*, которая описывает данное предложение по редактированию, например, содержит фрагмент для включения в текущее состояние базы знаний.

*утверждено** - это *бинарное отношение*, связывающее знак *действия* по рассмотрению какого-либо предложения по редактированию базы

знаний и знак того пользователя *ostis-системы*, который утвердил (одобрил) данное *действие*. Как правило, это *администратор** или *эксперт** соответствующего проекта.

*отклонено** - это *бинарное отношение*, связывающее знак *действия* по рассмотрению какого-либо предложения по редактированию базы знаний и знак того пользователя *ostis-системы*, который отклонил данное *действие*. Как правило, это *администратор** или *эксперт** соответствующего проекта.

*новая версия** - это *бинарное отношение*, связывающее знак *структуры*, описывающей предложение по редактированию базы знаний и знак *структуры*, обозначающей версию этого предложения после доработки его исполнителем. Наличие такой связи упрощает и ускоряет повторную проверку предложения по редактированию базы знаний.

Заключение

Рассмотренная в работе модель коллективной разработки баз знаний систем, управляемых знаниями, может быть применена при разработке такого рода систем любого назначения и сложности, поскольку рассчитана на иерархию разработчиков, соответствующую иерархии разделов базы знаний. Представленные в данной работе алгоритмы выполнения действий разработчиков базы знаний рассчитаны на одноуровневую иерархию разработчиков, но могут легко быть обобщены для случая, когда кроме администратора и менеджера всей базы знаний вводятся также роли администраторов и менеджеров ее разделов. В этом случае утверждение какого-либо предложения администратором раздела не приводит к интеграции предложения в соответствующий раздел, и, по сути, является аналогом утверждения предложения экспертом базы знаний или раздела.

Подробный демонстрационный диалог *ostis-системы* с разработчиками ее базы знаний можно найти на сайте системы IMS [IMS, 2016].

Работа выполнена при поддержке гранта БРФФИ-РФФИ-М «Методы и средства онтологического моделирования для семантических технологий проектирования интеллектуальных систем» (Ф15PM-074, 15-57-04047).

Библиографический список

- [Гаврилова и др., 2001] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник / Гаврилова Т.А.. [и др.]; – СПб. : Изд-во «Питер», 2001.
- [Гаврилова, 2008] Гаврилова, Т. А. Визуальные методы работы со знаниями: попытка обзора / Т. А. Гаврилова, Н. А. Гулякина // Искусственный интеллект и принятие решений, 2008, № 1, С. 15-21
- [Голенков, 2012] Голенков, В.В., Гулякина Н.А. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования. – В кн Междунар. научн.-техн. конф. . «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-

2012). Материалы конф. [Минск, 16-18 февр. 2012 г.]. – Минск: БГУИР, 2012.

[Голенков, 2015] Голенков, В.В. Семантическая технология компонентного проектирования систем, управляемых знаниями. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар.научн.-техн.конф./ В. В. Голенков, Н.А Гулякина// Мн.: БГУИР, 2015

[Давыденко, 2013] Давыденко, И.Т. Технология компонентного проектирования баз знаний на основе унифицированных семантических сетей. – В кн Междунар. научн.-техн. конф. . «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2013). Материалы конф. [Минск, 2013 г.]. – Минск: БГУИР, 2013.

[Кудрявцев, 2010] Кудрявцев Д.В. Системы управления знаниями и применение онтологий: Учеб. пособие / Д.В. Кудрявцев. - СПб.: Изд-во Политехн. ун-та, 2010.

[Тузовский и др., 2005] Тузовский А. Ф., Чириков С. В., Ямпольский В. З. (2005) Системы управления знаниями. — Томск : Изд-во науч.-техн. литературы.

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.

[IMS, 2016] Метасистема IMS.OSTIS [Электронный ресурс]. Минск, 2016. – Режим доступа: <http://ims.ostis.net/>. – Дата доступа: 15.01.2016.

THE SEMANTIC MODEL OF KNOWLEDGE BASES COLLECTIVE DESIGN

Davydenko I.T.

*Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus*

ir.davydenko@gmail.com

The paper presents the unified model of the knowledge bases collective design. Describes the unified structure of the system's knowledge base, managed by knowledge. Shows the classification of users systems, managed by knowledge, and typology of the developers actions.

The model of the knowledge base, managed by knowledge, considered in this work, can be applied in the development of such systems for any purpose and complexity, because the hierarchy of developers corresponds to the hierarchy of knowledge base sections.



УДК 004.822:514

SEMANTIC CLASSIFICATION OF ACTIONS FOR KNOWLEDGE INFERENCE

Igor Boyko

Belarus State University, Minsk, Republic of Belarus

igor_m_boyko@hotmail.com

This paper proposes further development of Universal Semantic Code (USC). The notion of an action, represented by a verb, has been considered as a main component for knowledge inference. USC represents actions through semantic strings and operates with semantic axioms to convert the strings to each other. That means the actions may be inferred from each other providing knowledge inference. In the example, semantic inference of actions applied to the text of the patent for revealing knowledge not included in the text and for reproducing full description of the inventive solution claimed in the patent.

Keywords: Universal Semantic Code, verb, action, axiom, knowledge inference, classifier, semantic string, natural language

Introduction

In Natural Language Processing (NLP) numerous approaches of lexical classification exist, but classification of words and classification of meanings of the words are not the same. Regular dictionaries give a definition of meanings but not their classification. It excludes a possibility of knowledge inference (KI) from the sources.

Different linguistic approaches to a hypernym-hyponym classification have been developed. For example, the WordNet classification includes fifteen clusters for verbs and twenty-six for nouns comprising sets of synonyms (synsets) [Fellbaum, 1998]. It is certainly an achievement to define the WordNet classes, but contradiction and incompleteness of the approach does not allow using it for KI. The list of the verb clusters comprises 'Contact verbs' and 'Creation verbs', but not 'Detach verbs' and 'Destruction verbs'. Non-functionality of some cluster names demonstrates inconsistency of the classification. So the cluster name 'Weather verbs' is not comparable with the name 'Motion verbs'.

Another example is the Levin's verb classification [Levin, 1993] which is more consistent because operates with opposite pairs of verb classes, for instance: 'push/pull'. Unfortunately the approach of opposite pairs is incomplete; however it seems reasonable to build the verb classification using opposition as one of the building blocks.

Concerning the Levin's classification M. Palmer wrote [Palmer et al., 1998]: "A primary task of lexical

semantics is to find correct correspondences between the underlying semantic representation of the verb and its alternative syntactic realizations." It would be reasonable to add the syntactic realization should be represented as a set of semantic strings for computer processing.

Traditional knowledge representation (KR) models like frames, semantic net, production rules, first order logic and others operate with formalisms apart of lexical classifications as internal component of the KR model [Harmelen et al., 2008].

The USC classification of actions represented by semantic strings covers the idea of combining NLP and KI in one tool [Martynov, 1992, 2001]. It may seem USC could be considered as a kind of the action language [Gelfond et al., 1998] but that is not true. There is only a terminological overlap in the word 'action'.

Various authors have created a considerable scientific background in the field of NLP used for KI. For example, there are the conceptual dependence model by Schank [Schank, 1975], the model "sense <— > text" by Melchuk [Melchuk, 1974], the generative lexical theory by Pustejovsky [Pustejovsky, 1991], Amarel's analysis about actions [Amarel, 1968]. Unfortunately their approaches do not provide formal representation paired with lexical base for semantic inference but provide strong basis, including philosophical and logical, for the evolution of semantic inference and inter alia for USC development.

1. USC Classes of Actions

USC postulates: knowledge can be kept by means of some internal semantic code and inference of the knowledge from the kept knowledge can be done on the basis of semantic axioms.

To implement any action USC defines four roles: X – subject, Y – instrument, Z – object, W – result. Such roles have shallow similarity with Fillmore’s cases [Fillmore, 1968, 2003].

The USC classification proposes two types of actions: physical and informational. They are mutually correlated. Each class action defines a name of the class. Each action controlling a physical object is in the physical class and each action controlling an informational object is in the informational class.

So, the physical action (PA) “insert” assumes some physical object, which should be inserted. The informational action (IA) “memorize” assumes some informational object, which should be memorized. PA and IA classes are strictly correlated (Fig.1). A complete list of the classes is in the appendices 1 and 2.

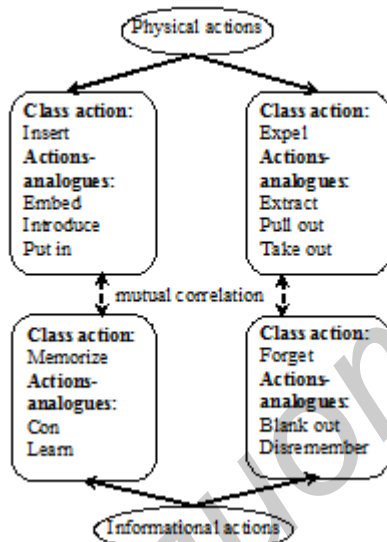


Figure 1-- The USC classifier

Each class action has a corresponded list of actions-analogues and represented by the semantic string. Each string has a natural language interpretation defining roles of the action members.

The action ‘insert’ has a definition ‘put or introduce into something’ and the USC interpretation “X by means of (bmo) Y inserts Z into W”. We can define the members of the action and their roles. For example, for the initial phrase: “A nurse bmo a needle-syringe inserts a vitamin into the blood”:

- X-nurse** is the subject
- Y-needle-syringe** is the instrument
- Z-vitamin** is the first object
- W-blood** is the second object

Each action-analogue of the class ‘insert’ has the same interpretation. So for the action “introduce”, as a

member of the class ‘insert’, the interpretation is: “X bmo Y introduces Z into W”.

The action ‘expel’ has a definition ‘draw or pull out, usually with some force or effort’ and the USC interpretation is: “X bmo Y expels Z from W”:

- X-nurse** is the subject
- Y-needle-syringe** is the instrument
- Z-blood** is the first object
- W-vein** is the second object

The initial phrase is: “A nurse bmo a needle-syringe expels the blood from a vein”.

Each action-analogue of the class “expel” has the same interpretation. So for the action “pull out”, as a member of the class “expel”, the interpretation is: “X bmo Y pulls Z out of W”.

2. Formal Representation of Actions

In USC each action has two parts: **stimulus and reaction**. In physical world the USC notation $((X \rightarrow Y) \rightarrow Z)$ means stimulus with interpretation: X bmo Y affects on Z. In informational world the USC notation $((X \rightarrow Y) \rightarrow X)$ means stimulus with interpretation: X bmo Y affects on X (or on itself).

To define a reaction three conditions should be kept (Martynov, 2001):

1) The first element of the reaction is always a last element of the stimulus: $(Z \rightarrow \dots)$, because some action has happened with the object from the stimulus, for example, $((X \rightarrow Y) \rightarrow Z) \rightarrow ((Z \rightarrow Z) \rightarrow W)$ or shortly $((XY)Z)((ZZ)W)$.

An operation of implication $[\rightarrow]$ demonstrates the direction of the action. Each implication in the string is a directed influence of one variable on another or first part of the string on the second part.

2) Reaction may be ‘active’ or ‘passive’. If reaction is ‘active’ the USC string in the second part is: $((XY)Z)((ZZ)W)$. If reaction is ‘passive’ the USC string in the second part is: $((XY)Z)(Z(ZW))$. It shows changing the position of the parenthesis in the right part of the string.

3) Spatial representation of members of the action.

In USC an operation $[\]$ is a pointer to the position of one object with respect to another in space and considered as a complement of the location.

According to the USC spatial model, all existing objects can have one of three locations: to be in, to be on a surface, to be out of the surface. The notations: W, W’, W’’ mean accordingly ‘inside’, ‘not inside’ that is equal to ‘superficially’, and ‘not superficially’ that is equal to ‘outside’. For example, the actions: ‘insert’ is in, ‘advance’ is on, and ‘target’ is out. Such locations can be easily visualized (Fig.2).

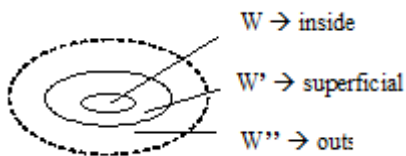


Figure2 -- Location of the elements in the USC model

An experience of Talmy was used here. However, Talmy's basic objective is to identify certain 'conceptual structures' in language that are, in general, parallel to the structuring mechanisms in other cognitive domains such as visual perception [Talmy, 1988].

So far we have considered two opposite physical actions: "insert" and "expel".

Insert – $((XY)Z)((ZZ)W)$ – "a nurse bmoa needle-syringe inserts a vitamin into the blood"

Expel – $((XY)Z)((ZW)Z)$ – "a nurse bmoa needle-syringe expels the blood from a vein"

Actions may be combined forming combined actions consisting of at least two actions; moreover the combined action can consist of two opposite actions. For example, the combined action 'filter' consists of two opposite actions 'insert' and 'expel'. It would be wrong to consider negation "not filter" to the action 'filter' as an opposition. Generally, negation does not mean opposition.

The action 'filter' is represented by the string $((XY)Z)[((ZZ)W)((ZW)Z)]$ where the left part is the stimulus $((XY)Z)$ and the right part is the reaction combined from the left parts of the strings for 'insert' and 'expel' $((ZZ)W)((ZW)Z)$. The interpretation of the string does not differ from the regular interpretations of the physical actions: X bmo Y filters Z of W.

3. USC Axioms

The axioms determine the rules of conversion of one USC string into another. Such conversion is knowledge inference or inference of a consequence of actions.

The consequence of actions cannot be arbitrary but explicitly inferred. Thus the phrase 'A child eats with his hands' will be axiomatically reconstructed as "A child eats with his **mouth, holding food with his hands**". It means the action 'hold' is the preceding action to the action 'eat'. Such reconstruction often is not important for a reader but is important for automatic semantic knowledge inference.

The formal part of the USC algebra has been determined as $\langle \mathbf{M}, \rightarrow, ' \rangle$, where \mathbf{M} is a set of elements, \rightarrow is a binary-non-commutative and non-associative operation on the given set (the operation of implication), $[']$ is a unary operation on the given set (the operation of complement).

3.1. Axiom of Transposition

The axiom defines shifting of internal parenthesis in the right part of the string:

$$((XY)Z)((ZZ)W) \rightarrow ((XY)Z)(Z(ZW)).$$

3.2. Axioms of Diffusion

The axiom defines transferring the variable from one position to another in the right part of the string.

a) Transferring the variable from the first position into the third: $((XY)Z)((ZY)W) \rightarrow ((XY)Z)((ZY)Z)$;

b) Transferring the variable from the second position to the third: $((XY)Z)((ZY)W) \rightarrow ((XY)Z)((ZY)Y)$;

c) Transferring the variable from the first position to the second: $((XY)Z)((ZY)W) \rightarrow ((XY)Z)((ZZ)W)$.

3.3. Axiom of Permutation

The axiom defines simultaneous transferring the variable from the second position into the third and the variable from the third position into the second in the right part of the string: $((XY)Z)((ZW)Y) \rightarrow ((XY)Z)((ZY)W)$.

3.4. Axiom of Substitution

The axiom defines simultaneous substitution of the variables of the second and third positions, in the right part of the string, only if the second and third position variables are equal to each other but not equal to the variable in the first position: $((XY)Z)((ZW)W) \rightarrow ((XY)Z)((ZY)Y)$

3.5. Axiom of Complement

The axiom defines converting one string into another, in the right part of the string, according to the spatial relation: $((XY)Z)((ZZ)W) \rightarrow ((XY)Z)((ZZ)W') \rightarrow ((XY)Z)((ZZ)W'')$.

3.6. Axiom of Internal Relation

The axiom defines relation single and combined strings: $((XY)Z)[((ZZ)W)((ZW)Z)] \rightarrow ((XY)Z)((ZZ)W)$.

4. Semantic Analysis for Engineering Solution

Every invention has been done to achieve some goal. This goal is stated in the invention description. The statement describes the goal function as one action or a set of actions, only one of which is final.

As a rule, a solution of the inventive problem is claimed as a patent for a method, device, or substance. The method is a sequence of the actions united to implement a technical process to achieve the goal. The device is a set of components assembled to implement the method.

Since, the method is a key concept, each action included in the method has semantics determining an order of its application. Only a strictly

limited number of actions precedes the specified action and these actions are not arbitrary.

Since each action is located either before or after the certain actions, the action can be simultaneously of two types: preceding and consequent. The type of the action depends on a point of view: ... → preceding action → current action → consequent action → ...

The example demonstrates how a complete consequence of actions can be extracted from the patent descriptions.

To implement semantic inference of the inventive solution the lexical data base (LDB) was compiled and linked to the USC classifier. The LDB consists of four components:

- Relations between actions represented by verbs and deverbal nouns: 'move' as *moving, movement*; 'connect' as *connecting, connection*. (Leech, Rayson, and Wilson, 2001).
- Relations between actions and change of a parameter: 'cool' as *decrease temperature*; 'accelerate' as *increase speed*.
- Action-analogues according to the action class: Class 'insert'; action-analogues: *embed, enter, introduce, move into*
- Combined class actions: 'spray' as *'move+spread'*; 'freeze on' as *'form+adjoin'*.

To show the example we take: the method of transportation of pulp through the conduit [patent 783154]. The goal of the invention is in reduction of abrasion of the conduit. The goal is achieved by the method of external cooling of walls of the conduit until forming on its internal surface a layer of frozen pulp.

Pulp is moving inside the conduit. An external refrigerant absorbs heat from the walls. The walls cool pulp inside of the conduit and freezing a protective layer on. This layer has the maximum thickness in the lower part of the conduit, which is the part mostly suffering from abrasion.

After freezing the layer of the calculated thickness the cooling device is turning off. The sensor, which signals about the level of abrasion of the protective layer from the frozen pulp, is installed on the conduit. When the layer is abraded to the defined value the cooling device repeatedly is turning on to produce additional freezing of the protective layer on.

The LDB relates the verb 'reduce' with the noun 'reduction'. It seems that the goal of the invention is determined by the action 'reduce'. Checking the USC classifier we find that the word 'reduce' is in the class 'change'. This class depends on the object of influence which in our case is a process of 'abrasion'.

In USC, a physical matter or parameter should be substituted in the position of the variable but not the name of the process. Therefore, the statement 'to reduce abrasion' is not correct since the concept 'abrasion' represents neither the physical object nor the physical parameter, but the process of abrasion. In the LDB, the noun 'abrasion' has a relation to the verb 'abrade' and

we find the action 'abrade' as a member of the class 'damage' in the USC classifier.

We can conclude that a real goal of the invention is in saving the walls of the conduit from the undesirable process of abrasion and should be determined by the action 'save'. This conclusion is supported by the axiom of permutation. Checking the USC classifier the opposite action for 'abrade' is 'save' and for the collocation 'reduce abrasion' the opposite action is 'save' as well.

Compiling together all actions, from the invention description, for achievement of the goal 'save' we receive the following sequence:

... → absorb → cool → freeze on → protect → save.

We can neglect with actions preceding the action 'absorb' because they are not the essential part of the invention and therefore not described in great details. Now we proceed mapping all important actions of the obtained sequence with the USC classifier and verify the members of the actions.

The action 'absorb' is in the PA class 'insert'.

Absorb – ((XY)Z)((ZZ)W) – 'X bmo Y inserts Z into W'

X – subject;
Y – refrigerant;
Z – heat;
W – refrigerant.

The value of the variable X is not specified because it is not important what kind of the device has been used. It is important that the device is using the refrigerant for cooling. Here, the refrigerant is absorbing heat that is why the values of the variables Y and W are the same.

The action 'cool' is in the PA class 'expel'.

Expel – ((XY)Z)((ZZ)W) – 'X bmo Y expels Z from W'

X – subject (device);
Y – refrigerant;
Z – heat;
W – pulp.

It is important to notice that expelling of heat from the object cools the object. That is why 'cool' semantically is an action-analogue for 'expel'. The object of cooling is the pulp and is the value of the variable W.

According to the USC classifier, action 'freeze on' is a combined action consisting of two simultaneous actions 'freeze' and 'adjoin'. The action 'freeze' is in the PA 'form' and the action 'adjoin' is the name of the class.

Freeze – ((XY)Z)(Z(WW)) – 'X bmo Y forms W from Z'

Adjoin – ((XY)Z)((ZY)W') – 'X bmo Y adjoins Z to W'

Freeze on – ((XY)Z)[(Z(WW))((ZY)W')] – 'X bmo Y freezes Z on W'

X – subject;
 Y – refrigerant;
 Z –pulp layer;
 W –internal surface of the conduit.

On that step we have to specify the values of the variables deeper then before. For instance, the variable Z has a value ‘pulp layer’, but not ‘pulp’, and the variable W has a value ‘internal surface of conduit’, but not ‘conduit’. It means we are moving from macro to micro level.

The action ‘protect’ is a name of the class.

Protect – ((XY)Z)((ZW)W”) – ‘X bmo Y protects Z from W’

X –refrigerant;
 Y –frozen pulp layer;
 Z –internal surface of conduit;
 W –liquid pulp.

On this step the variable X has a particular value ‘refrigerant’ as a subject of the action. Implicitly, USC is leading to isolate the operational zone where the undesirable action occurs and the problem should be solved.

The action ‘save’ is a name of the class.

Save – ((XY)Z)(Z(WW”)) – ‘X bmo Y saves Z in W’

X –refrigerant;
 Y –frozen pulp layer;
 Z –internal surface of conduit;
 W –conduit.

It is the final action of the whole process described in the invention. The variable X inherits the value ‘refrigerant’. The internal surface of the conduit is a part of the conduit and it is reasonable to define variables Z and W according to this relation.

Now we can write the solution of the problem in the form of the sequence of the right parts of the USC strings for the correspondent actions:

... →absorb→ cool → freeze on (form+adjoin) → protect → save

... → (ZZ)W → (ZW)Z → (Z(WW))((ZY)W’) → (ZW)W” → Z(WW”)

We expect that the final string can be inferred as sequential converting one string into another according to the USC axioms:

1) According to the axiom of permutation:
 $(ZZ)W \rightarrow (ZW)Z$;

2) No one axiom works for:
 $(ZW)Z \rightarrow (Z(WW))((ZY)W’)$;

3) No one axiom works for:
 $(Z(WW))((ZY)W’) \rightarrow (ZW)W”$;

4) According to the axiom of transposition:
 $(ZW)W” \rightarrow Z(WW”)$.

Looking through axioms we can conclude that our axiomatic inference works partly or the given sequence of the USC strings is incomplete. In the case of

incompleteness we are able to find out missing steps of the inference.

1) The axiom of permutation: $(ZZ)W \rightarrow (ZW)Z$

2) The axiom of diffusion b): $(ZW)Z \rightarrow (ZW)W$

3) The axiom of transposition: $(ZW)W \rightarrow Z(WW)$

4) The axiom of internal relation:

$Z(WW) \rightarrow (Z(WW))((ZY)W’)$

5) The axiom of internal relation:

$(Z(WW))((ZY)W’) \rightarrow (ZY)W’$

6) The axiom of diffusion c): $(ZY)W’ \rightarrow (ZW)W’$

7) The axiom of complement: $(ZW)W’ \rightarrow (ZW)W”$

8) The axiom of transposition: $(ZW)W” \rightarrow Z(WW”)$

Compiling the axiomatic sequence we receive:

... → (ZZ)W → (ZW)Z → (ZW)W → Z(WW) → (Z(WW))((ZY)W’) → (ZY)W’ → (ZW)W’ → (ZW)W” → Z(WW”)

After substitution of the strings with the corresponding actions of the USC classifier:

... →absorb→ cool →produce→form→ freeze on (form+adjoin) →adjoin→restore→ protect → save

So, the complete process has been inferred. To finalize the analysis we consider interpretation of the inferred actions.

The action ‘produce’ is a name of the class.

Produce – ((XY)Z)((ZW)W) – ‘X bmo Y produces W from Z’

X –subject;
 Y – refrigerant;
 Z – liquid pulp;
 W –frozen pulp.

The action ‘form’ is a name of the class.

Form – ((XY)Z)(Z(WW)) – ‘X bmo Y forms W from Z’

X – subject;
 Y – refrigerant;
 Z –frozen pulp;
 W –layer of frozen pulp.

The action ‘adjoin’ is a name of the class.

Adjoin – ((XY)Z)((ZY)W’) – ‘X bmo Y adjoins Z and W’

X – subject;
 Y – refrigerant;
 Z –layer of frozen pulp;
 W –internal surface of conduit.

The action ‘restore’ is a name of the class.

Restore – ((XY)Z)((ZW)W’) – ‘X bmo Y restores W from Z’

X –subject;
 Y –refrigerant;
 Z –abraded internal surface of conduit;
 W –internal surface of conduit.

To simplify the final representation we can exclude from the axiomatic sequence the strings $Z(WW)$ and $(ZY)W'$.

Now the final sequence is:

... $\rightarrow (ZZ)W \rightarrow (ZW)Z \rightarrow (ZW)W \rightarrow (Z(WW))(ZY)W' \rightarrow (ZW)W' \rightarrow (ZW)W'' \rightarrow Z(WW'')$
 ... \rightarrow absorb \rightarrow cool \rightarrow produce \rightarrow freeze on (form+ adjoin) \rightarrow restore \rightarrow protect \rightarrow save

The example is demonstrating how implicit knowledge is becoming explicit. Of course, it is necessary to use human intervention for defining the values of the variables but inference of the consequence of the USC strings is automatic. For a natural language it can be compared with revealing ellipses in the sentence.

Conclusion

Initially, the approach was applied to infer only knowledge not included in the description of the inventive solution. But the approach can be applied for inventive problem solving. The inventive solution can be generated starting from the statement of the goal and moving back to each previous action for compiling a chain of the actions related through USC axioms [Boyko, 2001], [Kandelinski et al., 2014]. The number of possible solutions depends on the number of chosen axiomatic passes.

We would like to underline that USC unites several components including: definitions of the actions, its formal representation, natural language interpretation and axioms of inference. The latest version of the USC classifier has 108 classes: 54 physical and 54 informational classes. The whole set of actions comprises 5200 entities [Boyko, 2006]. Most of the combined actions comprise two components but there are few three and four component actions. For example, the action 'cut off' consists of three simultaneous class actions 'touch+move+separate'.

Formal representation of actions as an intermediate code in "human-computer" interface is the essential property of USC. The USC strings have been used to represent not only actions, but also deverbal nouns and adjectives for development of the universal principles of text processing [Boyko, 2002].

However, we do not consider the USC model as a completed model. There are problems that should be developed, including: verification of combined actions, automatic substitution of members of the action, parallel inference of the consequences of the actions, evaluation of the quality of the inferred consequence.

Formal semantic coding for knowledge inference is a key component for KI. Majority of experiments in corpus-based natural language processing present results for some subtasks and there are few results that can be successfully integrated to build a complete NLP system with KI ability.

USC is the growing approach that can become a part of the class conceptual and computational framework

forming the foundation of effective scalable natural language systems capable to knowledge inference.

Acknowledgements

The author would like to thank the creator of USC Victor V. Martynov who encouraged further developing of the theory.

References

- [Amarel, 1968] Amarel, S. *On representations of problems of reasoning about actions*. In Michie, D., ed., *Machine Intelligence*, volume 3. Edinburgh: Edinburgh University Press. 1968.
- [Boyko, 2001] Boyko, I. *Computer Semantic Search of Inventive Solutions*. TRIZ Journal. USA. March. <http://www.triz-journal.com/archives/2001/03/d/>.
- [Boyko, 2002] Boyko I. *Terminological Abstractions for Terminology Classification*. 6th International Conference Terminology and Knowledge Engineering. Nancy, France. <http://www.sempl.net.2002>.
- [Boyko, 2009 edition] Boyko, I. *Formal Semantics Of Verbs For Knowledge Inference*. // Fifth Workshop On Inference in Computational Semantics (ICoS-5), 20 - 21 April 2006, Buxton, England. (2009 Edition) <http://unsemcode.com>.
- [Fellbaum, 1998] Fellbaum C. (editor). *WordNet An Electronic Lexical Database*. The MIT press. 1998. <http://citeseer.nj.nec.com/lin98wordnet.html>
- [Fillmore, 1968] Fillmore, Charles J. *The case for case*. In Bach and Harms (Ed.): *Universals in Linguistic Theory*. New York: Holt, Rinehart, and Winston. 1968.
- [Fillmore, Johnson, Petruck, 2003] Fillmore, Charles J., Christopher R. Johnson and Miriam R.L. Petruck. *Background to Framenet*. *International Journal of Lexicography*, Vol 16.3.2003.
- [Gelfond, Lifschitz, 1998] Gelfond, M., and Lifschitz, V. *Action languages*. *Electronic Transactions on Artificial Intelligence* 3.1998.
- [Harmelen, Lifschitz, Porter, 2008] Harmelen, F., Lifschitz, V., Porter, B. *Handbook of Knowledge Representation*, Elsevier. 2008.
- [Kandelinski ,Boyko, 2014] Kandelinski S, Boyko I. *Universal Semantic Code as an Abstracting Technology for Fuzzy Engineering Problem Solving*. OSTIS-2014 Conference. Minsk. 2014.
- [Leech, Rayson, Wilson, 2001] Leech G., Rayson P., and Wilson A. *Word Frequencies in Written and Spoken English*. Pearson Education. 2001.
- [Levin, 1993] Levin B. *English Verbs Classes and Alternations, A Preliminary investigation*. The University of Chicago Press. 1993.
- [Martynov, Boyko, Guminsky, 1992] Martynov V., Boyko I., Guminsky A. *Knowledge Bases Construction of Systems for Solving Intellectual Problems*. *Controlling Systems and Machines Journal*. Kiev. <http://www.sempl.net>. 1992.
- [Martynov, 2001] Martynov V. *Foundations of semantic coding. Summary*. European Humanity University. Minsk. <http://www.sempl.net>. 2001.
- [Minsky, 1975] Minsky M. *Framework for Representation of Knowledge*. The Psychology of Computer Vision. New-York. 1975.
- [Melchuk, 1974] Melchuk I. *Experience of Theory of Linguistic Models "Sense \leftrightarrow Text"*. Moscow. 1974.
- [Palmer, Rosenzweig, Schuler, 1998] Palmer M., Rosenzweig J., Schuler W. *Capturing Motion Verb Generalizations with Synchronous actionGs*. In *Predicative Forms in NLP*, ed by Dizier P.S. Kluwer Press. 1998.
- [Pustejovsky, 1991] Pustejovsky J. *The Generative Lexicon*. *Computational Linguistics*, 17(4). 1991.
- [Schank, 1975] Schank R. *Conceptual Information Processing*. Amsterdam-Oxford-New-York. 1975.
- [Talmy, 1988] Talmy L., *Force dynamics in language and cognition*. *Cognitive Science*, 12 (1). 1988.
- [Patent 783154] Patent 783154 USSR, MKI B 65/ G 53/00. Bykonurov, L., Krupnik, A., Ombaraev, O., Sobolev, I. *A Method of Transportation of Pulp through a Pipeline*. Bul. N44. (in Russian)

СЕМАНТИЧЕСКАЯ КЛАССИФИКАЦИЯ ДЕЙСТВИЙ ДЛЯ ВЫВОДА ЗНАНИЙ

Бойко И.М.

Белорусский Государственный
Университет, Минск, Республика Беларусь

igor_m_boyko@hotmail.com

Статья показывает дальнейшее развитие Универсального Семантического Кода, как инструмента для вывода знаний.

Основная часть

Основным элементом вывода является понятие 'действие', выраженное в естественном языке глаголом. Семантические цепочки УСК формально представляют понятие 'действие', что даёт возможность вывода их друг из друга. Для этого используются УСК аксиомы.

УСК генерирует пропущенные действия и предлагает возможности определить членов этих действий. Таким образом формируется фрагмент модели мира, как компонент для построения полной модели мира для вывода знаний.

Основными элементами УСК являются: семантический классификатор действий, семантические цепочки, формально представляющие действия, правила канонизированного чтения семантических цепочек, аксиоматика вывода одних семантических цепочек их других.

Пример вывода показан с использованием описания патента изобретения на метод. В описании пропущены некоторые шаги за счёт эллиптичности естественного языка. Эти шаги выводятся средствами УСК и показываются в явном виде.

Заключение

Только полное представление текста может стать основой системы автоматического вывода знаний безотносительно к предметной области.

Appendix 1

Classes of physical actions					
1.1	(ZY)W	Connect - make joined or united <i>X bmo Y connects Z and W</i>	2.1	(ZW)Y	Disconnect - make disconnected, disjoined <i>X bmo Y disconnects Z and W</i>
1.2	Z(YW)	Fasten - cause to be firmly attached <i>X bmo Y fastens Z and W</i>	2.2	Z(WY)	Unfasten - cause to be not firmly attached <i>X bmo Y unfastens Z and W</i>
1.3	(ZY)W'	Adjoin - make contact <i>X bmo Y adjoins Z and W</i>	2.3	(ZW)Y'	Separate - make a division or separation <i>X bmo Y separates Z and W</i>
1.4	Z(YW')	Touch - be in direct physical contact with <i>X bmo Y touches Z and W</i>	2.4	Z(WY')	Detach - come to be detached <i>X bmo Y detaches Z and W</i>
1.5	(ZY)W''	Oppose - place in close proximity <i>X bmo Y opposes Z to W</i>	2.5	(ZW)Y''	Distance - place in far from each other <i>X bmo Y distances Z and W</i>
1.6	Z(YW'')	Neighbor - be located near <i>X bmo Y matches Z and W</i>	2.6	Z(WY'')	Isolate - set apart <i>X bmo Y isolates Z and W</i>
3.1	(ZZ)W	Insert - put or introduce into something <i>X bmo Y inserts Z into W</i>	4.1	(ZW)Z	Expel - force to leave or move out <i>X bmo Y expels Z from W</i>
3.2	Z(ZW)	Fill - occupy the whole <i>X bmo Y fills Z with W</i>	4.2	Z(WZ)	Empty - became empty or void of its content <i>X bmo Y empties Z of W</i>
3.3	(ZZ)W'	Advance - move forward <i>X bmo Y advances Z to W</i>	4.3	(ZW)Z'	Remove - remove from a close position <i>X bmo Y removes Z from W</i>
3.4	Z(ZW')	Approach - move toward something <i>X bmo Y approaches Z to W</i>	4.4	Z(WZ')	Draw back - pull back or move away <i>X bmo Y draws back Z from W</i>
3.5	(ZZ)W''	Target - intend to move towards a certain goal <i>X bmo Y targets Z to W</i>	4.5	(ZW)Z''	Deflect - turn from a straight course or fixed direction <i>X bmo Y deflects Z from W</i>

3.6	Z(ZW'')	Line up - place in a line or arrange so as to be parallel <i>X bmo Y lines up Z and W</i>	4.6	Z(WZ'')	Angle - move or proceed at an angle <i>X bmo Y angles Z and W</i>
5.1	(ZY)Z	Contract - squeeze or press together <i>X bmo Y contracts Z</i>	6.1	(ZZ)Y	Expand - make bigger or wider in size, volume, or quantity <i>X bmo Y expands Z</i>
5.2	Z(YZ)	Compact - make more compact <i>X bmo Y compacts Z</i>	6.2	Z(ZY)	Widen - become broader or wider <i>X bmo Y widens Z</i>
5.3	(ZY)Z'	Concentrate - draw together in one common center <i>X bmo Y concentrates Z</i>	6.3	(ZZ)Y'	Spread - distribute over an area <i>X bmo Y spreads Z</i>
5.4	Z(YZ')	Gather - collect in one place <i>X bmo Y gathers Z</i>	6.4	Z(ZY')	Disperse - move away from each other <i>X bmo Y disperses Z</i>
5.5	(ZY)Z''	Hold - keep in a certain state, position <i>X bmo Y holds Z</i>	6.5	(ZZ)Y''	Release - free from hold <i>X bmo Y releases Z</i>
5.6	Z(YZ'')	Stay - remain in a certain state <i>X bmo Y stays Z</i>	6.6	Z(ZY'')	Leave - go away from a place <i>X bmo Y leaves Z</i>
7.1	(ZY)Y	Destroy - damage irreparably <i>X bmo Y destroys Z</i>	8.1	(ZW)W	Produce - make by combining materials and parts <i>X bmo Y produces W from Z</i>
7.2	Z(YY)	Deform - make formless <i>X bmo Y deforms Z</i>	8.2	Z(WW)	Form - give shape or form <i>X bmo Y forms W from Z</i>
7.3	(ZY)Y'	Break - destroy the integrity <i>X bmo Y breaks Z</i>	8.3	(ZW)W'	Restore - return to its original or usable condition <i>X bmo Y restores W from Z</i>
7.4	Z(YY')	Fracture - become fractured <i>X bmo Y fractures Z</i>	8.4	Z(WW')	Preserve - keep or maintain in unaltered condition <i>X bmo Y preserves W in Z</i>
7.5	(ZY)Y''	Attack - begin to injure <i>X bmo Y attacks Z</i>	8.5	(ZW)W''	Protect - shield from danger, destruction, or damage <i>X bmo Y protects W from Z</i>
7.6	Z(YY'')	Damage - cause or do harm <i>X bmo Y damages Z</i>	8.6	Z(WW'')	Save - save from ruin, destruction, or harm <i>X bmo Y saves W in Z</i>
9.1	(ZZ)Z	Change - cause a physical transformation <i>X bmo Y changes Z</i>			
9.2	Z(ZZ)	Transform - change or alter in a form, appearance, or nature <i>X bmo Y transforms Z</i>			
9.3	(ZZ)Z'	Move - perform a motion <i>X bmo Y moves Z</i>			
9.4	Z(ZZ')	Displace - put out of its place, position <i>X bmo Y displaces Z</i>			
9.5	(ZZ)Z''	Handle - cause to function <i>X bmo Y handles Z</i>			
9.6	Z(ZZ'')	Operate - perform as expected <i>X bmo Y operates Z</i>			

Appendix 2

Classes of informational actions					
1.1	(XY)W	Teach - impart skills or knowledge <i>X bmo Y teaches Z</i>	2.1	(XW)Y	Confuse - make unclear or incomprehensible <i>X bmo Y confuses Z</i>
1.2	X(YW)	Understand - comprehend the nature or meaning <i>X bmo Y understands Z</i>	2.2	X(WY)	Misunderstand - interpret in the wrong way <i>X bmo Y misunderstands Z</i>

1.3	(XY)W'	Inform - impart knowledge of some fact <i>X bmo Y informs Z</i>	2.3	(XW)Y'	Misinform - give false or misleading information <i>X bmo Y misinforms Z</i>
1.4	X(YW')	Know - be aware of information <i>X bmo Y knows Z</i>	2.4	X(WY')	Be unaware - be unaware of information <i>X bmo Y is unaware of Z</i>
1.5	(XY)W''	Disclose - make known publicly <i>X bmo Y discloses W</i>	2.5	(XW)Y''	Conceal - prevent from being known <i>X bmo Y conceals W</i>
1.6	X(YW'')	Follow - keep informed <i>X bmo Y follows W</i>	2.6	X(WY'')	Miss - fail to reach or get <i>X bmo Y misses W</i>
3.1	(XX)W	Memorize - commit to memory <i>X memorizes W</i>	4.1	(XW)X	Forget - dismiss from the mind <i>X forgets W</i>
3.2	X(XW)	Remember - keep in mind <i>X remembers W</i>	4.2	X(WX)	Discard - throw or cast away <i>X discards W</i>
3.3	(XX)W'	Associate - make a logical or causal connection <i>X associates W</i>	4.3	(XW)X'	Dissociate - cease or break association <i>X dissociates W</i>
3.4	X(XW')	Deduce - conclude by reasoning <i>X deduces W</i>	4.4	X(WX')	Speculate - to believe on uncertain grounds <i>X speculates W</i>
3.5	(XX)W''	Perceive - to become aware of through the senses <i>X perceives W</i>	4.5	(XW)X''	Lose - fail to perceive with the senses or the mind <i>X loses W</i>
3.6	X(XW'')	Consider - take into consideration <i>X considers W</i>	4.6	X(WX'')	Neglect - give little or no attention <i>X neglects W</i>
5.1	(XY)X	Encode - convert information into code <i>X bmo Y encodes W</i>	6.1	(XX)Y	Decode - convert code into information <i>X bmo Y decodes W</i>
5.2	X(YX)	Compress - make more compact <i>X bmo Y compresses W</i>	6.2	X(XY)	Decompress - restore to its uncompressed form <i>X bmo Y decompresses W</i>
5.3	(XY)X'	Summarize - give a summary <i>X bmo Y summarizes W</i>	6.3	(XX)Y'	Elaborate - add details <i>X bmo Y details W</i>
5.4	X(YX')	Abstract - consider a concept without thinking of a specific example <i>X bmo Y abstracts W</i>	6.4	X(XY')	Concretize - become specific <i>X bmo Y concretizes W</i>
5.5	(XY)X''	Focus - bring into focus or alignment <i>X bmo Y focuses W</i>	6.5	(XX)Y''	Distract - draw attention away <i>X bmo Y distracts W</i>
5.6	X(YX'')	Interest - engage the interest <i>X bmo Y interests W</i>	6.6	X(XY'')	Bore - cause to be born <i>X bmo Y bores W</i>
7.1	(XY)Y	Terminate - bring to an end or halt <i>X terminates Y</i>	8.1	(XW)W	Invent - come up with an idea, explanation, theory <i>X bmo Y invents W</i>
7.2	X(YY)	Expire - lose validity <i>X expires Y</i>	8.2	X(WW)	Innovate - bring something new to an environment <i>X bmo Y innovates W</i>
7.3	(XY)Y'	Disregard - give little or no attention to <i>X disregards Y</i>	8.3	(XW)W'	Research - attempt to find out in a systematically manner <i>X bmo Y researches W</i>
7.4	X(YY')	Err - to make a mistake <i>X errs in Y</i>	8.4	X(WW')	Solve - find the solution to a problem or question <i>X bmo Y solves W</i>

7.5	(XY)Y''	Disorder - bring disorder <i>X disorders Y</i>	8.5	(XW)W''	Order - bring order <i>X bmo Y orders W</i>
7.6	X(YY'')	Disorganize - destroy systematic arrangement <i>X disorganizes Y</i>	8.6	X(WW'')	Systematize - arrange according to a system <i>X bmo Y systematizes W</i>
9.1	(XX)X	Think - use the mind in order to make inferences, decisions <i>X thinks Y</i>			
9.2	X(XX)	Define - give a definition for the meaning <i>X defines Y</i>			
9.3	(XX)X'	Calculate - make a mathematical calculation or computation <i>X calculates Y</i>			
9.4	X(XX')	Determine - establish after a calculation <i>X determines Y</i>			
9.5	(XX)X''	Evaluate - place a value on <i>X evaluates Y</i>			
9.6	X(XX'')	Compare - examine and note the similarities or differences <i>X compares Y</i>			



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ФОРМАЛЬНОЕ СЕМАНТИЧЕСКОЕ ОПИСАНИЕ ЦЕЛЕНАПРАВЛЕННОЙ ДЕЯТЕЛЬНОСТИ РАЗЛИЧНОГО ВИДА СУБЪЕКТОВ

Шункевич Д.В. *, Губаревич А.В. *, Святкина М.Н. **, Моросин О.Л. ***

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

shunkevichdv@gmail.com

st.hubarevich@gmail.com

*** Московский государственный технический университет им. Н.Э. Баумана,
г. Москва, Россия*

maria.svyatkina@gmail.com

**** Национальный Исследовательский Университет «МЭИ»,
г. Москва, Россия*

oleg@morosin.ru

В работе рассматриваются формальные модели, позволяющие описывать целенаправленную деятельность субъектов различного рода с использованием семантических сетей с теоретико-множественной интерпретацией. В частности, рассматривается типология действий, понятие задачи как спецификации действия, отношения, используемые для спецификации действий. Также рассматриваются другие классы спецификаций действия, такие как, план, программа, решение и другие.

Ключевые слова: семантические технологии, формализация деятельности, многоагентные системы, проект OSTIS.

Введение

Целью данной работы является разработка базовых принципов описания целенаправленной деятельности различного вида субъектов на основе семантических сетей с базовой теоретико-множественной интерпретацией. Разработанные в данной работе формальные модели и средства используются при разработке различных систем, управляемых знаниями, построенных на основе *Технологии OSTIS* [Голенков, 2015]. Отметим, что разработанные модели и средства применимы для описания в семантической памяти не только деятельности субъектов в рамках самой этой памяти, но и описания деятельности субъектов, внешних по отношению к *ostis-системе* [Голенков, 2015], как в данный момент времени, так и в прошлом и будущем.

Семантическая теория деятельности подробно рассматривается в работах В.В. Мартынова [Мартынов, 1974], [Мартынов, 1977], [Мартынов, 1984], а также его учеников, в частности, в работе

[Воуко, 2016]. В указанных работах подробно рассматривается семантическая типология действий, описываются подходы к формализации описания деятельности различного рода субъектов с использованием разработанного В.В. Мартыновым Универсального семантического кода (УСК). В этих работах детально рассматриваются такие понятия, как воздействие, субъект воздействия (агент), объект и другие, рассматриваемые также и в данной работе.

Кроме того, в работе [Шенк, 1980] рассматриваются такие понятия, близкие рассматриваемым в данной, как воздействие, действие (акт), деятель (субъект) и другие.

Целью данной работы, в отличие от указанных, является создание формальных моделей и средств описания целенаправленной деятельности различного вида субъектов в системах, управляемых знаниями, на основе формальных моделей представления знаний, используемых в Технологии OSTIS [Голенков, 2012], что, однако, никак не

противоречит исследованиям, описанным в указанных работах.

Данная работа является частью базы знаний системы IMS OSTIS [IMS, 2016], в рамках которой она представлена в виде раздела, описывающего соответствующую предметную область [Давыденко, 2016b].

1. Базовая типология действий

1.1. Понятие действия и типология действий

В рамках предлагаемой модели понятие *действие* является частным случаем понятия *воздействие*, которое рассматривается в более общих предметных областях в рамках базы знаний IMS [IMS, 2016].

Каждому *воздействию* может быть поставлена в соответствие (1) некоторая *воздействующая сущность**, т.е. сущность, осуществляющая *воздействие* (в частности, это может быть некоторое физическое поле), и (2) некоторый *объект воздействия**, т.е. сущность, на которую воздействие направлено. Если *воздействие* связано с *материальной сущностью*, то его объектом воздействия является либо сама эта *материальная сущность*, либо некоторая ее пространственная часть.

В свою очередь, каждое *действие*, выполняемое тем или иным *субъектом*, одновременно можно трактовать и как процесс решения некоторой задачи, т.е. как процесс достижения заданной цели в заданных условиях.

Предполагается, что любое *действие*, выполняемое каким-либо *субъектом*, направлено на решение какой-либо задачи и выполняется *целенаправленно*. При этом явное указание *действия* и его связи с конкретной *задачей* может не всегда присутствовать в памяти. Некоторые задачи могут решаться определенными агентами перманентно, например, оптимизация базы знаний, поиск некорректностей и т.д., и для подобных задач не всегда есть необходимость явно вводить *структуру*, являющуюся формулировкой *задачи*.

Каждое *действие* может обозначать сколь угодно малое преобразование, осуществляемое во внешней среде либо в памяти некоторой системы, однако в памяти явно вводятся только знаки тех *действий*, для которых есть необходимость явно хранить в памяти их спецификацию в течение некоторого времени.

При выполнении действия можно выделить следующие этапы:

- построение плана деятельности; декомпозиция (детализация) исходного действия;
- выполнение построенного плана действий;

Формальная спецификация понятия *действие* в SCn-коде (в том числе – классификация действий):

действие

= *акция*
= *сделать*
= *работа*
= *процесс выполнения некоторой работы*
= *процесс решения некоторой задачи*
= *процесс достижения некоторой цели*
= *дело*
= *мероприятие*
= *воздействие*
= *целостный фрагмент некоторой деятельности*
= *целенаправленный процесс, управляемый некоторым субъектом*
= *процесс выполнения некоторого действия некоторым субъектом (исполнителем) над некоторыми объектами*
<= *разбиение**:
Разбиение класса действий по отношению к памяти информационной системы
=
{
• *информационное действие*
=> *включение**:
 действие в sc-памяти
• *поведенческое действие*
=> *включение**:
 действие во внешней среде ostis-системы
• *эффекторное действие*
=> *включение**:
 эффекторное действие ostis-системы
• *рецепторное действие*
=> *включение**:
 рецепторное действие ostis-системы
}
<= *разбиение**:
Разбиение класса действий по отношению к текущему моменту времени
=
{
• *иницированное действие*
• *планируемое действие*
= *будущее действие*
• *выполненное действие*
}

1.2. Типы действий по отношению к памяти информационной системы

Результатом выполнения *информационного действия* является в общем случае некоторое новое состояние памяти информационной системы (не обязательно *sc-памяти*), достигнутое исключительно путем преобразования информации, хранящейся в памяти системы, то есть либо посредством генерации новых знаний на основе уже имеющихся, либо посредством удаления знаний, по каким-либо причинам ставших ненужными. Следует отметить, что если речь идет об изменении состояния *sc-памяти*, то любое преобразование информации можно свести к ряду элементарных действий генерации, удаления или изменения

инцидентности *sc-элементов* друг относительно друга.

В случае *поведенческого действия* результатом его выполнения будет новое состояние внешней среды. Очень важно отметить, что под внешней средой в данном случае понимаются также и компоненты системы, внешние с точки зрения памяти, то есть не являющиеся хранимыми в ней информационными конструкциями. К таким компонентам можно отнести, например, различные манипуляторы и прочие средства воздействия системы на внешний мир, то есть к поведенческим задачам можно отнести изменение состояния механической конечности робота или непосредственно вывод некоторой информации на экран для восприятия пользователем.

Под *эффекторным действием* понимается действие, связанное с воздействием *ostis-системы* на внешнюю с точки зрения системы среду. Под *рецепторным действием* понимается действие, связанное с воздействием внешнего субъекта на *ostis-систему*, т.е. на ее датчики, рецепторы и т.п.

1.3. Типы действий по отношению к текущему моменту времени

Во множество *иницированных действий* входят *действия*, выполнение которых инициировано в результате какого-либо события.

В общем случае, *действия* могут быть инициированы по следующим причинам:

- *действие* инициировано явно путем проведения соответствующей *sc-дуги принадлежности* каким-либо *субъектом* (*заказчиком**). В случае *действия в sc-памяти*, оно может быть инициировано как внутренним *sc-агентом* системы, так и пользователем при помощи соответствующего пользовательского интерфейса. При этом, спецификация действия может быть сформирована одним *sc-агентом* [Шункевич, 2014], [Тарасов, 2002], а собственно добавление во множество *иницированных действий* может быть осуществлено позже другим *sc-агентом*.

- *действие* инициировано в результате того, что одно или несколько действий, предшествовавших данному в рамках некоторой декомпозиции, стали *прошлыми сущностями* (процедурный подход).

- действие инициировано в результате того, что в памяти системы появилась конструкция, соответствующая некоторому условию инициирования *sc-агента*, который должен выполнить данное *действие* (декларативный подход)

Следует отметить, что декларативный и процедурный подходы можно рассматривать как две крайности, использование только одной из которых не является удобным и целесообразным. При этом, например, принципы инициирования по процедурному подходу могут быть полностью сведены к набору декларативных условий

иницирования, но как было сказано, это не всегда удобно и наиболее рациональным будет комбинировать оба подхода в зависимости от ситуации.

По сути, попадание некоторого *действия* во множество *иницированных действий* говорит о том, что спецификация данного *действия*, полностью сформирована, т.е. никаких дополнительных элементов, необходимых для решения поставленной задачи, не требуется, и соответствующий *sc-агент* (либо коллектив *sc-агентов*, либо внешний *субъект*) может приступить к выполнению действия. Однако стоит отметить, что с точки зрения исполнителя такая спецификация *действия* в общем случае может оказаться недостаточной или некорректной.

иницированное действие

=> включение*:

выполняемое действие

= *настоящее действие*

<= разбиение*:

{

- *активное действие*

- *отложенное действие*

}

Во множество *выполняемых действий* входят *действия*, к выполнению которых приступил какой-либо из соответствующих *субъектов*.

Попадание *действия* в данное множество говорит о следующем:

- рассматриваемое *действие* уже попало во множество *иницированных действий*.

- существует как минимум один *субъект*, условие инициирования которого соответствует спецификации данного *действия*.

После того, как собственно процесс выполнения завершился, *действие* должно быть удалено из множества *выполняемых действий* и добавлено во множество *выполненных действий* или какое-либо из его подмножеств.

Понятие *выполняемое действие* является неосновным, и вместо того, чтобы относить конкретные *действия* к данному классу, их относят к классу *настоящих сущностей*.

Во множество *активных действий* входят *действия*, выполнение которых осуществляется непосредственно в данный момент каким-либо *субъектом*.

Во множество *отложенных действий* входят *действия*, которые уже были инициированы, однако их выполнение невозможно по каким-либо причинам, например в случае, когда у исполнителя в данный момент есть более приоритетные задачи.

Во множество *планируемых действий* входят *действия*, начать выполнение которых запланировано на какой-либо момент в будущем.

Во множество **выполненных действий** попадают *действия*, выполнение которых с точки зрения завершено с точки зрения *субъекта*, осуществлявшего их выполнение. В зависимости от результатов конкретного процесса выполнения, рассматриваемое *действие* может стать элементом одного из подмножеств множества **выполненных действий**.

Понятие **выполненное действие** является неосновным, и вместо того, чтобы относить конкретные *действия* к данному классу, их относят к классу *прошлых сущностей*.

Во множество **успешно выполненных действий** попадают *действия*, выполнение которых успешно завершено с точки зрения *субъекта*, осуществлявшего их выполнение, т.е. достигнута поставленная цель, например, получены решение и ответ какой-либо задачи, успешно преобразована какая-либо конструкция и т.д.

Если действие было выполнено успешно, то, в случае действия по генерации каких-либо знаний, к *действию* при помощи связки отношения *результат** приписывается *sc-конструкция*, описывающая результат выполнения указанного действия. В случае, когда *действие* направлено на какие-либо изменения базы знаний, *sc-конструкция*, описывающая результат действия, формируется в соответствии с правилами описания истории изменений базы знаний.

В случае, когда успешное выполнение *действия* приводит к изменению какой-либо конструкции в *sc-памяти*, которое необходимо занести в историю изменений базы знаний или использовать для демонстрации протокола решения задачи, то генерируется соответствующая связка отношения *результат**, связывающая *задачу* и *sc-конструкцию*, описывающую данное изменение.

выполненное действие

= *прошлое действие*

<= *разбиение**:

- {
- *успешно выполненное действие*
- *безуспешно выполненное действие*
- }

Во множество **безуспешно выполненных действий** попадают *действия*, выполнение которых не было успешно завершено с точки зрения *субъекта*, осуществлявшего их выполнение, по каким-либо причинам.

Можно выделить две основные причины, по которым может сложиться указанная ситуация:

- соответствующая *задача* сформулирована некорректно;
- формулировка соответствующей *задачи* корректна и понятна системе, однако решение данной задачи в текущий момент не может быть получено за удовлетворительные с точки зрения заказчика или исполнителя сроки.

Для конкретизации факта некорректности формулировки задачи можно выделить ряд более частных классов **безуспешно выполненных действий**, например:

- *действие*, спецификация которого противоречит другим знаниям системы (например, не выполняется неравенство треугольника);
- *действие*, при спецификации которого использованы понятия, неизвестные системе;
- *действие*, выполнение которого невозможно из-за недостаточности данных (например, найти площадь треугольника по двум сторонам);
- и другие

Для конкретизации факта безуспешности выполнения некоторого *действия* в системе могут также использоваться дополнительные подмножества данного множества, при необходимости снабженные естественно-языковыми комментариями.

Для указания приоритетности выполнения тех или иных действий дополнительно вводится понятие **приоритет действия**.

приоритет действия

≡ *действие с очень высоким приоритетом'*

≡ *действие с высоким приоритетом'*

≡ *действие со средним приоритетом'*

≡ *действие с низким приоритетом'*

≡ *действие с очень низким приоритетом'*

Множество **приоритет действия** представляет собой семейство ролевых отношений, элементами которых являются *sc-дуги принадлежности*, связывающие множество поддействий в рамках декомпозиции некоторого более сложного *действия* и сами эти поддействия. Таким образом, данные ролевые отношения задают приоритетность выполнения более частных поддействий при выполнении некоторого общего действия. Приоритетность выполнения влияет на *действия*, независимые с точки зрения *последовательности действий**, и отражает влияние каждого более частного действия на качество результата выполнения общего действия.

1.4. Типы действий по атомарности

класс действий

= *множество действий, однотипных в том или ином смысле*

<= *семейство подмножеств**:

действие

<= *разбиение**:

- {
- *атомарный класс действий*
- *неатомарный класс действий*
- }

Принадлежность некоторого *класса действий* множеству **атомарных классов действий** фиксирует факт того, что при указании всех необходимых аргументов принадлежности *действия*

данному классу достаточно для того, чтобы некоторый *субъект* мог приступить к выполнению этого действия.

При этом, даже если *класс действий* принадлежит множеству **атомарных классов действий**, не запрещается вводить более частные *классы действий*, для которых, например, заранее фиксируется один из аргументов.

Если конкретный **атомарный класс действий** является более частным по отношению к *действиям в sc-памяти*, то это говорит о наличии в текущей версии системы как минимум одного *sc-агента*, ориентированного на выполнение действий данного класса.

Принадлежность некоторого *класса действий* множеству **неатомарных классов действий** фиксирует факт того, что даже при указании всех необходимых аргументов принадлежности *действия* данному классу недостаточно для того, чтобы некоторый *субъект* приступил к выполнению этого действия, и требуются дополнительные уточнения.

2. Типология субъектов деятельности

Формальная спецификация понятия *субъект* в SCn-коде (в том числе – классификация субъектов):

субъект

= активная сущность

= сущность, способная самостоятельно выполнять некоторые виды действий

= агент деятельности

=> включение*:

- Собственное Я
- внутренний субъект *ostis-системы*
- внешний субъект *ostis-системы*, с которым осуществляется взаимодействие
- внешний субъект *ostis-системы*, с которым взаимодействие не происходит

Под **внутренним субъектом *ostis-системы*** понимается такой *субъект*, который выполняет некоторые *действия* в той же памяти, в которой хранится его знак.

К числу **внутренних субъектов *ostis-системы*** относятся входящие в нее *sc-агенты*, частные *sc-машины*, целые интеллектуальные подсистемы.

К числу **внешних субъектов *ostis-системы***, с которыми осуществляется взаимодействие, относятся конечные пользователи *ostis-системы*, ее разработчики, а также другие компьютерные системы (причем, не только интеллектуальные).

3. Средства детализации процесса выполнения действий

Рассмотрим набор отношений, предназначенных для описания детализации процесса выполнения того или иного действия, то есть выделения более простых частных действий.

декомпозиция действия*

= сведение действия ко множеству более простых взаимосвязанных действий*

∈ отношение декомпозиции

∈ квазибинарное отношение

Связки отношения **декомпозиция действия*** связывают *действие*, и множество частных *действий*, на которые декомпозируется данное *действие*. При этом первым компонентом связки является знак указанного множества, вторым компонентом – знак более общего *действия*.

Таким образом, **декомпозиция действия*** это *квазибинарное отношение*, связывающее действие со множеством действий более низкого уровня, к выполнению которых сводится выполнение исходного декомпозируемого действия.

Стоит отметить, что каждое *действие* может иметь несколько вариантов декомпозиции в зависимости от конкретного набора элементарных действий, которые способна выполнять та или иная система *субъектов*.

Принцип, по которому осуществляется такая декомпозиция в различных подходах к решению задач будем называть стратегией решения задач.

поддействие*

= частное действие*

∈ бинарное отношение

∈ отношение таксономии

Связки отношения **поддействие*** связывают *действие*, и некоторое более простое частное *действие*, выполнение которого необходимо для выполнения исходного более общего *действия*.

последовательность действий*

= порядок действий*

= бинарная ориентированная связка, описывающая то, какое действие может быть инициировано после завершения выполнения другого (предшествующего)*

= бинарная ориентированная связка, описывающая передачу управления от одного (предшествующего) действия к другому (последующему)*

= goto*

∈ отношение порядка

=> включение*:

- последовательность действий при положительном результате*
- последовательность действий при отрицательном результате*

Связки отношения **последовательность действий*** связывают знаки *действий*, выполняющихся в какой-либо последовательности в процессе решения какой-либо задачи. При этом считается, что если два *действия* связаны данным отношением, то *действие*, стоящее в данной связке на втором месте может быть выполнено только после выполнения *действия*, стоящего в данной связке на первом месте. Таким образом, каждое

действие может быть инициировано после завершения выполнения любого из предшествующих действий.

Для обеспечения возможности синхронизации выполнения действий используется класс действий *конъюнкция предшествующих действий*.

При этом дополнительно может указываться абсолютный *приоритет действия*, характеризующий принципиальную важность действия и срочность его выполнения, не всегда зависящую напрямую от других действий, но при этом влияющую на порядок выполнения действий из некоторого множества в целом.

конъюнкция предшествующих действий

= действие, заключающееся только в ожидании установлении факта завершения всех предшествующих действий

<= включение*:

действие

Действия класса *конъюнкция предшествующих действий* используются в тех случаях, когда выполнение некоторого действия должно начинаться только после того, как будут выполнены все предшествующие действия, а не только одно из них. После того, как все предшествующие действия выполнены, инициируются действия, следующие за *конъюнкцией предшествующих действий*.

В некоторых случаях бывает необходимо управлять процессом выполнения какой-либо последовательности действий в зависимости от выполнения дополнительных условий. Для осуществления таких проверок вводится класс действий *проверка условия*.

проверка условия

= if-действие

= действие, направленное на установление истинности или ложности заданного высказывания

<= включение*:

действие

Действия класса *проверка условия* предполагают проверку истинности или ложности некоторого высказывания (условия), и после выполнения в зависимости от результата данной проверки становятся *успешно выполненными действиями* или *безуспешно выполненными действиями*.

последовательность действий при положительном результате*

= then*

∈ отношение порядка

Переход по связкам отношения *последовательность действий при положительном результате** от предшествующего действия проверки условия к последующему действию происходит при условии, если указанная проверка даст положительный результат, то есть предшествующее действие станет *успешно выполненным действием*.

последовательность действий при отрицательном результате*

= else*

∈ отношение порядка

Переход по связкам отношения *последовательность действий при отрицательном результате** от предшествующего действия проверки условия к последующему действию происходит при условии, если указанная проверка даст отрицательный результат, то есть предшествующее действие станет *безуспешно выполненным действием*.

4. Спецификация действий

4.1. Задача как спецификация действия

Под *задачей* понимается формальное описание условия некоторой задачи, то есть, по сути, формальная спецификация некоторого действия, направленного на решение данной задачи, достаточная для выполнения данного действия каким-либо *субъектом*. В зависимости от конкретного класса задач, описываться может как внутреннее состояние самой интеллектуальной системы, так и требуемое состояние внешней среды. *sc-элемент*, обозначающий *действие* входит в *задачу* под атрибутом *ключевой sc-элемент'*.

Каждая *задача* представляет собой спецификацию действия, которое либо уже выполнено, либо выполняется в текущий момент (в настоящее время), либо планируется (должно) быть выполненным, либо может быть выполнено (но не обязательно).

Классификация задач может осуществляться по дидактическому признаку в рамках каждой предметной области, например, задачи на треугольники, задачи на системы уравнений и т.п.

Каждая *задача* может включать:

- факт принадлежности *действия* какому-либо частному классу *действий* (например, *действие. сформировать полную семантическую окрестность указываемой сущности*), в том числе состояния *действия* с точки зрения жизненного цикла (инициированное, выполняемое и т.д.);
- описание *цели** (*результата**) *действия*, если она точно известна;
- указание *заказчика** *действия*;
- указание *исполнителя** *действия* (в том числе, коллективного);
- указание *аргумента(ов) действия'*;
- указание инструмента или посредника *действия*;
- описание *декомпозиции действия**;
- указание *последовательности действий** в рамках *декомпозиции действия**, т.е построение плана решения задачи. Другими словами, построение плана решения представляет собой декомпозицию соответствующего *действия* на

систему взаимосвязанных между собой поддействий;

- указание области *действия*;
- указание условия инициирования *действия*;
- момент начала и завершения *действия*, в том числе планируемый и фактический, предполагаемая и/или фактическая длительность выполнения;

Некоторые задачи могут быть дополнительно уточнены контекстом – дополнительной информацией о сущностях, рассматриваемых в формулировке *задачи*, т.е. описанием того, что дано, что известно об указанных сущностях.

Построение плана решения задачи это декомпозиция спецификации процесса решения заданной задачи на систему

Кроме этого, *задача* может включать любую дополнительную информацию о действии, например:

- перечень ресурсов и средств, которые предполагается использовать при решении задачи, например список доступных исполнителей, временные сроки, объем имеющихся финансов и т.д.;
- ограничение области, в которой выполняется *действие*, например, необходимо заменить одну *sc-конструкцию* на другую по некоторому правилу, но только в пределах некоторого *раздела базы знаний*;
- ограничение знаний, которые можно использовать для решения той или иной задачи, например, необходимо решить задачу по алгебре используя только те утверждения, которые входят в курс школьной программы до седьмого класса включительно, и не используя утверждения, изучаемые в старших классах;
- и прочее

С одной стороны, решаемые системой задачи, можно классифицировать на *информационные задачи* и *поведенческие задачи*.

С точки зрения формулировки поставленной задачи можно выделить *декларативные формулировки задачи* и *процедурные формулировки задачи*. Следует отметить, что данные классы задач не противопоставляются, и могут существовать формулировки задач, использующие оба подхода.

Формальная спецификация понятия задача в SCn-коде:

задача

<= *включение**:

семантическая окрестность

=> *включение**:

- *процедурная формулировка задачи*
- *декларативная формулировка задачи*

=> *включение**:

- *вопрос*
- *команда*

В случае *процедурной формулировки задачи*, в формулировке задачи явно указываются аргументы соответствующего задаче *действия*, и в частности, вводится семантическая типология *действий*. При этом явно не уточняется, что должно быть результатом выполнения данного действия. Заметим, что, при необходимости, *процедурная формулировка задачи* может быть сведена к *декларативной формулировке задачи* путем трансляции на основе некоторого правила, например определения класса действия через более общий класс.

В случае *декларативной формулировки задачи*, при описании условия задачи специфицируется цель *действия*, т.е. результат, который должен быть получен при успешном выполнении *действия*.

Под *вопросом* понимается задача, направленная на удовлетворение информационной потребности некоторого субъекта-заказчика

Под командой понимается спецификация инициированного действия, т.е., по сути, инициированная задача.

класс команд

<= *семейство подмножеств**:

команда

=> *включение**:

- *класс интерфейсных пользовательских команд*

=> *включение**:

- *класс интерфейсных команд пользователя ostis-системы*

=> *включение**:

- *класс команд без аргументов*
- *класс команд с одним аргументом*
- *класс команд с двумя аргументами*
- *класс команд с произвольным числом аргументов*

<= *разбиение**:

{

- *атомарный класс команд*
- *неатомарный класс команд*

}

Принадлежность некоторого *класса команд* множеству *атомарных классов команд* фиксирует факт того, что данная спецификация является достаточной для того, чтобы некоторый субъект приступил к выполнению соответствующего действия.

При этом, даже если *класса команд* принадлежит множеству *атомарных классов команд* не запрещается вводить более частные *классы команд*, в состав которых входит информация, дополнительно специфицирующая соответствующее *действие*.

Если соответствующий данному *классу команд* *класс действий* является более частным по отношению к *действиям в sc-памяти*, то попадание данного класса команд во множество *атомарных*

классов команд говорит о наличии в текущей версии системы как минимум одного *sc-агента*, условие инициирования которого соответствует формулировке команд данного класса.

Принадлежность некоторого *класса команд* множеству **неатомарных классов команд** фиксирует факт того, что данная спецификация не является достаточной для того, чтобы некоторый субъект приступил к выполнению соответствующего действия, и требует дополнительных уточнений.

4.2. Отношения, специфицирующие действия

Рассмотрим набор отношений, используемых для формальной спецификации действий в рамках задачи.

Связки отношения **результат*** связывают *sc-элемент*, обозначающий действие, и *sc-конструкцию*, описывающую результат выполнения рассматриваемого действия, другими словами, цель, которая должна быть достигнута при выполнении действия.

Результат может специфицироваться как атомарным высказыванием, так и неатомарным, т.е. конъюнктивным, дизъюнктивным, строго дизъюнктивным и т.д.

В случае, когда успешное выполнение действия приводит к изменению какой-либо конструкции в *sc-памяти*, которое необходимо занести в историю изменений базы знаний или использовать для демонстрации протокола решения задачи, генерируется соответствующая связка отношения **результат***, связывающая задачу и *sc-конструкцию*, описывающую данное изменение. Конкретный вид указанной *sc-конструкции* зависит от типа действия.

Связки отношения **исполнитель*** связывают *sc-элементы*, обозначающие действие и *sc-элементы*, обозначающие субъекта, который предположительно будет осуществлять, осуществляет или осуществлял выполнение указанного действия. Данное отношение может быть использовано при назначении конкретного исполнителя для проектной задачи по развитию баз знаний.

В случае, когда заранее неизвестно, какой именно субъект* будет исполнителем данного действия, связка отношения **исполнитель*** может отсутствовать в первоначальной формулировке задачи и добавляться позже, уже непосредственно при исполнении.

Когда действие выполняется (является настоящей сущностью) или уже выполнено (является прошлой сущностью), то исполнитель этого действия в каждый момент времени уже определен. Но когда действие только инициировано, тогда важно знать:

1) кто хочет выполнить это действие и насколько важно для него стать исполнителем данного действия;

2) кто может выполнить данное действие и каков уровень его квалификации и опыта;

3) кто и кому поручает выполнить это действие и каков уровень ответственности за невыполнение (приказ, заказ, официальный договор, просьба...);

При этом следует помнить, что связь отношения **исполнитель*** в данном случае также является временной прогнозируемой сущностью.

Первым компонентом связок отношения **исполнитель*** является знак действия, вторым - знак субъекта-исполнителя.

Связки отношения **заказчик*** связывают *sc-элементы*, обозначающие действие и *sc-элементы*, обозначающие субъекта, который «заинтересован» в выполнении данного действия и, как правило, инициирует его выполнение. Данное отношение может быть использовано при указании того, кто поставил проектную задачу по развитию баз знаний.

Первым компонентом связок отношения **заказчик*** является знак действия, вторым - знак субъекта-заказчика.

Связки отношения **инициатор*** связывают *sc-элемент*, обозначающий инициированное действие, и знак субъекта, который является инициатором данного действия, то есть субъектом, который инициировал данное действие и, как правило, заинтересован в его успешном выполнении.

Связки отношения **объект*** связывают *sc-элемент*, обозначающий действие, и знак той сущности, над которой (по отношению к которой) осуществляется данное действие, например знак структуры, подлежащий верификации.

Связки отношения **контекст действия*** связывают *sc-элементы*, обозначающие действие и *sc-структуры*, обозначающие контекст выполнения данного действия, то есть некоторую дополнительную информации о тех сущностях, которые входят в описание цели*. Как правило, контекст используется для указания собственно условия некоторой задачи, того, что дано, т.е. тех знаний, которые можно использовать для вывода новых знаний при решении задачи. Таким образом, контекст непосредственно влияет на то, как будет решаться та или иная задача, при этом даже задачи соответствующие одному классу действий, могут решаться по-разному.

контекст действия*

= *задачная ситуация**

= *что дано**

= *дополнительная информация о тех сущностях, которые входят в описание цели**

= *связь между некоторой задачей (формулировкой задачи) и состоянием базы знаний, возможностей и навыков некоторого субъекта,*

*перед которым поставлена указанная задача**
= *связь между формулировкой задачи, т.е. описанием того, что требуется, и контекстом этой задачи, т.е. описанием имеющихся ресурсов, описанием того, что дано**

Контекст может быть представлен не только в виде атомарного фактографического высказывания, но и в виде высказывания более сложного вида. Это может быть, например:

- определение множества, используемого в описании *цели**;
- утверждение, учет которого может быть полезен в решении задач;

Первым компонентом связок отношения *контекст действия** является знак *действия*, вторым - знак *структуры* [Давыденко, 2016a], обозначающей контекст.

аргумент действия'

=> *включение**:

- *первый аргумент действия'*
- *второй аргумент действия'*
- *третий аргумент действия'*

∈ *ролевое отношение*

Связки ролевого отношения **аргумент действия'** указываются в рамках конкретного действия те *sc-элементы*, которые обозначают непосредственно аргументы данного *действия*, если они явно указываются (в случае процедурной формулировки задачи).

класс аргументов*

= *класс аргументов класса команд**

= *быть классом sc-элементов, экземпляры которого являются аргументами для заданного класса команд**

=> *включение**:

- *класс первых аргументов**
- *класс вторых аргументов**

Связки отношения **класс аргументов*** связывают *классы команд* (подмножества множества *команд*), и *классы sc-элементов*, которые могут быть аргументами действий, соответствующих данному *классу команд*. В случае, когда *команды* данного класса имеют один аргумент, используется собственно отношение **класс аргументов***, в случае, когда больше команд данного класса имеют более одного аргумента, то используются подмножества данного отношения, такие как *класс первых аргументов**, *класс вторых аргументов** и т.д.

Если для некоторого *класса команд* не указан тип какого-либо из аргументов, то предполагается, что в качестве данного аргумента может выступать любой *sc-элемент*.

Первым компонентом связок отношения **класс аргументов*** является знак *класса команд*, вторым – знак *класса sc-элементов*, которые могут быть

аргументами действий', соответствующих данному *классу команд*.

4.3. Другие виды семантических спецификаций действий

Кроме задачи как спецификации действия в целом необходимо выделить еще ряд *семантических окрестностей* [Давыденко, 2016a] специфицирующих конкретное действие с той или иной стороны.

4.3.1. План выполнения действия

Каждый **план** представляет собой *семантическую окрестность*, *ключевым sc-элементом'* является *действие*, для которого дополнительно детализируется предполагаемый процесс его выполнения. Основная задача такой детализации – локализация области базы знаний, в которой предполагается работать, а также набора агентов, необходимого для выполнения описываемого действия. При этом детализация не обязательно должна быть доведена до уровня элементарных действий, цель составления плана – уточнение подхода к решению той или иной задачи, не всегда предполагающее составления подробного пошагового решения.

При описании **плана** может быть использован как процедурный, так и декларативный подход. В случае процедурного подхода для соответствующего *действия* указывается его декомпозиция на более частные поддействия, а также необходимая спецификация этих поддействий. В случае декларативного подхода указывается набор подцелей (например, при помощи логических утверждений), достижение которых необходимо для выполнения рассматриваемого *действия*. На практике оба рассмотренных подхода можно комбинировать.

В общем случае **план** может содержать и переменные, например в случае, когда часть плана задается в виде цикла (многократного повторения некоторого набора действий). Также план может содержать константы, значение которых в настоящий момент не установлено и станет известно, например, только после выполнения предшествующих ему *действий*.

Каждый **план** может быть задан заранее как часть спецификации *действия*, т.е. *задачи*, а может формироваться *субъектом* уже собственно в процессе выполнения *действия*, например, в случае использования стратегии разбиения задачи на подзадачи. В первом случае **план включается*** в *задачу*, соответствующую тому же действию.

4.3.2. Программа выполнения класса действий

программа

= *программа выполнения действий некоторого класса*

= *обобщенный план*

= *обобщенный план выполнения некоторого класса действий*

= обобщенный план решения некоторого класса задач
 = обобщенная спецификация декомпозиции любого действия, принадлежащего заданному классу действий
 = знание о некотором классе действий (и соответствующем классе задач), позволяющее для каждого из указанных действий достаточно легко построить план его выполнения
 <= включение*:
 знание
 => включение*:
 программа в *sc*-памяти

Каждая **программа** представляет собой обобщенный план выполнения действий, принадлежащих некоторому классу, то есть семантическую окрестность, ключевым *sc*-элементом' является класс действий, для элементов которого дополнительно детализируется процесс их выполнения.

В остальном описание **программы** аналогично описанию плана выполнения конкретного действия из рассматриваемого класса действий.

Одному классу действий может соответствовать несколько **программ**.

Входным параметрам **программы** в традиционном понимании соответствуют аргументы, соответствующие каждому действию из класса действий, описываемого программой. При генерации на основе **программы** плана выполнения конкретного действия из данного класса эти аргументы принимают конкретные значения.

Каждая **программа** представляет собой систему описанных действий с дополнительным указанием для действия:

- либо последовательности выполнения действий* (передачи инициирования), когда условием выполнения (инициирования) действий является завершение выполнения одного из указанных или всех указанных действий;
- либо события в базе знаний или внешней среде, являющегося условием его инициирования;
- либо ситуации в базе знаний или внешней среде, являющейся условием его инициирования;

Рассмотрим пример программы базового языка программирования, ориентированного на обработку семантических сетей – Языка SCP (Рисунок 1). Данный пример представляет собой *scp*-программу, имеющую два аргумента (множество и элемент, который надо добавить во множество). В соответствии с алгоритмом программы осуществляется проверка того, имеется ли указанный элемент в указанном множестве, и если нет, то добавляет элемент во множество. В рамках данной программы исходный *scp*-процесс

декомпозируется на три более простых подпроцесса (поддействия*), связанных соответствующими подклассами отношения последовательность выполнения действий*. Часть используемых в данном примере понятий идентифицировано при помощи системных идентификаторов*, например *nrel_then*, *nrel_else* и т.д.

4.3.3. Протокол выполнения действия

Каждый **протокол** представляет собой семантическую окрестность, ключевым *sc*-элементом' является действие, для которого собственно описывается весь процесс его выполнения, то есть все более простые поддействия, в том числе те, выполнение которых, как выяснилось позже, не было целесообразным. Подразумевается, что *sc*-элемент, обозначающий данное действие, входит во множество прошлых сущностей.

Таким образом, **протокол** представляет собой *sc*-текст, содержащий декомпозицию рассматриваемого действия на поддействия с указанием порядка их выполнения, а также необходимой спецификацией каждого такого поддействия.

В отличие от плана, **протокол** всегда формируется по факту выполнения соответствующего действия.

4.3.4. Решение задачи, соответствующей действию

Каждое **решение** представляет собой семантическую окрестность, ключевым *sc*-элементом' является действие, для которого собственно описывается процесс его выполнения, то есть решение соответствующей задачи. Подразумевается, что *sc*-элемент, обозначающий данное действие, входит во множество успешно выполненных действий.

Таким образом, **решение** представляет собой *sc*-текст, содержащий декомпозицию рассматриваемого действия на поддействия с указанием порядка их выполнения, а также необходимой спецификацией каждого такого поддействия.

Стоит отметить, что в случае отношения **решение*** в декомпозиции действия* указываются только те поддействия, без которых решение поставленной задачи было бы невозможным, то есть из протокола исключаются ложные или избыточные шаги, сделанные в процессе поиска пути решения задачи, которые, в свою очередь, могут присутствовать при описании непосредственно текущего хода решения задачи.

Для конкретного действия его **решение** будет нестрого включаться* в соответствующий протокол решения.

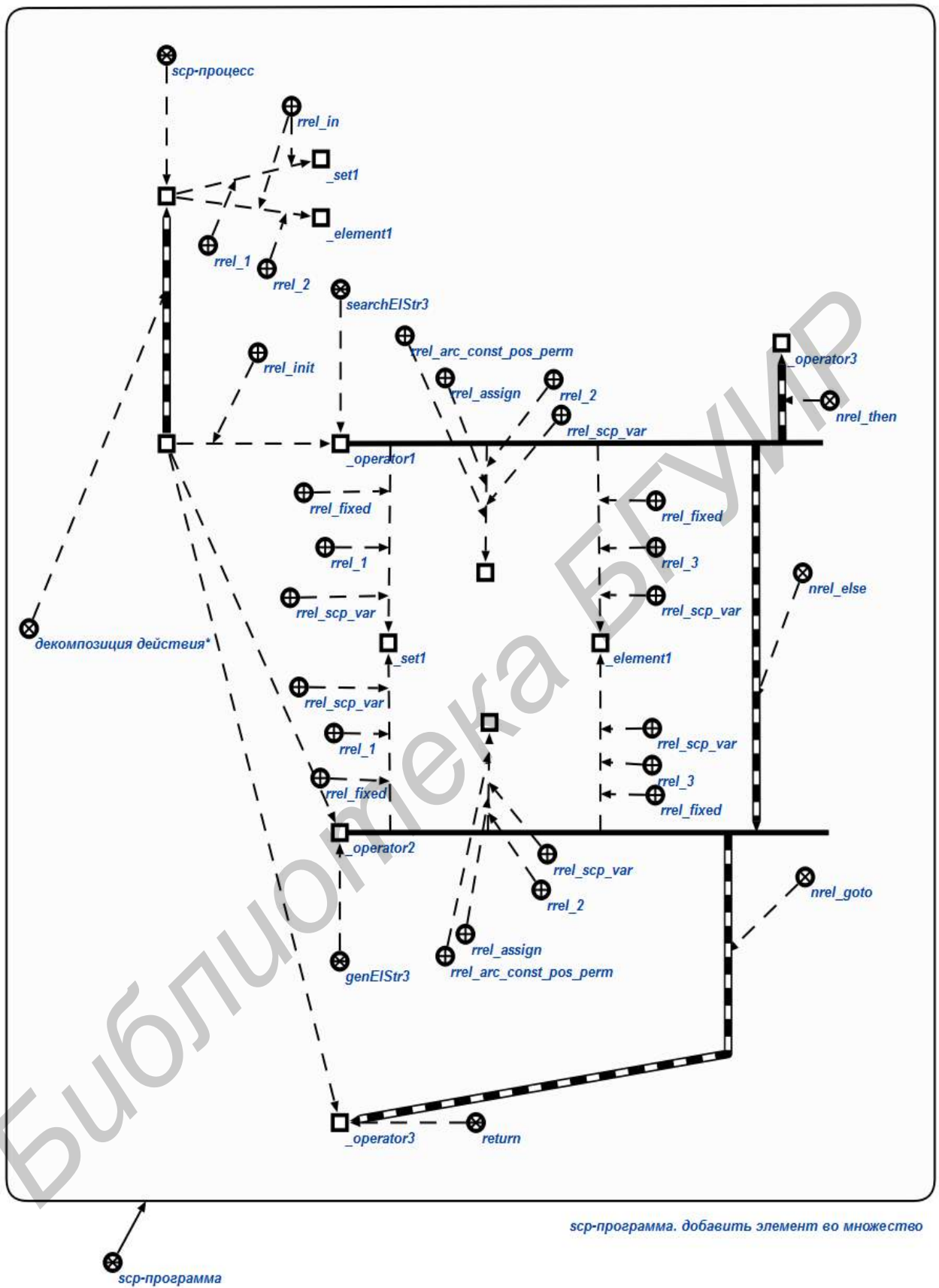


Рисунок 1 – Пример программы на языке SCP

Заклучение

В работе рассмотрены базовые принципы описания целенаправленной деятельности различного вида субъектов на основе семантических сетей с базовой теоретико-множественной интерпретацией, применимые для описания в семантической памяти деятельности различного рода субъектов.

В работе представлены результаты, полученные в научных группах НИУ «МЭИ» (кафедра прикладной математики) и БГУИР (кафедра интеллектуальных информационных технологий) совместно.

Работа выполнена при поддержке гранта БРФФИ-РФФИ-М Ф15РМ-074 (номер с российской стороны - 15-57-04047 Бел-мол-а) «Методы и средства онтологического моделирования для семантических технологий проектирования интеллектуальных систем».

Библиографический список

[**Boyko, 2016**] Boyko, I. Semantic classification of actions for knowledge inference. Open semantic technologies for intelligent systems (OSTIS-2016): materials of VI International.sc.-tech.conf./ I. Boyko // Мн.: BSUIR, 2016

[**IMS, 2016**] Метасистема IMS.OSTIS [Электронный ресурс]. Минск, 2016. – Режим доступа: <http://ims.ostis.net/>. – Дата доступа: 15.01.2016.

[**Голенков, 2015**] Голенков, В.В. Семантическая технология компонентного проектирования систем, управляемых знаниями. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар.научн.-техн.конф./ В. В. Голенков, Н.А Гулякина// Мн.: БГУИР, 2015

[**Голенков, 2012**] Голенков, В.В., Гулякина Н.А. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования./В.В. Голенков, Н.А., Гулякина// Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2012): материалы Междунар. научн.-техн.конф. Мн.: БГУИР, 2012 – С.23-52

[**Давыденко и др., 2016а**] Давыденко, И.Т. Описание свойств различных видов знаний в системах, управляемых знаниями. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2016): материалы VI Междунар.научн.-техн.конф./ И. Т. Давыденко [и др.] // Мн.: БГУИР, 2016

[**Давыденко и др., 2016б**] Давыденко, И.Т. Иерархическая структуризация баз знаний систем, управляемых знаниями. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2016): материалы VI Междунар.научн.-техн.конф./ И. Т. Давыденко [и др.] // Мн.: БГУИР, 2016

[**Мартынов, 1974**] Мартынов, В.В. Семиологические основы информатики / В.В. Мартынов - Мн.: Наука и техника, 1974. – 192 с.

[**Мартынов, 1977**] Мартынов, В.В. Универсальный семантический код (Грамматика. Словарь. Тексты) / В.В. Мартынов – Мн.: Наука и техника, 1977. – 191 с.

[**Мартынов, 1984**] Мартынов, В.В. Универсальный семантический код: УСК-3 / В.В. Мартынов – Мн.: Наука и техника, 1984. – 132 с.

[**Тарасов, 2002**] Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям / В.Б. Тарасов; – М. :Изд-во УРСС, 2002.

[**Шенк, 1980**] Шенк, Р. Обработка концептуальной информации. Пер. с англ. / Р. Шенк; – М.: Энергия; 1980.

[**Шункевич, 2014**] Шункевич, Д.В. Машина обработки знаний интеллектуальной метасистемы поддержки проектирования интеллектуальных систем. /Д.В. Шункевич//

Открытые семантические технологии интеллектуальных систем (Ostis-2014): Материалы IV международной науч.-тех. конф -- Минск: БГУИР,2014. – с.93-96

FORMAL SEMANTIC DESCRIPTION OF THE VARIOS TYPES OF ACTORS PURPOSEFUL ACTION

Shunkevich D.V. *, Hubarevich A.V. *, Svyatkina M.N. **, Morosin O.L. ***

**Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

shunkevichdv@gmail.com

st.hubarevich@gmail.com

***Bauman Moscow State Technical University, Moscow, Russia*

maria.svyatkina@gmail.com

****Moscow Power Engineering Institute*

oleg@morosin.ru

In this paper the formal models, that allowed to describe the varios types of actors purposeful action by the semantic networks with set-theoretic interpretation, are considered. Particularly, the tepology of actions, the concept of task as the action specification, relations, used for action specification are considered. At the same time taked up the other classes of action specification: plan, programm, solution, ect.

Key words: semantic technologies, the action formalisation, multy-agent systems, OSTIS project.

Introduction

The aim of this work is to develop the basic principles of the description of the purposeful activity of various types of subjects based on semantic networks with set-theoretic interpretation. The models and instruments submitted in this work are used in the knowledge-driven systems, built on the OSTIS tecnology, development.

This work is a part of the knowledge base of the IMS OSTIS system, in which it is presented as the section, describing the relevant domain.

Main Part

This paper considered of the next sections:

- The basic typology of action:
 - The concept of action and the typology of it;
 - Types of action towards the information system's memory;
 - Types of action towards the current time;
 - Types of action atomicity;
- The typology of action subjects;
- Instruments of process of performing actions detalisation;
- Specification of actions:
 - Task as the action specification
 - Action specification relations
 - Other types of action specification



УДК 004.822:514

ВЗАИМОДЕЙСТВИЕ АСИНХРОННЫХ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ ОБРАБОТКИ ЗНАНИЙ В ОБЩЕЙ СЕМАНТИЧЕСКОЙ ПАМЯТИ

Шункевич Д.В.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

shunkevichdv@gmail.com

В работе рассматривается базовая модель построения машин обработки знаний систем, управляемых знаниями, построенных по открытой семантической технологии проектирования интеллектуальных систем. В частности, описываются базовые принципы взаимодействия асинхронных параллельных процессов обработки знаний в общей семантической памяти, типология действий, выполняемых агентами в такой памяти, типология и средства описания блокировок, необходимых для синхронизации деятельности агентов над общей семантической памятью.

Ключевые слова: семантические технологии, формализация деятельности, многоагентные системы, проект OSTIS.

Введение

Системы, управляемые знаниями, построенные по открытой семантической технологии проектирования интеллектуальных систем (Технологии OSTIS) будем называть *ostis-системами* [Голенков, 2015]. В качестве модели представления знаний в данной технологии используется унифицированная семантическая сеть с теоретико-множественной интерпретацией [Голенков и др., 2001]. Такую модель представления будем называть *SC-кодом* (Semantic computer code) [Голенков, 2012]. Элементы такой семантической сети будем называть *sc-узлами* и *sc-коннекторами* (*sc-дугами*, *sc-ребрами*).

Каждая *ostis-система* состоит из не зависящей от платформы реализации унифицированной логико-семантической модели этой системы (*sc-модели компьютерной системы*) и платформы интерпретации таких моделей. В свою очередь каждая *sc-модель компьютерной системы* может быть декомпозирована на *sc-модель базы знаний*, *sc-модель машины обработки знаний*, *sc-модель интерфейса*. В данной работе внимание будет уделено *sc-модели машины обработки знаний*.

В рамках Технологии OSTIS *sc-модель машины обработки знаний* включает графодинамическую *sc-память*, в которой хранятся конструкции *SC-кода*, и коллектив агентов взаимодействующих через эту память (*sc-агентов*).

Данная модель предполагает, что каждый *sc-агент* реагирует на соответствующий ему класс ситуаций и/или событий, происходящих в *sc-памяти*, и осуществляет определенное преобразование *sc-текста* (текста *SC-кода*), находящегося в семантической окрестности обрабатываемой ситуации и/или события.

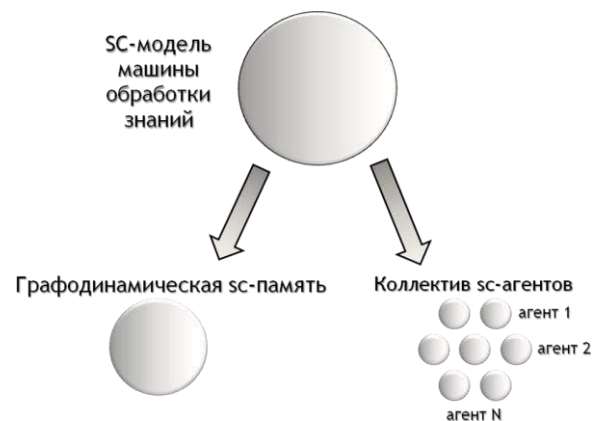


Рисунок 1 – Модель машины обработки знаний

Более подробно семантическая модель машины обработки знаний и преимущества такого подхода к ее построению рассмотрены в [Голенков, 2012], [Голенков, 2013], [Шункевич, 2013].

Перечислим некоторые достоинства предлагаемого подхода к построению машин обработки знаний:

- поскольку машина обработки знаний состоит из агентов, которые обмениваются сообщениями только через общую память, добавление нового агента или исключение (деактивация) одного или нескольких существующих агентов, как правило, не приводит к изменениям в других агентах, поскольку агенты не обмениваются сообщениями напрямую;

- часто агенты работают параллельно и независимо друг от друга, выполняя разные *действия в sc-памяти*; таким образом, даже существенное расширение числа агентов в рамках одной системы не приводит к ухудшению ее производительности;

- в качестве основного языка реализации программ агентов обработки знаний предполагается использовать *Язык SCP*, тексты программ которого записываются при помощи тех же семантических сетей, что и обрабатываемая информация, при этом подход к интерпретации *scp-программ (программ Языка SCP)* предполагает создание при каждом вызове *scp-программы* уникального *scp-процесса*. Таким образом, одновременно могут выполняться несколько операций, при этом разные копии агентов могут выполняться на разных серверах, за счет распределенной реализации интерпретатора *sc-моделей* (платформы реализации *sc-моделей*). Более того, язык реализации программ агентов позволяет осуществлять параллельные асинхронные вызовы подпрограмм с последующей синхронизацией, и даже параллельно выполнять операторы в рамках одной программы;

- как уже было сказано, программы, описывающие деятельность агентов, могут быть записаны на том же языке, что и обрабатываемые знания, и храниться в той же базе знаний, равно как и их спецификации. Таким образом, перенос агента из одной системы в другую заключается в простом переносе фрагмента базы знаний, без каких-либо дополнительных операций, зависящих от платформы интерпретации;

- спецификации агентов и их программы могут быть записаны на том же языке, что и обрабатываемые знания, что существенно сокращает перечень специализированных средств, предназначенных для проектирования машин обработки знаний, и упрощает их разработку за счет использования более универсальных компонентов;

- тот факт, что для интерпретации *scp-программы* создается соответствующий ей уникальный *scp-процесс*, позволяет по возможности оптимизировать план выполнения перед его реализацией и даже непосредственно в процессе выполнения без потенциальной опасности испортить общий универсальный алгоритм всей программы.

Следует отдельно отметить факт возможности параллельного выполнения процессов (как правило-целенаправленных действий в *sc-памяти*), а также возможности параллельного выполнения операторов в рамках одной *scp-программы*. При этом отнесение какой-либо конкретной машины

обработки знаний к какому-либо классу параллельных систем, рассматриваемых в литературе [Хоар, 1989], [Шпаковский, 2013] представляется возможным только, во-первых, с учетом конкретных классов решаемых такой машиной задач, во-вторых, с учетом возможностей используемой *платформы реализации sc-моделей* касающихся параллельной обработки знаний.

1. Понятие процесса и действия в *sc-памяти*

Понятие *процесса* как изменения в целом более подробно описано в общих предметных областях в рамках базы знаний IMS [IMS, 2016].

Понятие *действия* вообще, как частного случая *воздействия*, и, соответственно, *процесса*, применительно к системам, управляемым знаниями, подробно рассматривается в [Шункевич, 2016].

В рамках данной работы основное внимание уделяется понятию процесса в *sc-памяти* и действия в *sc-памяти*, как целенаправленного изменения информации хранящейся в памяти *ostis-системы*.

Рассмотрим спецификацию понятия *действие в sc-памяти* в SCn-коде:

действие в sc-памяти

= *внутреннее действие ostis-системы*

= *действие, выполняемое в sc-памяти*

= *действие, выполняемое в абстрактной унифицированной семантической памяти*

= *действие, выполняемое машиной обработки знаний ostis-системы*

= *действие, выполняемое агентом или коллективом агентов ostis-системы*

= *информационный процесс над базой знаний, хранимой в sc-памяти*

= *процесс решения информационной задачи в sc-памяти*

<= включение*:

процесс в sc-памяти

Каждое *действие в sc-памяти* обозначает некоторое преобразование, выполняемое некоторым *sc-агентом* (или коллективом *sc-агентов*) и ориентированное на преобразование *sc-памяти*. Спецификация действия после его выполнения может быть включена в протокол решения некоторой задачи.

Преобразование состояния базы знаний включает, в том числе и информационный поиск, предполагающий (1) локализацию в базе знаний ответа на запрос, явное выделение структуры ответа и (2) трансляцию ответа на некоторый внешний язык.

Во множество *действий в sc-памяти* входят знаки действий самого различного рода, семантика каждого из которых зависит от конкретного контекста, т.е. ориентации действия на какие-либо

конкретные объекты и принадлежности действия какому-либо конкретному классу действий.

Следует четко отличать:

- Каждое конкретное **действие в sc-памяти**, представляющее собой некоторый переходный процесс, переводящий sc-память из одного состояния в другое;

- Каждый тип **действий в sc-памяти**, представляющий собой некоторый класс однотипных (в том или ином смысле) действий;

- sc-узел, обозначающий некоторое конкретное **действие в sc-памяти**;

- sc-узел, обозначающий структуру, которая является описанием, спецификацией, заданием, постановкой соответствующего действия.

2. Типология действий в sc-памяти

Приведем семантическую типологию действий в sc-памяти, представленную в SСп-коде:

действие в sc-памяти

=> включение*:

- действие в sc-памяти, иницируемое вопросом
- действие редактирования базы знаний *ostis-системы*
- действие установки режима *ostis-системы*
- действие редактирования файла, хранимого в sc-памяти
- действие интерпретации программы, хранимой в sc-памяти

действие в sc-памяти, иницируемое вопросом

= действие, направленное на формирование ответа на поставленный вопрос

=> включение*:

- действие. сформировать заданный файл
- действие. сформировать заданную структуру
=> включение*:
 - действие. верифицировать заданную структуру
=> включение*:
 - действие. установить истинность или ложность указываемого логического высказывания
 - действие. установить корректность или некорректность указываемой структуры
 - действие. сформировать структуру, описывающую некорректности, имеющиеся в указываемой структуре
 - действие. установить тип заданного sc-элемента
=> включение*:
 - действие. установить позитивность/негативность указываемой sc-дуги

принадлежности или принадлежности

- действие. сформировать семантическую окрестность
=> включение*:

- действие. сформировать полную семантическую окрестность указываемой сущности

- действие. сформировать базовую семантическую окрестность указываемой сущности

- действие. сформировать частную семантическую окрестность указываемой сущности

- действие. сформировать структуру, описывающую связи между указываемыми сущностями
=> включение*:

- действие. сформировать структуру, описывающую сходства указываемых сущностей

- действие. сформировать структуру, описывающую различия указываемых сущностей

- действие. сформировать структуру, описывающую способ решения указываемой задачи

- действие. сформировать план генерации ответа на указанный вопрос

- действие. сформировать протокол выполнения указываемого действия

- действие. сформировать обоснование корректности указываемого решения

- действие. верифицировать обоснование корректности указываемого решения

- действие, одним из аргументов которого является некоторая обобщенная структура

- действие, направленное на установление темпоральных характеристик указываемой сущности

- действие, направленное на установление пространственных характеристик указываемой сущности

действие редактирования базы знаний

=> включение*:

- действие. исправить ошибки в заданной структуре

- действие. преобразовать указанную структуру в соответствии с указанным правилом

- действие. отождествить два указанных sc-элемента

- действие. включить множество = сделать все элементы множества *si* явно принадлежащими множеству *sj*, то есть

сгенерировать соответствующие sc-дуги принадлежности

- действие генерации sc-элементов => включение*:

- действие генерации, одним из аргументов которого является некоторая обобщенная структура => включение*:

- действие. сгенерировать структуру, изоморфную указываемому образцу

- действие. сгенерировать sc-элемент указанного типа => включение*:

- действие. сгенерировать sc-коннектор указанного типа

- действие. сгенерировать sc-узел указанного типа

- действие. сгенерировать структуру, содержащую указанные sc-элементы

- действие. сгенерировать файл с заданным содержанием

- действие. обновить понятия = действие. заменить неиспользуемое понятие на его определение через используемое понятие

- действие. установить указанный файл в качестве основного идентификатора указанного sc-элемента

- действие. протранслировать содержимое указываемого файла в sc-память

- действие. интегрировать указанную структуру в текущее состояние базы знаний

- действие. сгенерировать структуру, описывающую историю эволюции ostis-системы

- действие. сгенерировать структуру, описывающую историю эксплуатации ostis-системы

- действие удаления sc-элементов => включение*:

- действие. удалить указанные sc-элементы => включение*:

- действие. удалить указанный sc-элемент

- действие. удалить sc-элементы, входящие в состав указанной структуры и не являющиеся ключевыми узлами каких-либо sc-агентов

- действие. исключить указанные sc-элементы из клиентской части базы знаний

Далее более детально рассмотрим некоторые из выделенных классов действий в sc-памяти.

Каждое **действие. отождествить два указанных sc-элемента** может быть выполнено как **действие. физически отождествить два указанных sc-элемента** или **действие. логически отождествить два указанных sc-элемента**. В случае логического отождествления в протоколе деятельности агентов сохраняется само действие с его спецификацией, включающей обязательное указание того, какие элементы были сгенерированы, а какие удалены. В случае физического отождествления протокол действия не сохраняется.

Каждое **действие. обновить понятия** обозначает переход от какой-то группы понятий, использовавшихся ранее, к другой группе понятий, которые будут использоваться вместо первых, и станут **основными понятиями**.

В общем случае **действие. обновить понятия** состоит из следующих этапов:

- Определить заменяемые понятия на основе заменяющих;

- Внести соответствующие изменения в программы sc-агентов, ключевыми узлами которых являются обновляемые понятия;

- Заменить все конструкции в базе знаний, содержащие заменяемые понятия, в соответствии с определениями этих понятий через заменяющие их понятия;

- При необходимости, sc-элементы, обозначающие замененные таким образом понятия, могут быть полностью выведены из текущего состояния базы знаний.

Первым аргументом (входящим в знак **действия** под атрибутом 1') **действия. обновить понятия** является знак множества sc-узлов, обозначающих заменяемые понятия, вторым (входящим в знак **действия** под атрибутом 2') - знак множества sc-узлов, обозначающих заменяющие понятия. В общем случае любое или оба этих множества могут быть **синглетами**.

Каждое **действие. удалить указанные sc-элементы** может быть выполнено как **действие. физически удалить указанные sc-элементы** или **действие. логически удалить указанные sc-элементы**. В случае логического удаления в протоколе деятельности агентов сохраняется само действие с его спецификацией, включающей обязательное указание того, какие элементы были удалены, т.е. по сути, элементы просто исключаются из текущего состояния базы знаний. В случае физического удаления протокол действия не сохраняется.

В случае удаления какого-либо sc-элемента, инцидентные ему **связки**, в том числе **sc-коннекторы**, также удаляются.

Для того, чтобы выполнить **действие. интегрировать указанную структуру в текущее состояние базы знаний**, необходимо склеить sc-элементы, входящие в интегрируемую **структуру** с синонимичными им sc-элементами, входящими в

текущее состояние базы знаний, заменить неиспользуемые (например, устаревшие) понятия, входящие в интегрируемую *структуру*, на используемые (т.е. заменить неиспользуемые понятия на их определения через используемые), явно включить все элементы интегрируемой *структуры* в число элементов утвержденной части базы знаний и явно включить все элементы интегрируемой *структуры* в число элементов одного из атомарных разделов утвержденной части базы знаний.

3. Принципы взаимодействия агентов в *sc-памяти*

Как было сказано ранее, все действия в *sc-памяти* независимо от конкретного класса выполняются *sc-агентами*, которые могут в том числе, инициироваться пользователями *ostis-системы*.

Понятие агента в *sc-памяти* (*sc-агента*) и их типология подробно рассмотрены в [Шункевич, 2014], теория многоагентных систем и типология агентов в целом описаны в [Тарасов, 2002].

Ниже в данной работе подробно рассматриваются принципы взаимодействия *sc-агентов* в памяти *ostis-системы*.

3.1. Общие принципы организации взаимодействия *sc-агентов* и пользователей *ostis-системы* через общую *sc-память*.

Каждая *ostis-система* представляет собой многоагентную систему, агенты которой взаимодействуют между собой только (!) через общую для них *sc-память*. При этом пользователи *ostis-системы* также считают агентами этой системы. Кроме того, *sc-агенты* делятся на внутренние, рецепторные и эффекторные. Взаимодействие между агентами через общую *sc-память* сводится к следующим видам действий:

(1) К использованию общедоступных для соответствующей группы агентов части хранимой базы знаний. В простейшем случае по уровню прав доступа агенты *ostis-системы* разбиваются на две группы – главные администраторы базы знаний (их может быть несколько) вместе с обслуживающими их *sc-агентами* и все остальные агенты;

(2) К формированию (генерации) новых фрагментов базы знаний и/или к корректировке (редактированию) каких-либо фрагментов доступной части базы знаний;

(3) К интеграции (погружению) новых и/или обновленных фрагментов в состав доступной части базы знаний;

Пользователь *ostis-системы* не может сам непосредственно выполнить какое-либо действие в *sc-памяти*, но он может средствами пользовательского интерфейса инициировать построение (генерацию, формирование в *sc-памяти*) *sc-текста*, являющегося спецификацией

действия в sc-памяти, выполняемого либо одним атомарным *sc-агентом* за один акт, либо одним атомарным *sc-агентом* за несколько актов, либо коллективом *sc-агентов*. В спецификации каждого такого *действия в sc-памяти*, инициированного пользователем, этот пользователь указывается как заказчик этого действия. Таким образом, пользователь *ostis-системы* дает поручения (задания, команды) *sc-агентам* этой системы на выполнение различных специфицируемых им действий в *sc-памяти*.

Каждый *sc-агент*, выполняя некоторое *действие в sc-памяти*, должен «помнить», что *sc-память*, над которой он работает, является общим ресурсом не только для него, но и для всех остальных *sc-агентов*, работающих над этой же *sc-памятью*. Поэтому *sc-агент* должен соблюдать определенную этику поведения в коллективе таких *sc-агентов*, которая должна минимизировать помехи, которые он создает другим *sc-агентам*.

Деятельность каждого агента *ostis-системы* дискретна и представляет собой множество элементарных действий (актов). При этом при выполнении практически каждого акта агент выделяет некий фрагмент базы знаний, который вообще не должен быть «виден» другим агентам (только ему самому) и/или некоторый фрагмент базы знаний, который может быть «виден», но не может изменяться другими агентами. Указанная блокировка – это своего рода «забор» (ограждение) через который другим агентам перелезть запрещено. Эта блокировка устанавливается самим агентом при выполнении соответствующего акта и снимается им же на последнем этапе выполнения этого акта.

Если некий *sc-агент* выполняет некоторое действие в *sc-памяти*, то он на время выполнения этого действия может:

(1) Запретить другим *sc-агентам* изменять состояние некоторых *sc-элементов*, хранимых в *sc-памяти* – удалять их, изменять тип;

(2) Запретить другим *sc-агентам* добавлять или удалять элементы некоторых множеств, обозначаемых соответствующими *sc-узлами*;

(3) Запретить другим *sc-агентам* доступ на просмотр некоторых *sc-элементов*, то есть эти *sc-элементы* становятся полностью «невидимыми» (полностью заблокированными) для других *sc-агентов* но только на время выполнения соответствующего действия.

Указанные блокировки должны быть полностью сняты до завершения выполнения соответствующего действия. Подчеркнем, что число *sc-элементов*, блокируемых на время выполнения некоторого действия, в основном входят атомарные и неатомарные связки, и не должны входить *sc-узлы*, обозначающие бесконечные классы каких-либо сущностей, и, тем более, *sc-узлы*, обозначающие различные понятия (ключевые классы различных предметных областей).

Этичное (неэгоистичное) поведение *sc-агента*, касающееся блокировки *sc-элементов* (то есть ограничения к ним доступа другим *sc-агентам*) предполагает соблюдение следующих правил:

(1) Не следует блокировать больше *sc-элементов*, чем это необходимо, то есть не следует «жадничать»;

(2) Как только для какого-либо *sc-элемента* необходимость его блокировки отпадает до завершения выполнения соответствующего действия, этот *sc-элемент* желательно сразу деблокировать (снять блокировку);

Для того, чтобы *sc-агент* проверил возможность работы с каким-либо произвольным *sc-элементом*, он должен либо убедиться в том, что этот *sc-элемент* не входит во множество «полностью заблокированный *sc-элемент*», либо убедиться в том, что указанный *sc-элемент* входит в указанное множество, но при это связан отношением *полностью заблокированный sc-элемент** с действием, выполняемым этим *sc-агентом*. Очевидно, что больших затрат времени указанная проверка не потребует.

Особой группой полностью заблокированных *sc-элементов* (на время выполнения действия *sc-агентом*) являются вспомогательные *sc-элементы* («леса»), создаваемые только на время выполнения этого действия. Эти *sc-элементы* в конце

выполнения действия должны не деблокироваться, а удаляться.

Если действие в *sc-памяти*, выполняемое *sc-агентом*, завершилось (т.е. стало прошлой сущностью), то *sc-агент* оформляет результат («сухой остаток») этого действия, указывая (1) удаленные *sc-элементы* и (2) сгенерированные *sc-элементы*. Это необходимо, если нам придется сделать откат этого действия, т.е. возвратиться к состоянию базы знаний до выполнения указанного действия;

3.2. Средства спецификации блокировок в *sc-памяти*

Рассмотрим более подробно типологию и средства описания *структур*, заблокированных на время выполнения того или иного действия в *sc-памяти*.

Бинарное отношение *блокировка** связывает знаки действий в *sc-памяти* со знаками *структур* (ситуативных), которые содержат элементы, заблокированные на время выполнения данного действия или на какую-то часть этого периода. Каждая такая *структура* принадлежит какому-либо из типов блокировки.

Первым компонентом связок отношения *блокировка** является знак действия в *sc-памяти*, вторым – знак заблокированной *структуры*.

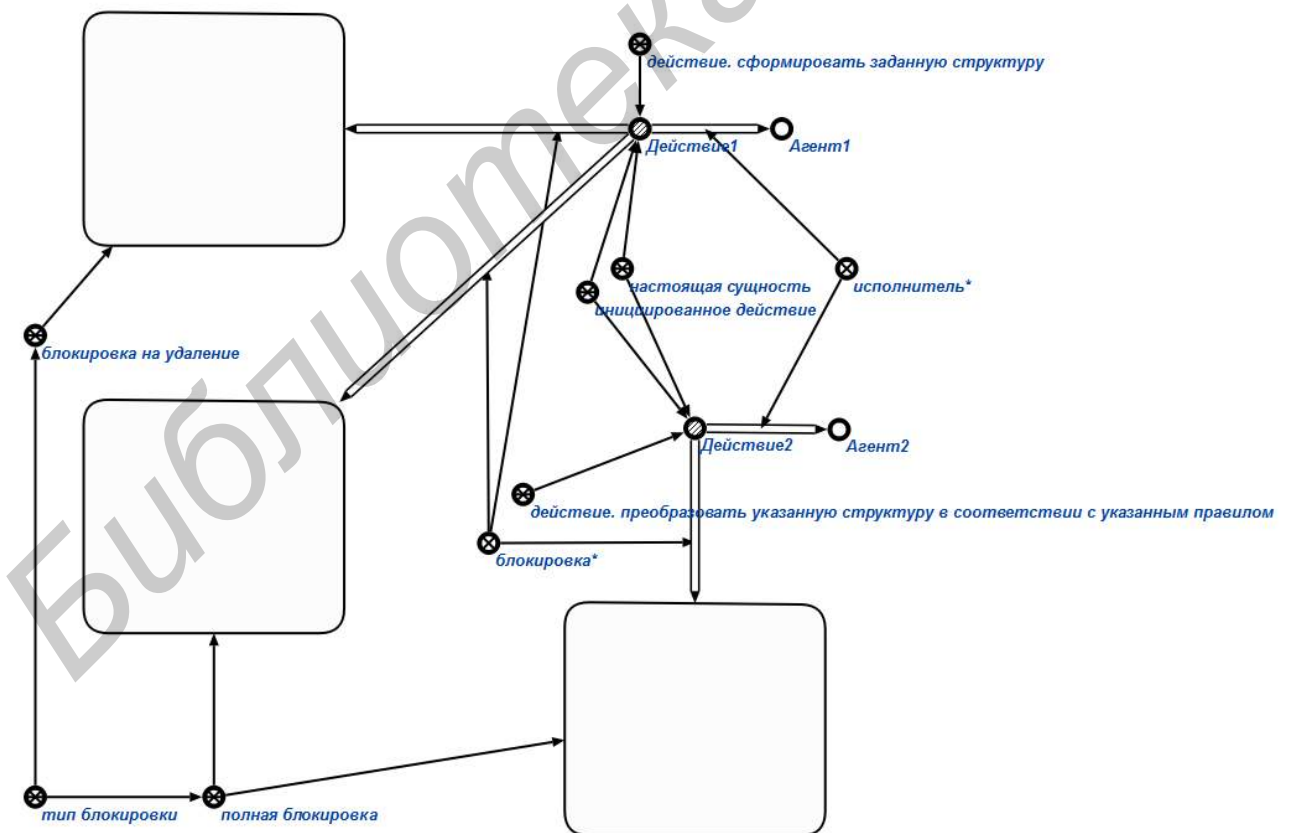


Рисунок 2 – Представление в *sc-памяти* заблокированных структур

Множество *тип блокировки* содержит все возможные классы блокировок, т.е. *структуры*, содержащие *sc-элементы*, заблокированные каким-либо *sc-агентом* на время выполнения им некоторого действия в *sc-памяти*.

тип блокировки

≡ полная блокировка

≡ блокировка на любое изменение

≡ блокировка на удаление

Каждая *структура*, принадлежащая множеству *полная блокировка* содержит *sc-элементы*, просмотр и изменение (удаление, добавление инцидентных *sc-коннекторов*, удаление самих *sc-элементов*, изменение содержимого в случае файла) которых запрещены всем *sc-агентам*, кроме собственно *sc-агента*, выполняющего соответствующее данной структуре действие в *sc-памяти*, связанное с ней отношением *блокировка**.

Для того, чтобы исключить возможность реализации *sc-агентов*, которые могут внести изменения в конструкции, описывающие блокировки других *sc-агентов*, все элементы этих конструкций, в том числе, сам знак *структуры*, содержащей заблокированные *sc-элементы* (принадлежащей как множеству *полная блокировка*, так и любому другому типу блокировки) и связи отношения *блокировка**, связывающие эту *структуру* и конкретное действие в *sc-памяти*, добавляются в *полную блокировку*, соответствующую данному действию в *sc-памяти*. Таким образом, каждой *полной блокировке* соответствует петля принадлежности, связывающая ее знак с самой собой.

Каждая *структура*, принадлежащая множеству *блокировка на любое изменение* содержит *sc-элементы*, изменение (физическое удаление, добавление инцидентных *sc-коннекторов*, физическое удаление самих *sc-элементов*, изменение содержимого в случае файла) которых запрещено всем *sc-агентам*, кроме собственно *sc-агента*, выполняющего соответствующее данной структуре действие в *sc-памяти*, связанное с ней отношением *блокировка**. Однако не запрещен просмотр (чтение) этих *sc-элементов* любым *sc-агентом*.

Каждая *структура*, принадлежащая множеству *блокировка на удаление* содержит *sc-элементы*, физическое удаление которых запрещено всем *sc-агентам*, кроме собственно *sc-агента*, выполняющего соответствующее данной структуре действие в *sc-памяти*, связанное с ней отношением *блокировка**. Однако не запрещен просмотр (чтение) этих *sc-элементов* любым *sc-агентом*, добавление инцидентных *sc-коннекторов*.

Заключение

В работе рассмотрена типология действий, выполняемых в *sc-памяти* агентами обработки знаний в рамках систем, управляемых знаниями,

описаны принципы взаимодействия таких агентов и средства синхронизации их деятельности.

Работа выполнена при поддержке гранта БРФФИ-РФФИ-М Ф15PM-074 «Методы и средства онтологического моделирования для семантических технологий проектирования интеллектуальных систем».

Библиографический список

- [IMS, 2016] Метасистема IMS.OSTIS [Электронный ресурс]. Минск, 2016. – Режим доступа: <http://ims.ostis.net/>. – Дата доступа: 15.01.2016.
- [Гаврилова и др., 2001] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник / Гаврилова Т.А. [и др.]; – СПб.: Изд-во «Питер», 2001.
- [Голенков и др., 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков [и др.] – Мн.: БГУИР, 2001.
- [Голенков, 2012] Голенков, В.В., Гулякина Н.А. Принципы Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования. – В кн Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2012). Материалы конф. [Минск, 16-18 февр. 2012 г.]. – Минск: БГУИР, 2012, с. 23-52.
- [Голенков, 2013] Голенков, В.В., Гулякина Н.А. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем./В.В. Голенков, Н.А. Гулякина// Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III междунар. науч.-техн.конф. – Минск: БГУИР, 2013 – с.55-78
- [Голенков, 2015] Голенков, В.В. Семантическая технология компонентного проектирования систем, управляемых знаниями. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар.научн.-техн.конф./ В. В. Голенков, Н.А Гулякина// Мн.: БГУИР, 2015
- [Непейвода, 2000] Непейвода Н.Н. Прикладная логика. Учебное пособие/ Непейвода Н.Н.; – Новосибирск.: ИГУ, 2000.
- [Поспелов, 1994] Поспелов Д.А. Информатика. Энциклопедический словарь. / Д.А.Поспелов; – М.: «Просвещение», 1994.
- [Рассел, Норвиг 2006] Рассел С., Норвиг П. Искусственный интеллект. Современный подход / Рассел С., Норвиг П.; - М.: Вильямс, 2006.
- [Тарасов, 2002] Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям / В.Б. Тарасов; – М.: Изд-во УРСС, 2002.
- [Финн, 2011] Финн, В.К. Искусственный интеллект. Методология, применение, философия / В.К. Финн. М.: Изд-во «Красанд», 2011.
- [Хоар, 1989] Хоар, Ч. Взаимодействующие последовательные процессы / Ч. Хоар; – М.: «Мир», 1989.
- [Шпаковский, 2013] Шпаковский Г.И. Коротко о параллельном программировании и аппаратуре / Г.И. Шпаковский; – Мн.: БГУ, 2013.
- [Шункевич, 2013] Шункевич, Д.В. Модели и средства компонентного проектирования машин обработки знаний на основе семантических сетей. – В кн Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2013). Материалы конф. [Минск, 2013 г.]. – Минск: БГУИР, 2013.
- [Шункевич, 2014] Шункевич, Д.В. Машина обработки знаний интеллектуальной метасистемы поддержки проектирования интеллектуальных систем. /Д.В. Шункевич// Открытые семантические технологии интеллектуальных систем (Ostis-2014): Материалы IV международной науч.-тех. конф -- Минск: БГУИР, 2014. – с.93-96
- [Шункевич, 2016] Шункевич, Д.В. Формальное семантическое описание целенаправленной деятельности различного вида субъектов / Д.В. Шункевич // Открытые семантические технологии интеллектуальных систем (Ostis-2016): Материалы VI международной науч.-тех. конф -- Минск: БГУИР, 2016.

ASYNCHRONOUS AND PARALLEL KNOWLEDGE PROCESSING INTERACTION IN COMMON SEMANTIC MEMORY

Shunkevich D.V.

*Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus*

shunkevichdv@gmail.com

The paper deals with the basic model of construction machinery knowledge processing systems, knowledge-driven, built on open semantic technology of intelligent systems design. Particularly, it describes the basic principles of asynchronous and parallel knowledge processing interaction in common semantic memory, typology of acts, agents run in this memory, typology and means of describing locks, necessary for synchronization of agents run on a common semantic memory.

Key words: semantic technologies, action formalisation, multi-agent systems, OSTIS project.

Introduction

The systems managed by knowledge, built on open semantic technology for intelligent systems, will be called *ostis-systems*. As a model of knowledge representation in this technology the unified semantic network with set-theoretic interpretation is used. This model of representation will be called SC-code (Semantic computer code). The elements of this semantic network will be called *sc-nodes* and *sc-connectors* (sc-arches, sc-rib).

Every *ostis-system* consists of platform-independent unified logical-semantic model of this system (*computer system sc-model*) and platform interpretation of such models. Every computer system *sc-model* might be decomposed on *sc-model of knowledge base*, *sc-model of knowledge processing machine*, *sc-model of interface*. In this work we will focus on *sc-model of knowledge processing machine*.

We list some of the advantages of the proposed approach to the construction of knowledge processing machines:

- Such as knowledge processing machine consists of agents, which exchange messages only through the common memory, the addition of a new agent or exclusion of one or some existing agents, doesn't mean any changes in other agents;
- Agents run parallel and independent of each other, performing different actions in *sc-memory*. Consequently, even a significant increase in the number of agents in the same system does not lead to a deterioration of its performance;
- Basic language for implementing software processing knowledge is the SCP language, and its texts are written by the same semantic networks as the processes information. At the same time every time you call the program created the unique scp-process. That gives a possibility to perform multiple operations

simultaneously;

- The transfer agent from one system to another is a simple transfer of a fragment of the knowledge base, without any additional operations that depend on the interpretation of the platform;

- The agent specifications and their programmes might be written at the same language with processed knowledge, which significantly reduces the list of special funds intended for knowledge processing machine design.

Conclusion

The paper considers the typology of actions performed in the *sc-memory* processing agent of knowledge within the system, manages knowledge, describes the principles of interaction between these agents and synchronization of their activities.

This work was supported by grant BRFFR-RFFR – Y F15 PM-074 "Methods and tools for the ontological modeling of semantic technologies for the intelligent systems design."



УДК 004.822:514

ПРИНЦИПЫ ПЛАТФОРМЕННОЙ НЕЗАВИСИМОСТИ И ПЛАТФОРМЕННОЙ РЕАЛИЗАЦИИ OSTIS

Ивашенко В.П., Татур М.М.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

**ivp@tut.by
tatur@i-proc.com**

В работе рассматривается подход к спецификации платформ, их сравнению и принципы их реализации, рассматриваются виды платформенной независимости и даются схемы построения платформенно независимых компонентов интеллектуальных систем, использующих в качестве языка представления знаний однородные семантические сети с теоретико-множественной интерпретацией.

Ключевые слова: семантические сети, интеллектуальные системы, платформенная независимость, абстрактные машины.

Введение

Для облегчения разработки, развития технологий разработки интеллектуальных систем и самих интеллектуальных систем важным свойством является возможность многократного использования компонентов интеллектуальных систем [Голенков и др, 2015]. Многократность использования разработанных компонентов означает, что сокращаются затраты на разработку нового компонента-аналога в случае необходимости построения более совершенной интеллектуальной системы. Кроме того, наличие в технологии разработки средств интеграции знаний [Ивашенко, 2015], позволяющих добавлять новые компоненты без остановки работы интеллектуальной системы, сводит процесс разработки новой более совершенной интеллектуальной системы к процессу обучения существующей. Таким образом, при наличии механизмов интеграции знаний в процессе работы системы и возможности многократного использования компонентов технология позволяет создавать совершенствуемые и обучаемые интеллектуальные системы. Необходимым свойством таких систем является сохранение основных потребительских свойств, включая производительность и потребление ресурсов.

Для реализации платформы необходима некоторая машина, поэтому при спецификации платформы будем сопоставлять ей некоторую абстрактную машину [Кузьмицкий В.М., 1998].

В общих чертах семейство реализуемых абстрактных машин можно рассмотреть в пространстве следующих признаков (рисунок 1): аппаратная или программная реализация, уровень языка платформы (способность представлять на нём данные сложной структуры и ассоциации), универсальность абстрактной машины (способность её моделировать другие машины) [Kazuhiro, 1984], [Hewitt, 2009].

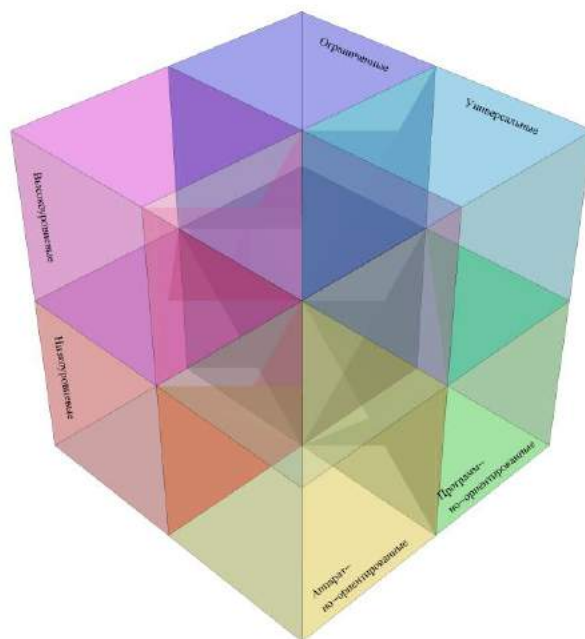


Рисунок 1 – Реализуемые платформы (абстрактные машины)

Далее будут рассмотрены более детально устройство и виды абстрактных машин, подходы к спецификации платформ, отношения между ними и принципы реализации интеллектуальных систем, имеющих платформенно-независимые компоненты.

1. Абстрактные машины

1.1. Память абстрактных машин

Каждая абстрактная машина соответствует формальной модели обработки информации [Голенков и др., 2001], задаваемой языком, начальной информационной конструкцией и множеством операций. Хранимые в памяти абстрактной машины информационные конструкции являются текстами некоторого языка (языка представления). Рассмотрим абстрактные машины, хранящие тексты языков, являющихся подмножествами A^* , замыкания замыканий алфавита A .

В зависимости от свойств языка абстрактные машины разделяются на классы:

- абстрактные машины, способные хранить тексты бесконечных языков, такие машины обладают неограниченной памятью, к таким машинам относятся стековые автоматы, машина Тьюринга, универсальные абстрактные машины;
- абстрактные машины, способные хранить только тексты конечного языка, к таким машинам относятся конечные автоматы;
- абстрактные машины, хранящие тексты симметричного языка A^{size} , где $size$ – количество ячеек памяти, память таких машин может рассматриваться как линейно организованная относительно A ;
- абстрактные машины с распределённой памятью, язык этих машин является ассоциацией (подмножеством декартова произведения) языков абстрактных машин каждого из её узлов,

1.2. Операции абстрактных машин

Операции (языки исполнения) абстрактной машины являются бинарными отношениями на множестве текстов языка абстрактной машины, ассоциациями языка представления абстрактной машины самого с собой.

В зависимости от свойств можно различать следующие виды операций:

- транзитивные операции,
- симметричные операции,
- детерминированные операции, которые являются функциями, частным случаем детерминированных операций являются алгебраические операции,
- недетерминированные операции.

К свойствам множества операций абстрактной машины можно отнести обратимость (наличие обратных операций), сходимости (наличие

аттракторов), взаимную исключительность (отсутствия операций с общими элементами) и независимость, когда ни одна операция не может быть представлена как объединение других операций или их композиций.

Отдельно выделяемым видом операций абстрактной машины являются операции ввода (сенсорные операции), которые позволяют изменить состояние, переходить к новому состоянию памяти абстрактной машины за счёт ввода данных из внешней среды.

2. Платформы

2.1. Виды платформ

Для описания платформ воспользуемся формальными моделями онтологий [Гаврилова и др., 2000], которые задаются тройкой конечных множеств

$$\langle G, R, O \rangle, \quad (1)$$

где G – конечное множество обозначений элементов объекта описания, R – строка отношений (языков структуры) на множестве G , O – строка функций (языков интерпретации) на множестве G .

Каждое состояние памяти абстрактной машины, соответствующей платформе, описывается формальной моделью его онтологии, поэтому для исследования операций абстрактной машины можно применить модель спецификации знаний [Ивашенко, 2015].

Модель спецификации знаний задаётся множеством формальных моделей онтологий Z , множеством морфизмов, соответствий между ними, и множеством отношений между парами формальных моделей и множеством соответствий и морфизмов.

$$\langle Z \cup 2^{\{\omega(z)|z \in Z\}^2}, 2^{\Theta((Z,Z))} \rangle; \quad (2)$$

$$\Theta(\langle X, Y \rangle) = \bigcup_{(x,y) \in X \times Y} (\{x\} \times \{y\}) \times 2^{\omega(x) \times \omega(y)},$$

где $2^{\Theta((Z,Z))}$ – множество отношений модели спецификации; ω – функция элементов онтологической модели

$$\omega(z_i) = G_i \cup \{r | r = R_{ij}\} \cup \{o | o = O_{ij}\} \cup \{k | k \in R_{ij}\} \cup \{p | p \in O_{ij}\} \cup \{a | \langle a, v \rangle \in O_{ij}\}. \quad (3)$$

Таким образом, каждой операции абстрактной машины платформы взаимно-однозначно соответствует отношение модели спецификации знаний, которое каждой паре формальных моделей онтологий (состояний абстрактной машины платформы) сопоставляет множество соответствий и морфизмов между такими онтологиями.

Платформа от абстрактной машины отличается тем, что для неё определены физические характеристики, включающие затраты памяти,

временные затраты, энергоинформационные затраты или потери и другие. Функции, вычисляющие эти затраты представимы в модели оценки качества, которая использует модель спецификации знаний.

Для вычисления мер в соответствии с выделенным языком L и записываемыми в его текстах требованиями используются функции вида

$$\left\langle \left\langle L, 2^{\Phi(\zeta(L), \zeta(L))} \right\rangle \middle| L \subseteq A^{**} \right\rangle, \quad (4)$$

где $\Theta(\langle Z, Z \rangle)$ – объединение отношений модели спецификации на Z ; ζ – отображение множества текстов языка спецификации на множество их онтологических моделей; Φ – упорядоченное множество оценок.

Таким образом, если определена некоторая детерминированная операция модели $\varphi \in M^M$ или недетерминированная $\psi \in 2^{M \times M}$, то существуют предикаты $\pi_{\Phi\varphi} \in 2^{\Phi(\zeta(M), \zeta(M))}$ и $\pi_{\Phi\psi} \in 2^{\Phi(\zeta(M), \zeta(M))}$ такие, что при $\Phi = \{0, 1\}$

$$\pi_{\Phi\varphi}(\langle \langle x, y \rangle, w \rangle) = \begin{cases} 1 & \exists u \exists v \exists x \exists y P(\langle u, v, x, y \rangle) \\ 0 & \neg \exists u \exists v \exists x \exists y P(\langle u, v, x, y \rangle) \end{cases}$$

$$P(\langle u, v, x, y \rangle) = (\langle x, y \rangle \in (\zeta(\{u\}) \times \zeta(\{v\}))) \wedge \wedge (w \subseteq \omega(x) \times \omega(y)) \wedge (\langle u, v \rangle \in \psi) \quad (5)$$

Затраты или потери в этом случае могут быть выражены:

$$\langle x, y \rangle \in (\zeta(\{u\}) \times \zeta(\{v\}))$$

$$\mu_{\Phi\psi}(\langle \langle x, y \rangle, w \rangle) = \pi_{\Phi\psi}(\langle \langle x, y \rangle, w \rangle) * \mu_{\Phi\psi}(\langle u, v \rangle), \quad (6)$$

где $\mu_{\Phi\psi}$ – функция затрат на множестве Φ' при переходе из состояния u в v .

Для сравнения операций необходимо перейти от формальных моделей онтологий, описывающих состояния, к формальным моделям онтологий, описывающим операции. Когда мощность операций бесконечна, такой переход не возможен без привязки формальной модели онтологии операции к формальным моделям онтологий состояний отношениями модели спецификации знаний. Поэтому для привязки используются три типа отношений модели спецификации знаний: отношения объединения пар формальных моделей онтологий состояний, соответствующих парам состояний операции, отношения реификации соответствия формальных моделей онтологий пар состояний операции и отношения проецирования формальных моделей онтологий реифицированного соответствия пар состояний операции на формальную модель онтологии операции. Тогда в модели оценки качества становится возможным дать оценку каждой паре формальных моделей онтологий операций.

Вид платформы может зависеть как от свойств её абстрактной машины, так и от её характеристик. Можно выделить два вида платформ: масштабируемые, немасштабируемые платформы.

Среди масштабируемых платформ можно различать масштабируемые по множествам затрат или потерь платформы. Среди масштабируемых по множествам затрат или потерь можно различать платформы, масштабируемые по множеству затрат или потерь по входу (входному состоянию), масштабируемые по множеству затрат или потерь по выходу (выходному состоянию), полностью масштабируемые по множеству затрат или потерь. Среди платформ, масштабируемых по множеству затрат или потерь по входу, выходу либо полностью можно выделять экспоненциально, полиномиально, квадратично, линейно, полилогарфмически, логарифмически масштабируемые (по множеству затрат или потерь по входу, выходу либо полностью (или немасштабируемые)) и другие (в соответствии с классом сложности или некоторым условием).

2.2. Модель платформ

Модель платформ рассматривает отношения между платформами. Одним из важных отношений является реализуемость одних платформ на базе других [Кузьмицкий В.М., 1998]. Реализуемость является транзитивным замыканием объединения отношений эмуляции, его обращения и отношения непосредственной реализации.

Непосредственная реализация заключается в переассоциации языка представления (абстрактной машины, виртуальной машины) платформы в язык абстрактной машины реализуемой платформы. Каждая операция (язык исполнения) реализуемой платформы должна быть сводима, (зависима, алгоритмически вычислима) к набору операций реализующей платформы. Переассоциация языков – отношение, являющееся объединением отношения ассоциации языков с композицией отношений объединения диссоциаций языков и ассоциации языков.

Отношение эмуляции платформ сводится к изоморфному вложению языка представления абстрактной машины эмулируемой платформы в язык эмулирующей платформы с точностью до неразличимости структуры элементов алфавита и изоморфному вложению всех операций (языков исполнения) эмулирующей машины в эмулируемую. При этом вложение языков исполнения осуществляется, таким образом, что для всех сопоставленных вложением аргументов их образы соответствующих операций взаимно-однозначно сопоставлены этим изоморфным вложением.

Редукция – отношение между множеством платформ и платформой, где множество языков реализующих платформ переассоциируемо в язык представления реализуемой платформы. Каждая операция реализуемой платформы должна быть алгоритмически сводима (зависима) к операциям

реализующей платформы. Платформы, не имеющие непосредственно реализующей платформы, могут рассматриваться как аппаратно-реализуемые [Ивашенко и др., 2015].

Характеристики всех непосредственно реализуемых платформ ограничиваются характеристиками этой платформы. Так, значения характеристик затрат (времени, памяти и т.п.) или потерь не возрастают для каждого двух пар соответствующих состояний реализуемой и реализующей платформы. При редукации характеристики затрат (памяти или времени) редуцируемой платформы меньше максимальных затрат (памяти или времени) одной из редуцирующих платформ. В случае параллельной редукации временные затраты реализуемой платформы могут быть меньше суммы временных затрат редуцирующих платформ. Поэтому для обеспечения требуемых характеристик реализуемой платформы каждая платформа каждого уровня (непосредственной) реализации должна гарантировать некоторые требования интересующих характеристик. Таким образом, каждый уровень реализации должен предоставлять, давать гарантии для обеспечения требований к реализуемой платформе. Как правило хорошими гарантиями затрат являются затраты менее квадратичных.

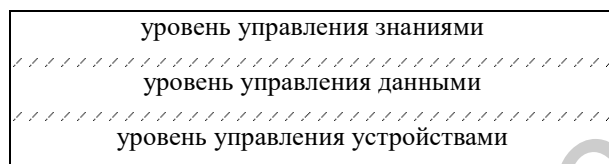


Рисунок 2 – Уровни управления (при включении языков представления друг в друга границы между уровнями могут быть неразделимы, что проиллюстрировано волнистой линией)

Относительно пары выделяемых языка представления данных и языка представления знаний выделяются уровни уровень управления устройствами, уровень управления данными, уровень управления знаниями. В уровень управления устройствами входят платформы, реализующие платформу языка представления данных, в уровень управления данными входят платформы, реализуемые на платформе с языком представления данных и реализующие платформу языка представления знаний, в уровень управления знаниями входят все платформы, реализуемые с использованием языка представления знаний (рисунок 2).

3. Платформенная независимость

3.1. Виды платформенной независимости

Платформа обладает платформенной независимостью, если существует реализующая её платформа, не реализующая её непосредственно и не реализующая её через другие непосредственно реализуемые платформы. Соответственно в зависимости от числа таких платформ платформа

может обладать большей или меньшей степенью платформенной независимости.

При платформенной независимости платформ с конечными языками представления обычно важным является различие простых характеристик (затрат времени или памяти при переходе между состояниями), выражаемое в виде отношения или разности соответствующих величин. Для платформ с бесконечными языками представления, как правило, важно сохранение (вида) масштабируемости платформы. В этом случае платформенная независимость различается по сохраняемому виду масштабируемости платформ.

3.2. Принципы построения платформенно независимых систем, управляемых знаниями

Структура системы, управляемой знаниями, в общем виде может быть иллюстрирована схемой (рисунок 3), включающей базу знаний, решатель и (пользовательский) интерфейс. Рассматриваемые системы ориентированы на обработку знаний представленных в SC-коде [Голенков и др, 2001], [Ивашенко, 2015].

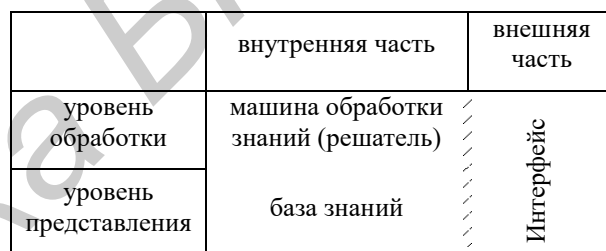


Рисунок 3 – Общая схема системы, управляемой знаниями

Более детализированная схема [Голенков и др, 2013] включает несколько уровней, основанных на агентно-ориентированном подходе (рисунок 4).

В соответствии с методикой компонентного проектирования систем, управляемых знаниями, выделяются следующие основные этапы [Шункевич и др., 2015b]:

- выбор и установка платформы реализации производной (дочерней) sc-системы, включая выбор и установку sc-хранилища и установку scr-машины;
- установка ядра базы знаний производной sc-системы;
- установка Ядра sc-машин, то есть набора базовых многократно используемых компонентов sc-машин, необходимых для работы даже первого прототипа sc-системы;
- установка Ядра sc-моделей интерфейсов, то есть набора базовых многократно используемых компонентов sc-моделей интерфейсов, необходимых для работы даже первого прототипа sc-системы;
- установка ядра подсистемы поддержки проектирования дочерней системы в составе проектируемой производной sc-системы.

К неосновным этапам относятся: расширение базы знаний производной sc-системы; расширение

машины обработки знаний производной sc-системы; расширение возможностей пользовательского интерфейса производной sc-системы.

Каждый агент (sc-агент [Голенков и др, 2015]) в реализации интеллектуальной системы может рассматриваться как некоторая платформа, поэтому к анализу его свойств применимы соответствующие подходы. К платформенно-независимым sc-агентам относятся sc-агенты, реализуемые на некоторой базовой sc-машине (scr-машине).

Коллектив интегрированных интеллектуальных систем	
Обрабатываемая база знаний интеллектуальной системы	
Система предметных областей, онтологий и формальных теорий	
Система семантически целостных фрагментов базы знаний	
Система взаимосвязанных sc-элементов	
Коллектив интегрированных решателей задач	
Решатель задач	
Знания решателя задач используемые для управления процессом обработки знаний на основании используемой модели решения задач	Коллектив неатомарных агентов над обрабатываемой базой знаний Коллектив атомарных агентов над обрабатываемой базой
Хранимые scr-программы, описывающие поведение атомарных агентов над базой знаний	Неатомарный агент интерпретации хранимых scr-программ Коллектив атомарных агентов интерпретации хранимых scr-программ
Библиотека scr-программ, описывающих базовые преобразования обрабатываемых знаний	
Техническая реализация sc-памяти	Техническая реализация атомарных агентов интерпретации хранимых scr-программ

Рисунок 4 – Уровни детализации интеллектуальной системы

Следует отметить, что в зависимости от реализации sc-хранилища [Шункевич и др., 2015a] и мощности соответствующего sc-языка [Ивашенко, 2015], возможны разные (частичные) реализации (версии) scr-машин, не каждая из которых реализуема на другой. Поэтому, говоря о платформенно-независимых компонентах (sc-агентах), следует уточнять о платформенной

независимости относительно какой реализации (версии) scr-машины идёт речь.

При платформенной реализации на её разных уровнях решаются разные задачи (рисунок 5).

В случае использования линейно адресуемой памяти к операциям над данными могут быть отнесены: операции над непрерывными участками памяти, содержащими данные (массивами, непрерывными функциями), операции над несвязными (прерывными) участками памяти с данными (списками, деревьями, функциями).

3	уровень управления знаниями	операции обработки знаний
2	уровень управления данными	операции над данными
1	уровень управления памятью (устройствами)	перевыделение участков памяти выделение и высвобождение участков памяти

Рисунок 5 – Задачи, решаемые на разных уровнях управления

Следует отметить, что важными принципами [] реализации операций обработки знаний (рисунок 5) для интеллектуальных систем, к которым относятся системы, управляемые знаниями относятся: принцип учёта НЕ-факторов, принцип семантического протоколирования процесса работы системы, принцип распределённой обработки потока данных (знаний).

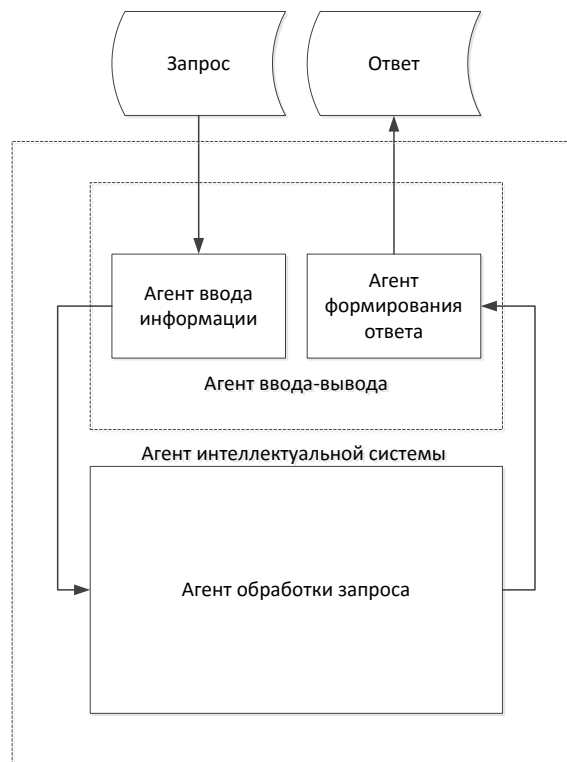


Рисунок 6 – Схема взаимодействия агентов в системе, управляемой знаниями

С точки зрения агентно-ориентированного подхода (в соответствии с рисунком 3) в упрощённом виде схема системы включает sc-агенты (рисунок 6): агент ввода-вывода (декомпозированный на агент ввода информации и агент формирования ответа (вывода)) и агент обработки запроса (решатель).

Агент ввода обеспечивает разбор входных данных (запроса) в соответствии с известной агенту грамматикой и формирует условия (команды) запуска агентов обработки запроса. Агент ввода (как и агент вывода) может редуцироваться к другим агентам ввода, если язык входных (выходных) данных является ассоциацией языков.

Агент вывода обеспечивает синтез и отправку ответа пользователя в соответствии с правилами синтеза текстов выходного языка.

Агент ввода-вывода может содержать другие редуцирующие агенты ввода-вывода помимо изображённых агентов ввода и вывода.

Агент обработки запроса (решатель) и его база знаний обеспечивает решение прикладной задачи в соответствии с некоторым планом решения этой задачи являющегося результатом логического вывода. Агент обработки запроса работает на уровне управления знаниями, который так же может быть разбит на уровни: уровень управления систематикой (первичных) элементов предметной области, уровень управления отношениями предметной области, уровень управления утверждениями о предметной области, уровень управления задачами предметной области и т.д. Для каждого уровня строится своя онтология (база знаний), обеспечивающая решение задачи управления на данном уровне.

Агент обработки запроса (решатель) в общем случае может содержать не только агент или агенты, ориентированные на решение прикладной задачи, но и агенты обеспечивающие реализацию основных принципов: агент интеграции знаний, учитывающий НЕ-факторы; агент семантического протоколирования процесса обработки знаний (работы системы); агент сборки мусора на основе анализа семантического протокола; агенты управления потоком знаний.

Примером системы, построенной с учётом перечисленных принципов, является система решения задач на графах, решающая задачи определения вида графа. Для этого решатель использует методы и алгоритмы канонической разметки графа, поиска мостов, точек сочленения, маршрутов и циклов в графе.

Заключение

В работе рассмотрены виды платформы, отношения между ними, понятие платформенной независимости и принципы реализации интеллектуальных систем в соответствии с технологией OSTIS.

Библиографический список

[Гаврилова и др., 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000.

[Голенков и др., 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков[и др.] – Мн. : БГУИР, 2001.

[Голенков и др., 2013] Голенков, В.В., Гулякина Н.А., Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем Материалы. Международной научн.-техн. Конференции OSTIS,2013:Минск, Республика Беларусь, БГУИР 21-23 февраля 2013.

[Голенков и др., 2015] Голенков, В.В., Гулякина Н.А., семантическая технология компонентного проектирования систем, управляемых знаниями Материалы Международной научн.-техн. Конференции OSTIS,2015:Минск, Республика Беларусь, БГУИР 19-21 февраля 2015.

[Ивашенко, 2015] Ивашенко, В.П. Модели и алгоритмы интеграции знаний на основе однородных семантических сетей. Материалы Международной научн.-техн. Конференции OSTIS,2015:Минск, Республика Беларусь, БГУИР 19-21 февраля 2015.

[Ивашенко и др., 2015] Ивашенко, В.П. Представление семантических сетей и алгоритмы их организации и семантической обработки на вычислительных системах с массовым параллелизмом. Материалы Международной научн.-техн. Конференции OSTIS,2015:Минск, Республика Беларусь, БГУИР 19-21 февраля 2015 / Ивашенко, В.П., Вереник Н.Л., Гирель А.И., Сейткулов Е.Н., Татур М.М. // Мн.: БГУИР, 2015.

[Кузьмицкий В.М., 1998] Кузьмицкий, В.М. Принципы построения графодинамического параллельного ассоциативного компьютера, ориентированного на переработку сложноструктурированных знаний / В.М. Кузьмицкий // Интеллектуальные системы : сб. науч. тр. / НАН Беларуси, Ин-т техн. кибернетики ; науч. ред. А.М. Крот. – Минск, 1998. – Вып. 1. – С. 156–166.

[Шункевич и др., 2015a] Шункевич, Д.В. Средства поддержки компонентного проектирования систем, управляемых знаниями. Материалы Международной научн.-техн. Конференции OSTIS,2015:Минск, Республика Беларусь, БГУИР 19-21 февраля 2015 / Шункевич Д.В., Давыденко И.Т., Корончик Д.Н., Жуков И.И., Паркалов А.В. // Мн.: БГУИР, 2015.

[Шункевич и др., 2015b] Шункевич, Д.В. Методика компонентного проектирования систем, управляемых знаниями. Материалы Международной научн.-техн. Конференции OSTIS,2015:Минск, Республика Беларусь, БГУИР 19-21 февраля 2015 / Шункевич Д.В., Давыденко И.Т., Корончик Д.Н., Губаревич А.В., Борискин А.С. // Мн.: БГУИР, 2015.

[Hewitt, 2009] Carl Hewitt Middle History of Logic Programming: Resolution, Planner, Prolog and the Japanese Fifth Generation Project ArXiv 2009. Mode of access: <http://arxiv.org/abs/0904.3036>. Date of access: 30.11.2015.

[Kazuhiro, 1984] Kazuhiro Fuchi, Revisiting Original Philosophy of Fifth Generation Computer Systems Project, FGCS 1984, pp. 1-2.

FOUNDATIONS OF PLATFORM INDEPENDENCY AND OSTIS PLATFORM IMPLEMENTATION

Ivashenko V.P., Tatur M.M.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

ivp@tut.by

tatur@i-proc.com

The paper deals with an approach to the platform specifications, platform comparison, principles of their implementation and types of platform independence. The framework of platform independent components of intelligent systems are described. This framework uses homogenous semantic network with set-theoretic interpretation as the language of knowledge representation.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8

ПРИНЦИПЫ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ SC-ХРАНИЛИЩА ПРОЕКТА OSTIS И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ПЛАТФОРМЕ .NET FRAMEWORK

Каешко А.И.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ondister@gmail.com

Рассмотрен вопрос практического использования программной реализации sc-памяти проекта OSTIS в связке с программным обеспечением, реализованным на языке C#. Рассмотрены команды sctp-протокола для сетевого взаимодействия с sc-хранилищем.

Ключевые слова: OSTIS, C#, .Net Framework, sctp, sc-хранилище.

Введение

За последние несколько лет программная реализация технологий проекта OSTIS шагнула далеко вперед. Программная реализация sc-хранилища вполне может конкурировать с такими продуктами как Neo4j и HyperGraphDB. Тем не менее, при подробном изучении проекта OSTIS становится очевидным недостаток информации, касающейся программной реализации, как sc-хранилища, так и внешних инструментов взаимодействия с ним. Более того, обилие довольно сложной для понимания человеком, незнакомым с OSTIS, информации, посвященной теоретическим основам проекта, делает порог вхождения еще выше.

Итак, проект OSTIS [OSTIS, 2015] предназначен для проектирования интеллектуальных информационных систем и их компонентов. При этом такие системы будут основаны на знаниях, представленных в виде семантической сети. Однако информационные системы, реализуемые на основе технологии OSTIS, могут быть и не совсем интеллектуальными, если не стоит такой задачи, а просто служить для накопления информации, формализованной сложной моделью.

Вне зависимости от назначения, процесс проектирования интеллектуальной системы на основе технологии OSTIS производится в два основных этапа [Корончик, 2013]:

- разработка логико-семантической модели системы (sc-модель);

- разработка программного интерпретатора этой модели;

Однако это не означает, что инициировав создание информационной системы на основе технологии OSTIS каждый раз необходимо создавать новый программный интерпретатор модели. Предполагается, что технология OSTIS, является компонентной, и компоненты, в том числе и программные, создаются сообществом OSTIS. Так по состоянию на декабрь 2015 года уже созданы программные реализации sc-хранилища [sc-machine, 2015], языка SCP [SCP, 2015], а так же набор программных средств, ориентированный на взаимодействие системы с внешней средой. Базовые компоненты: sc-хранилище и интерпретатор языка SCP реализованы на языке C для 32-х и 64-х битных версий операционных систем семейства Linux и 64-х битных систем Windows.

1. Sc-хранилище

Sc-хранилище это информационная подсистема, предназначенная для хранения sc-графов в виде семантической сети [Корончик, 2013]. В упрощенном понимании оно является аналогом поSql базы данных. Семантическая сеть в sc-хранилище строится из 3 базовых элементов: узла, коннектора и ссылки. Эти элементы и образуют конструкции, называемые sc-графами, которые составляют семантическую сеть. Каждый элемент хранилища имеет уникальный адрес, состоящий их сегмента (segment) и смещения (offset). Sc-графы вне хранилища можно описать посредством семейства sc-языков, например линейного (scs) или графического (scg). Существует также транслятор

scs и scg кода в sc-хранилище [sc-machine, 2015].

Однако sc-хранилище отвечает не только за организацию хранения sc-графов, оно также предоставляет программный интерфейс для добавления, удаления и извлечения хранимой информации. На программном уровне это возможно осуществить через динамическую библиотеку sc-памяти, которая управляет sc-хранилищем. Однако, учитывая, что эта динамическая библиотека реализована на языке C, подключить ее к таким высокоуровневым языкам программирования, как C# или Java довольно проблематично.

Для решения этой проблемы, а так же для организации сетевого взаимодействия был разработан протокол sctp (Semantic Code Transport Protocol). Тем самым открылись возможности для управления семантическим хранилищем с использованием практически любого языка программирования.

2. Sctp-протокол

Sctp-протокол является бинарным протоколом обмена данными между sctp-сервером и sctp-клиентами. Sctp-сервер использует библиотеку sc-памяти для выполнения команд, поступивших от клиента, и возвращает клиенту результаты выполнения этих команд. Сервер уже реализован на языке C++ и является программным компонентом проекта OSTIS. Каждая команда состоит из заголовка и аргументов. Сами команды довольно подробно описаны в документации sctp-протокола [sctp, 2015]. Предполагается, что протокол будет мультиверсионным, однако в настоящее время существует только одна его версия. Так же предполагалось, что sctp-сервер будет асинхронным, однако в дальнейшем от асинхронной работы пришлось отказаться и сейчас сервер работает только в синхронном режиме приема и передачи данных.

Итак, протокол содержит описание, а сервер предоставляет обработку следующих основных команд:

- Проверка существования элемента с указанным адресом;
- Создание новой дуги указанного типа с указанным начальным и конечным элементом;
- Создание новой ссылки;
- Создание нового узла указанного типа;
- Создание подписки на событие;
- Удаление элемента с указанным адресом;
- Удаление подписки на событие;
- Запрос всех произошедших событий;
- Поиск элемента (узла) по его системному идентификатору;
- Поиск всех ссылок с указанным содержимым;
- Получение начального и конечного элементов дуги;
- Получение содержимого ссылки по ее

адресу;

- Поиск конструкций;
- Поиск сложных конструкций;
- Установка содержимого ссылки;
- Установка системного идентификатора узла.

Как видно из перечня команд, sctp-протокол предоставляет широкие возможности по управлению sc-хранилищем, за исключением модели управления доступом. Поэтому при разработке реальных информационных систем работу с sc-хранилищем посредством sctp-протокола лучше оставлять полностью на серверной стороне, а общение сервера и клиента реализовать с использованием безопасных протоколов передачи данных. Так, например, реализован пользовательский интерфейс интеллектуальной метасистемы поддержки проектирования интеллектуальных систем [Корончик, 2014].

3. Реализация команд sctp-протокола на платформе .Net Framework

Изначально sctp-клиент реализовывался как часть проекта OSTIS.MED, посвященного созданию инструментария для создания медицинских информационных систем. И язык платформа .Net Framework была выбрана за ее универсальность и скорость написания кода.

Как уже указывалось ранее, sctp-протокол бинарный, поэтому основную сложность при реализации клиента на платформе .Net Framework составляли методы преобразования объектов в байтовые массивы.

Реализация клиента велась с полным следованием объектно-ориентированной парадигмы программирования. В результате было реализовано 2 библиотеки. Они не используют никаких сторонних библиотек и реализованы на платформе .Net Framework 4.5. Таким образом, они не совместимы с операционной системой Windows XP. Первая библиотека, ostis.sctp, создавалась, как низкоуровневая библиотека, и содержит только команды протокола [dotnet.sctp, 2015]. Она поддерживает как синхронный, так и асинхронный режим работы. Однако работа с библиотекой крайне неудобна и потребует большого количества повторяемого кода. Для увеличения скорости создания приложений и удобства написания кода была создана библиотека ostis.sctp.tools, которая зависима от первой, и имплементирует абстрактную базу знаний. Кэширование коллекций часто используемых элементов базы знаний позволяет ускорить доступ к ним в реальных приложениях. Кроме этого, она содержит так называемые неатомарные команды, которые отсутствуют в описании протокола, например получение системного идентификатора узла по его адресу. Безусловно, быстродействие второй библиотеки ниже, чем базовой, однако в реальных приложениях часто важнее оптимизация цикла создания кода, чем быстродействие конечного продукта.

В процессе разработки клиента было выполнено полное документирование кода, Все команды снабжены примерами работы, а особо сложные, для лучшего понимания снабжены иллюстрациями. Так же были разработаны модульные тесты, объем кода, покрытого тестами, составил 85,6%. Работоспособность библиотеки была протестирована на операционных системах Windows (7, 8.1, 10 64 bit), Linux (Ubuntu 14.2 32 bit с фреймворком Mono).

Таким образом, удалось создать библиотеку для работы с sc-памятью на языке высокого уровня с явной типизацией, поддерживающего объектно-ориентированную парадигму программирования.

Сами библиотеки довольно хорошо документированы, поэтому подробное описание их архитектуры в рамках статьи излишне.

4. Принципы и примеры использования sctp.Net клиента

Работа с базовой библиотекой клиента происходит в несколько этапов:

- соединение с сокетом сервера;
- подготовка команды;
- отправка команды;
- получение ответа;
- работа с ответом от сервера.

Ниже представлен код метода для создания узла в базе знаний с использованием базовой библиотеки:

```
public void TestCreateNodeSync()
{
    const string defaultAddress =
SctpProtocol.TestServerIp;
    string serverAddress = defaultAddress;
    int serverPort =
SctpProtocol.DefaultPortNumber;
    sctpClient = new SctpClient(serverAddress,
serverPort);
    sctpClient.Connect();
    if (sctpClient.IsConnected == true)
    {
        var commandCreate = new
CreateNodeCommand(ElementType.ConstantNode_c);
        var responseCreate =
(CreateNodeResponse)sctpClient.Send(commandCreate
);
    }
}
```

При такой работе необходимо отслеживать доступность сокета и наличие активного соединения, если необходимо выполнить несколько команд одновременно. Более того, необходимо обрабатывать ошибки.

Использование библиотеки ostis.sctp.tools позволяет сократить объем кода и обработать ошибки:

```
public void TestNodeTools()
```

```
{
    KnowledgeBase kb = new
KnowledgeBase(Sctp.SctpProtocol.TestServerIp,
Sctp.SctpProtocol.DefaultPortNumber);
    Node node = new
Node(Sctp.ElementType.ConstantNode_c);
    kb.Nodes.Add(node);
}
```

Более того, при использовании этой библиотеки можно использовать несколько баз знаний одновременно. В библиотеке второго уровня предусмотрено кэширование часто используемых элементов, что позволяет сократить время доступа к ним. Обычно в базе знаний существует множество неизменяемых отношений. На момент написания статьи библиотека ostis.sctp.tools реализована не полностью. Достаточно полно реализованы действия с узлами, а с дугами и ссылками работать можно только через свойство Commands класса базы знаний, который имплементирует все команды библиотеки ostis.sctp и содержит несколько сборных команд.

Несмотря на все достоинства при реальном использовании клиентских библиотек выявлен ряд недостатков. В первую очередь, это особенности работы с событиями и не достаточно проработанная обработка ошибок. Так, запрос произошедших событий не всегда возвращает все события. Кроме того, при реальной работе приходится производить множество обращений к базе знаний для выполнения достаточно простых операций. Поэтому производительность реальных приложений оставляет желать лучшего. Скорее всего, на данном технологическом этапе придется создавать гибридные системы, где будут использоваться несколько типов хранилищ данных, включая реляционные базы.

Заключение

Для применения технологии OSTIS в студенческих, магистерских и аспирантских проектах уже разработаны sctp-клиенты для языков Python, Java, C++. Однако степень их проработки, использованные парадигмы программирования различаются. Созданный нами клиент для sctp-сервера на языке C# может быть использован для создания как учебных, так и реальных приложений на различных платформах и языках программирования, которые поддерживает .Net Framework. Благодаря хорошей документации проект может быть использован людьми, слабо знакомыми и с технологией OSTIS и с языком C#. А sc-хранилище может быть использовано не только для реализации интеллектуальных систем, но и в проектах, где требуется гибкое и высокоэффективное семантическое хранилище, в том числе и в гибридных системах.

Библиографический список

[OSTIS, 2015] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2015. - Режим доступа: <http://ostis.net>. – Дата доступа: 15.12.2015.

[Корончик, 2013] Реализация хранилища унифицированных семантических сетей. – В кн Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2013). Материалы конф. [Минск, 21-23 февр. 2013 г.]. – Минск: БГУИР, 2013, с. 125-128.

[Корончик, 2014] Пользовательский интерфейс интеллектуальной метасистемы поддержки проектирования интеллектуальных систем. – В кн Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2014). Материалы конф. [Минск, 20-22 февр. 2014 г.]. – Минск: БГУИР, 2014, с. 79-82.

[Голенков, 2012] Графодинамические модели параллельной обработки знаний / В. В. Голенков, Н. А. Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2012

[sctp, 2015] Протокол sctp [Электронный ресурс]. – 2015 – Режим доступа: <https://github.com/deniskoronchik/sc-machine/wiki/sctp> – Дата доступа: 02.12.2015

[sc-machine, 2015] sc-память [Электронный ресурс]. – 2015 – Режим доступа: <https://github.com/deniskoronchik> – Дата доступа: 02.12.2015

[SCP, 2015] Интерпретатор SCP [Электронный ресурс]. – 2015 – Режим доступа: <https://github.com/ShunkevichDV/sc-machine> – Дата доступа: 02.12.2015

[dotnet.sctp, 2015] Реализация протокола sctp на платформе .Net Framework [Электронный ресурс]. – 2015 – Режим доступа: https://github.com/ondister/dotnet_sctp_client – Дата доступа: 07.12.2015

PRINCIPLES AND SOFTWARE IMPLEMENTATION OF INTERACTION OSTIS SC-MEMORY AND .NET FRAMEWORK PLATFORM

Kayeshko A.I.

Belorussian state university of informatics and radioelectronics, Minsk, Republic of Belarus

ondister@gmail.com

The article about of the practical use of a software implementation, OSTIS sc-memory in conjunction with the software is implemented in C #. Shows sctp-protocol usage for networking cooperation with sc-storage.

Introduction

Over the past few years, the software implementation of the OSTIS project technology has leaped forward. Software implementation sc-store can compete with no SQL databases such as Neo4j and Hyper Graph DB. However, the detailed study of the project OSTIS becomes apparent lack of information regarding the software implementation, as the sc-storage and external tools to interact with it. Moreover, the abundance of pretty difficult to understand a person unfamiliar with OSTIS, information dedicated to the theoretical foundations of the project, does the threshold of entering higher.

Main Part

Sc-store this is information subsystem for storing sc-graphs in a semantic network form. However, sc-store is responsible not only for the organization of storage sc-graphs, it also provides a programming interface to add, delete, and retrieve stored information.

For the organization of networking interface was developed sctp-protocol (Semantic Code Transport Protocol). Thus opening the possibility for managing semantic repository with almost any programming language.

Sctp-protocol is a binary protocol communication between the sctp- server and sctp-clients. Sctp-server uses the library sc-memory to execute commands received from the client, and returns to client results of those commands. Server has already been implemented in C ++ and is a software component of the OSTIS project. We have implemented FTP client in C # language.

Conclusion

Due to to the good documentation of the project can be used by people poorly familiar with the technologies both the OSTIS and the C #. A sc-storage can be used not only for the implementation of intelligent systems, but also in projects requiring highly flexible and semantic storage, including hybrid systems.



УДК 004.822:514

«УМНЫЙ ДОМ», УПРАВЛЯЕМЫЙ ЗНАНИЯМИ

Корончик Д. Н.

** Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь
denis.koronchik@gmail.com*

В работе приводится описание подхода к реализации «Умного дома» на основе технологии OSTIS, который дает целый ряд преимуществ, в сравнении с существующими подходами. Суть описываемого подхода заключается в том, что устройства в доме управляются знаниями.

Ключевые слова: умный дом, интернет вещей, виртуальная модель, управление знаниями

Введение

Количество устройств подключенных к интернету растет большими темпами. По некоторым оценкам количество таких устройств в 2014 году составило около 8 миллиардов, а к 2018 году их количество превысит 18 миллиардов [forecastjoy, 2014]. В связи с этим все чаще стала упоминаться концепция «интернет вещей» (ИВ) [wp, 2015]. Суть данной концепции заключается в том, что все устройства (физические или виртуальные объекты) связаны между собой в одну большую вычислительную сеть, в которой они решают различные задачи, взаимодействуя друг с другом. Работу в этом направлении ведут некоторые компании. К примеру, компания Google инициировала открытый проект «Physical Web» [Physical Web, 2014].

Одним из способов реализации «Интернета вещей» является подход на основе агентов и базы знаний [hb, 2015]. Технология OSTIS изначально ориентирована на обработку знаний в базе знаний с помощью агентов, что в свою очередь позволяет реализовать описанный в [hb, 2015] подход к построению «Интернета вещей».

Частным видом интернета вещей может выступать «Умный дом». В данной статье будут рассмотрены наработки проекта OSTIS, которые могут быть использованы для реализации «Интернета вещей» на примере реализации виртуальной модели «Умного дома».

1. Существующие подходы

Существующие подходы к построению «Умного дома» имеют целый ряд недостатков:

- сложность встраивания новых устройств в инфраструктуру «Умного дома». Проблема заключается в том, что нет единого стандарта, а также подключение устройств может отнимать много времени и средств (так как зачастую это требует специализированных навыков);
- зачастую технологии, созданные для создания «Умных домов», позволяют решать узкий спектр задач (заложенных в них разработчиками). Основную роль при решении задач выполняет человек;
- расширение функционала «Умного дома», сводится к программированию необходимых паттернов поведения, что требует достаточного уровня знаний (хотя есть проекты в которых это сведено к минимуму, за счет использования визуальных языков программирования [majordomo, 2015]).

2. Архитектура

Как и любая система, построенная с использованием технологии OSTIS, «Умный дом» состоит из 3-х основных подсистем: базы знаний, машины обработки знаний и пользовательского интерфейса.

В базе знаний состоит из следующих разделов:

- описание моделей устройств (предметная область / онтология устройств). В данном разделе описываются различные устройства и их модели (указывается производитель, класс устройства, потребляемая мощность и другие его характеристики);
- структура дома. В данном разделе указываются устройства, которые относятся к конкретному дому, их модели, положение в доме и

т. д. Благодаря тому, что системе известны модели устройств и их полное описание, у нее появляется возможность отвечать на различные вопросы. Примеры возможных вопросов:

- каким образом можно подключить одно устройство к другому;
- как пользоваться теми или иными функциями устройства;
- когда необходимо провести техническое обслуживание устройства;
- и т. д.

Показатели жизнедеятельности дома. В данном разделе присутствуют показания с различных датчиков дома и состояние устройств.

Кроме перечисленных выше в базе знаний имеются разделы описывающие машину обработки знаний и пользовательский интерфейс.

Машина обработки знаний «Умного дома» состоит из множества агентов, которые в основном являются частью устройств. Схема взаимодействия устройств с базой знаний показана на рисунке 1.



Рисунок 1 - Схема взаимодействия устройств с базой знаний

Чтобы показать каким образом происходит управление устройствами через базу знаний, рассмотрим простой пример: включение (выключение) устройства. Каждое устройство имеет два агента. Один из этих агентов отвечает за включение устройства и условием его инициирования является добавление устройства во множество включенных устройств (рисунок 2).

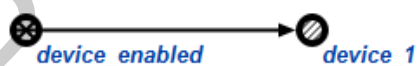


Рисунок 2 - Пример включенного устройства

Второй агент отвечает за выключение устройства. Его условие инициирования - это удаление устройства из множества включенных устройств.

Разные устройства содержат разный набор таких агентов. К примеру, телевизор может содержать агентов, которые отслеживают уровень звука,

текущую программу, текущий источник сигнала и т. д. Агенты могут быть реализованы как аппаратным так программным способами.

Помимо агентов, которые отвечают за управление устройствами, машина обработки знаний содержит в себе целый набор агентов, которые отвечают за решение задач и ответа на вопросы пользователя. Примерами таких агентов могут быть:

- агент поиска рецепта блюда по ингредиентам, которые имеются в холодильнике;
- определения способа подключения одного устройства к другому;
- получение прогноза погоды из интернета;
- и т. д.

Пользовательский интерфейс для системы «Умного дома» является достаточно сложной частью с точки зрения реализации. Основным способом взаимодействия пользователя с системой является речевой интерфейс. После распознавания команды, в базе знаний формируется команда или вопрос с аргументами, на что реагирует соответствующий агент, которые выполняет данную команду. Пример команды включения устройства представлен на рисунке 3.

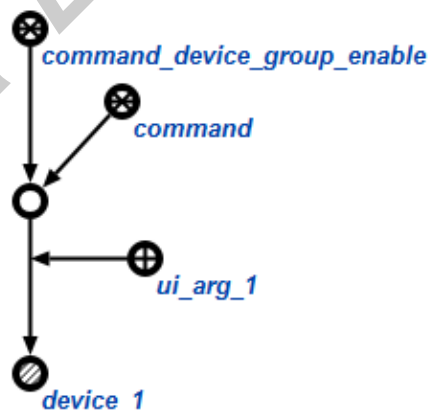


Рисунок 3 - Пример команды включения устройства

В результате инициирования команды (путем добавления её во множество инициированных команд), устройство `device_1` будет добавлено во множество включенных устройств (`device_enabled`). Это приведет в действие агента, который отвечает за включение устройства и оно включится. На этом работа пользовательского интерфейса не завершена. Чтобы дать обратную связь пользователю, для каждой команды заданы текстовые шаблоны ответов, которые используются для синтеза речи (рисунок 4).

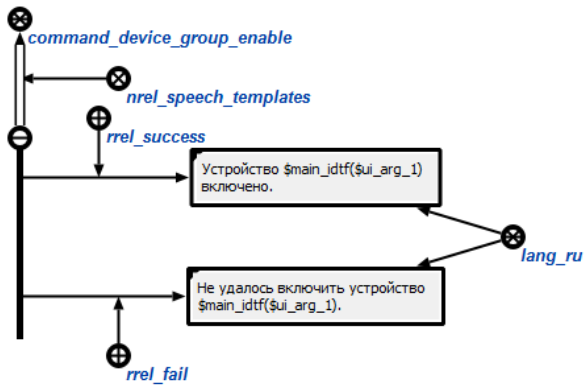


Рисунок 4 - Пример текстовых шаблонов

Для каждой завершенной команды (вопроса) инициируется команда генерации текстового ответа (рисунок 5). Где первым аргументом является команда, на результат которой необходимо сгенерировать текст. Второй аргумент указывает на язык, который необходимо использовать для генерации текста. Третий аргумент указывает на атрибут, который необходимо использовать для выбора шаблона (если команда завершена успешно, то *rrel_success*, иначе *rrel_fail*).

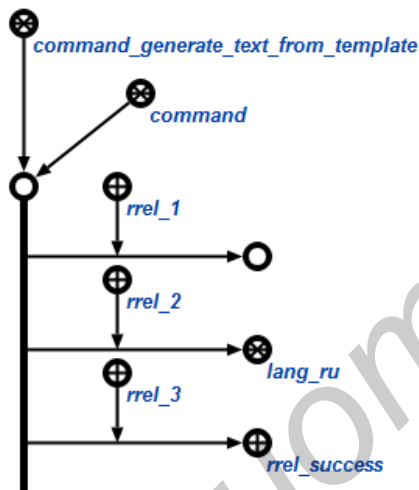


Рисунок 5 - Пример команды генерации текста по шаблону

В результате успешного выполнения команды, генерируется sc-ссылка, содержимым которого является сгенерированный текст. Этот текст подается на вход генератору речи, которая проигрывается пользователю.

Помимо речевого интерфейса могут быть реализованы и другие интерфейсы, например графический или какие-то другие.

3. Реализация

Реализация прототипа данной системы была осуществлена на языке программирования C++, с использованием Unreal Engine 4 [ue, 2015] в качестве программной платформы, в которой была реализована виртуальная модель дома с устройствами (рисунок 6).



Рисунок 6 - Реализованная виртуальная модель дома

Для взаимодействия виртуальных устройств с базой знаний был реализован сетевой клиент на C++. Помимо виртуальной модели дома было реализовано приложение для Android, на Java. Оно реализует речевой ввод и вывод. Общая схема взаимодействия между реализованными компонентами и приложениями представлена на рисунке 7.

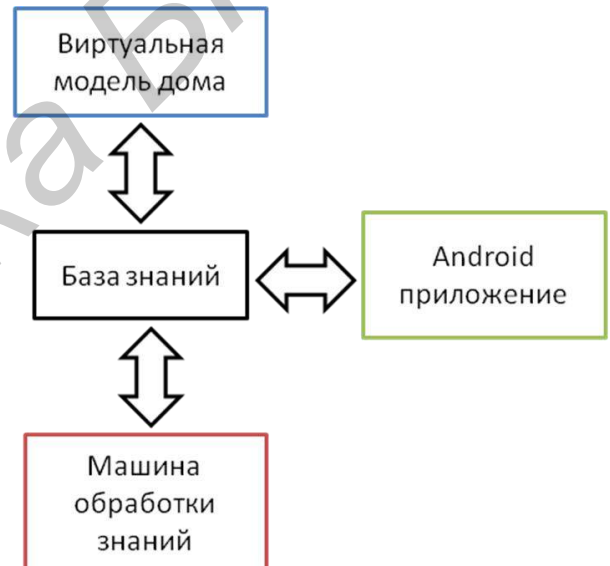


Рисунок 7 - Схема взаимодействия реализованных приложений

Распознавание голосовых команд реализовано с помощью стандартных средств Android. Распознанный текст обрабатывается с помощью сервиса api.ai [api.ai, 2015]. В результате обработки запроса, данный сервис возвращает JSON следующего вида:

```
{
  "id": "2f3bc87a-a875-423f-a5c9-aeb597a55ff9",
  "timestamp": "2016-01-28T21:09:52.053Z",
  "result": {
    "source": "agent",
    "resolvedQuery": "включи телевизор",
    "action": "ui_menu_iot_device_group_enable",
    "actionIncomplete": false,
    "parameters": {
      "ui_arg_1": "group_volume_tv"
    }
  },
  "contexts": [],
}
```

```

"metadata": {
  "intentId": "cff78e8d-dbf8-47a4-ba23-1a5e739d17f3",
  "intentName": "включить"
},
"fulfillment": {
  "speech": "Включаю"
}
},
"status": {
  "code": 200,
  "errorType": "success"
}
}

```

На основе полей *action* и *parameters*, генерируется команда в базе знаний и добавляется во множество инициированных команд. Далее происходит ожидание, пока инициированная команда будет завершена, после чего генерируется и иницируется команда генерации текстового ответа по шаблону. На основе результата этой команды происходит синтез речи (с помощью стандартных средств Android).

В настоящий момент в модели реализованы следующие виды устройств: лампочка, телевизор, холодильник. Каждое из этих устройств имеет агенты, которые отвечают за их включение и выключение.

Заключение

В результате работы был реализован прототип «Умного дома», который управляется базой знаний. Имеющаяся реализация позволяет расширять функционал виртуальной модели, за счет добавления новых агентов для решения задач, за счет добавление новых устройств. В дальнейшем, виртуальные устройства могут быть заменены на реальные, а имеющаяся система может работать на небольшом компьютере (Raspberry Pi) и управлять целым домом.

Реализованный подход дает некоторые преимущества:

- благодаря базе знаний, система имеет представление, о том какие возможности есть у дома в целом. Это позволяет решать задачи управления домом более эффективно. Помимо того, в будущем, онтология устройств может быть общей для всего мира и пополняться производителями как wikipedia;
- так как система построена на основе агентов, то это значительно упрощает интеграцию устройства в среду «Умного дома». В конечном итоге это сводится лишь к подключению питания к нему и автоматической регистрации устройства.

Библиографический список

[forecastjoy, 2014] Forecast joy [Электронный ресурс]. – 2014 – Режим доступа: <http://forecastjoy.com/tag/internet-of-things/> – Дата доступа: 02.12.2015

[Physical Web, 2014] Сайт проекта Physical Web [Электронный ресурс]. – 2014 – Режим доступа: <http://google.github.io/physical-web/> – Дата доступа: 30.11.2015

[wp, 2015] Статья wikipedia: Интернет вещей [Электронный ресурс]. – 2014 – Режим доступа: https://ru.wikipedia.org/wiki/Интернет_вещей – Дата доступа: 20.11.2015

[hb, 2015] Статья: Умный интернет вещей - кто он и с чем его едят [Электронный ресурс]. - 2015 - Режим доступа: <http://habrahabr.ru/post/259243/> - Дата доступа: 20.11.2015

[majordomo, 2015] MajorDoMo - умный дом и домашняя автоматика своими руками [Электронный ресурс]. - 2015 - Режим доступа: <http://majordomo.smartliving.ru/Main/HomePage> - Дата доступа: 10.11.2015

[ue, 2015] Unreal Engine Technology [Электронный ресурс]. - 2015 - Режим доступа: <https://www.unrealengine.com> - Дата доступа: 20.11.2015

[api.ai, 2015] Free Speech-to-text, Intent Recognition and NLP for IoT [Электронный ресурс]. -2015 - Режим доступа: <https://api.ai/> - Дата доступа: 22.11.2015

SMART HOME, THAT CONTROLS BY KNOWLEDGES

Koronchik D. N

**Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

denis.koronchik@gmail.com

Article proof of concept of smart home, based on knowledge base and implemented as agent-based. It contains information about simple agents workflow and principles of communication between them.

Introduction

Smart home is a small part of internet of things. There are no any technology that allows to create really intelligent systems to control home. Current state of existing technologies allows just to automate some home functionality. In this work made attempt to implement smart home model based on OSTIS technology. That allows to resolve some problems.

Main Part

Implemented model of smart home controlled by knowledge base. That means, that each device has set of agents, that start working by emitting events from knowledge base. Knowledge base contains information about each device, so system knows all available functionality and use it. This approach allows system to resolve more complex tasks, like an answer to question "What I can cook from food, that I have in my refrigerator?" or "Where I can to see news?" and etc.

Conclusion

Implemented model (with Unreal Engine 4, C++) shows, that described approach gives more opportunities, then existing technologies, to make smart home more intelligent. Implemented virtual devices can be replaced by real devices. Also speech interface was implemented by using Android and Java. It allows to run commands by speech.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004. 832.2

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АВТОМАТИЧЕСКОГО РЕШЕНИЯ ЗАДАЧ ПО ПЛАНИМЕТРИИ

Курбатов С.С., Лобзин А.П., Хахалин Г.К.

Научно-исследовательский центр электронной вычислительной техники, Москва, Россия

curbatow.serg@yandex.ru

lobzin@rambler.ru

gkhakhalin@yandex.ru

В статье рассмотрена система автоматического решения задач по планиметрии с использованием онтологии и естественно-языкового интерфейса. В рамках системы разработано программное обеспечение, использующее онтологию при поиске решения задачи, сформулированной на предметно-ориентированном естественном языке. Для найденного решения разработана визуализация на предметном и онтологическом уровнях. Программное обеспечение включает макросы Word для работы с графикой и решатель, реализованный в программной среде онтологии.

Ключевые слова: решение геометрических задач; лингвистическая поддержка; прикладная онтология.

Введение

Исторически геометрия явилась провозвестником аксиоматического метода, нашедшего ныне широкое распространение не только в математике, но и в ряде других наук - физике, химии, биологии, лингвистике. Свою роль в осмыслении таких фундаментальных понятий как аксиоматика, формальная система, непротиворечивость, модель, интерпретация, геометрия не утратила и на современном этапе. Это обусловлено большой наглядностью геометрических построений, обеспечивающей возможность опираться на интуицию. Последняя не может заменить доказательство, но существенно облегчает его понимание. В современных условиях, учитывая графические возможности ЭВМ, геометрия приобретает дополнительную значимость. Проектируются графические системы обучения геометрии и решения задач, существенно повышающие качество образования [Давыденко и др., 2011].

Интерес к автоматическому решению задач и доказательству теорем, относящихся к геометрии, возник уже на раннем этапе исследований в области искусственного интеллекта (ИИ). В ранних работах был четко сформулирован отказ от полного перебора в пользу использования эвристических методов [Геллернтер, 1967]. В дальнейшем работы этого плана не получили достаточного развития и внимание было уделено представлению знаний,

экспертным системам, обработке естественного языка и т.п. В данной работе описывается программное обеспечение (ПО) системы автоматического решения задач по планиметрии с использованием онтологии и естественно-языкового интерфейса. Интеграция лингвистических методов, возможностей онтологии для представления знаний предметной области и современной графики позволяет по-новому взглянуть на роль геометрии в исследованиях по искусственному интеллекту и на возможности компьютерного обучения.

Эвристики, используемые при разработке данного ПО, базируются на более высоком концептуальном уровне, чем применяемые в [Геллернтер, 1967]. Такой уровень обеспечивается возможностями онтологии и средствами обращения к синтезируемому чертежу. Элементы высокоуровневых эвристик, конкретизируемых в данной предметной области (планиметрия), намечены в [Курбатов, 2010]. Высокий уровень эвристик позволяет не только существенно сократить перебор, но и предоставляет возможность пояснить процесс получения решения. Средства взаимодействия ПО с онтологией позволяют в диалоге обосновать выбор некоторой эвристики и графически отобразить по шагам путь к найденному решению. Отчасти эти средства реализуют концепцию решения задач, развитую в классической работе [Пойя, 1970], в которой описан процесс выдвижения эмпирических догадок, используемых в дальнейшем для строгого доказательства.

1. Структура системы

Система автоматического решения задач по планиметрии с использованием онтологии и естественно-языкового интерфейса представляет собой специализацию общей системы, в комплексе интегрирующей возможности анализа/синтеза естественного языка и анализа/синтеза изображений на базе онтологии [Khakhalin et al, 2012]. Блок-схема системы приведена на рис.1.

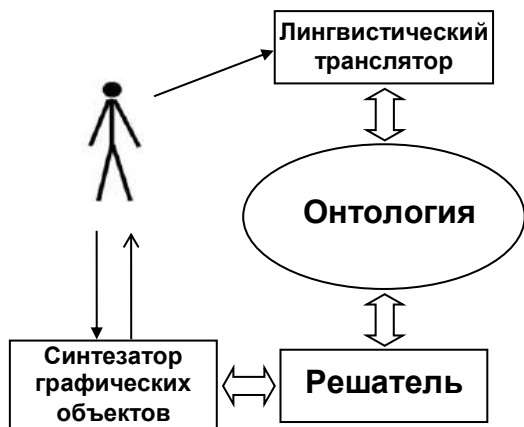


Рисунок 1 – Блок-схема системы

Пользователь формулирует планиметрическую задачу на предметно-ориентированном естественном языке (ЕЯ). Лингвистический транслятор переводит эту формулировку в онтологическую структуру, в которой специальным образом помечены известные элементы (дано) и неизвестные (требуется найти; эти концепты помечаются вопросами).

Задача решателя - с помощью базовых операций означить неизвестные элементы. Для этого он расширяет входную структуру, используя аксиоматику предметной области, эвристики и базовые операции преобразования с целью снятия вопросов. В случае успешной работы решателя формируется финальная расширенная онтологическая структура решения задачи, которая графически отображается пользователю.

Задача синтезатора графических объектов – визуализировать решение, что позволяет пользователю наглядно видеть весь процесс решения задачи и его графический результат с возможностью пошагового отображения.

Задача лингвистического транслятора - по ЕЯ-тексту планиметрической задачи построить концептуальную структуру описания ситуации в терминах понятий и отношений прикладной онтологии, которую решатель использует для поиска ответа с помощью выполнения базовых операций.

Обработка ЕЯ-текстов планиметрических задач основана на лингвистической концепции перифразирования [Апресян и др., 1989], адаптированной для поставленных в работе целей.

С помощью правил перифразирования предметно-ориентированный текст переводится в канонические структуры, непосредственно отображаемые в онтологию.

Работу реализованного алгоритма лингвистического анализа в текущей версии удобно описать на примере конкретной задачи в рассматриваемой области. Более подробно работа лингвистического транслятора описана в [Курбатов и др., 2015]

Формулировка задачи: *Построить окружность, проходящую через две заданные точки, с центром на заданной прямой.*

Опустим этапы морфологического и синтаксического анализа (хотя они и не являются тривиальными). В результате выполнения соответствующих правил перифразирования текст задачи представляется в виде следующего (канонического) описания:

Построить произвольную точку А. Построить произвольную точку В.

Построить произвольную прямую. Построить окружность.

Точка А принадлежит окружности. Точка В принадлежит окружности.

Центр окружности принадлежит прямой.

Текст канонического описания непосредственно отображается в структуру онтологии, где помечены как "известны" (или "дано") *точка А, точка В и прямая. Точка С и окружность* помечены как "неизвестны" (или "найти"). Графическое представление этой онтологической структуры дано на рис. 2.

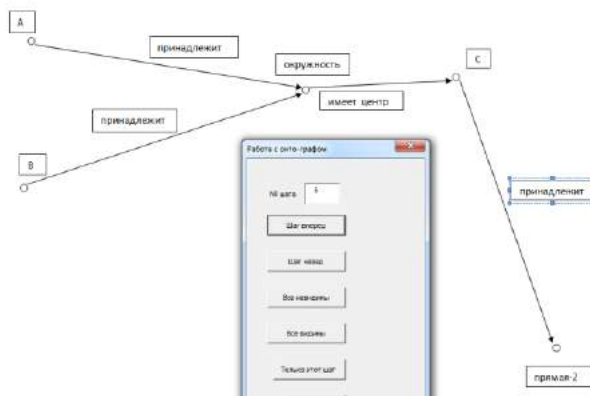


Рисунок 2 – Онтологическая структура задачи

После успешной лингвистической трансляции в онтологической структуре фиксируются объекты и отношения, определяющие условия задачи и требуемый результат. Онтологический решатель должен найти последовательность базовых операций, позволяющих построить требуемый объект (*точку, окружность, прямую* и т.д.). В качестве базовых операций используются "создать точку" (*отрезок, прямую, окружность*), "найти середину отрезка", "провести перпендикуляр из точки на отрезке прямой", "опустить перпендикуляр из точки на прямую" и т.п.

Набор базовых операций не фиксирован и может пополняться. Кроме того, базовые операции могут быть заблокированы при поиске результата. Это позволяет искать решение в условиях ограничений типа "использовать только линейку и окружность с заданным центром (без циркуля)". Базовые функции реализованы с помощью макросов Word (графические функции) и для их вызова онтология формирует формальный текст, включающий имена функций и имена и значения соответствующих параметров. Стиль вызова отличается от используемого в [Литвинович, 2012], поскольку интерпретация текста макроса выполняется только внутри документа. Примеры графических функций приведены в Таблице 1.

Таблица 1 – Примеры графических функций

Crt_tch (x, y, name, rnd)	Построить имитацию точки с координатами x и y и именем name*.
Crt_otr (P1, P2, rnd)	Построить отрезок с концевыми точками P1 и P2.
Crt_line (... , rnd)	Построить прямую **.
Crt_cyrc (P1, L1, rnd)	Построить окружность по центру(P1) и радиусу(L1).
Rel_line (L1, L2)	Определить отношение прямых L1 и L2.
Peres_line (L1, L2, P)	Построить точку с именем P, являющуюся точкой пересечения прямых L1 и L2.
Rel_LC (L, C)	Определить отношение прямой L и окружности C.
Peres_LC (L, C, P1, P2)	Построить точки пересечения прямой и окружности P1 и P2.
Seredina (L, P)	Построить точку P середины отрезка L.
Perpen (L1, P, L2)	Построить перпендикуляр L2 к отрезку L1 в точке P.
Sravn_otr (L1, L2)	Сравнить отрезки L1 и L2 ***.
Perpen_line (P1, L, PL)	Опустить перпендикуляр из точки P1 на прямую L. Фиксировать результат в виде точки или отрезка PL.
Ctr_ugol (P1, P2, P3, U, rnd)	Построить угол U по трем точкам. P2 - вершина угла.
Crt_biss (U, L)	Построить биссектрису L угла U.
Crt_treug (... , rnd)	Построить треугольник.
Crt_paral (... , rnd)	Построить параллелограмм.
Crt_trap (... , rnd)	Построить трапецию.
Vidim/nevidim (...)	Определить видимость/невидимость графического объекта.

*) Параметр rnd определяет случайное задание графического объекта (точка, прямая, треугольник, трапеция и т.п.).

***) По двум точкам, по принадлежащему ей отрезку или по точке и коэффициенту наклона.

****) Отрезки могут быть равны (конгруэнтны) или могут быть в отношении больше/меньше.

2. Онтология

В онтологии задаются знания о естественном языке (синтаксис, семантика языка, правила перефразирования, а также знания об отображении ЕЯ-описаний в формальные конструкции предметной области), знания о предметной области (концепты, связи, аксиоматика, эвристики) и знания о графике.

2.1. Общее описание онтологии

Нижний уровень онтологии представлен базовыми структурами, задающими структурированную семантическую сеть. Системные отношения сети определяют вложенность фрагментов сети, а также связи "род-вид" и "часть-целое". Внесистемные отношения определяются конкретикой предметной области.

Онтология реализована на инструментальной СУБД Progress и использует ее возможности для интерфейса с онтологией.

Интерфейс с онтологией обеспечивает возможности текстового ввода/вывода онтологических структур и их редактирования. Более высокий уровень работы с онтологией позволяет, в частности, использовать гиперграфы и автоматизировать ввод правил перефразирования с помощью ЕЯ-интерфейса. Полное текстовое описание базовых онтологических структур дано в [Лобзин, 2015].

2.2. Концепты, отношения, поиск решения

В текущей версии системы рассматриваются планиметрические задачи на построение. В качестве концептов выступают базовые понятия (точка, прямая, плоскость, окружность) и производные понятия (отрезок, луч, угол, треугольник и т.д.). В качестве отношений используются отношения принадлежность, пересечение, параллельность, перпендикулярность и т.д. Введение декартовых координат позволяет задавать аксиоматику концептов и отношений в алгебраическом виде. Например, точка задается парой чисел (x, y), прямая - двумя точками (x1, y1) - (x2, y2), а аксиоматика точка A (x, y) принадлежит прямой, если и только если

$$x = x_1 = x_2 \text{ (для вертикальной прямой),}$$

$$y = y_1 = y_2 \text{ (для горизонтальной прямой)}$$

и, наконец, $y = k \cdot x + b$, где $k = (y_2 - y_1) / (x_2 - x_1)$, $b = y_1 - (y_2 - y_1) / (x_2 - x_1) \cdot x_1$.

Для отрезка (x1, y1) - (x2, y2) аксиоматика записывается так: точка A (x, y) принадлежит горизонтальному отрезку, если и только если точка A (x, y) принадлежит прямой (x1, y1) - (x2, y2) и $x_1 \leq x \leq x_2$ при $x_1 < x_2$ или $x_2 \leq x \leq x_1$ при $x_1 > x_2$

$> x_2$ (для не вертикального отрезка). Аналогично описание для вертикального отрезка. В онтологии аксиоматика зафиксирована декларативно и поддерживается программно. В частности, известно, что прямые могут *совпадать*, *быть параллельны* или *пересекаться*. Соответствующая программная поддержка позволяет определить какое из данных отношений имеет место для конкретных прямых.

Онтология обращается к чертежу для выполнения графических операций типа "создать точку", "провести прямую", "создать окружность", "создать отрезок" и т.п. Дополнительно онтология может запросить по чертежу эмпирические данные (равенство углов или отрезков, перпендикулярность или параллельность и т.д.) и использовать их для сокращения перебора при поиске решения. Стиль такого взаимодействия с чертежом описан в [Курбатов, 2010].

Поиск решения использует эвристики онтологии для сокращения перебора. В полученной после лингвистической трансляции структуре одни объекты отмечены, как <известны>, а другие как <НЕ-известны>. Задача решателя - путем ввода новых объектов/отношений и использования аксиоматики предметной области добиться, чтобы требуемые в задаче объекты получили статус <известны>. В процессе перебора строится дерево, вершины которого соответствуют текущей онтологической структуре (и соответствующему чертежу). Каждой вершине дерева приписан вес, определяющий <перспективность> вершины.

Перебор всегда ведется с вершины, обладающей максимальной перспективностью. Алгоритм вычисления перспективности зависит от эвристических соображений типа <аналогичность>, <наличие объектов со статусом - известны>, <расстояние до корня дерева> и т.п. В текущей реализации перспективность рассчитывается по формуле: $k \cdot (\sum v_i) - r$, где k - коэффициент базовой операции, v_i - объекты со статусом <известны>, а r - расстояние до корня дерева. Коэффициент k отражает полезность использования базовой операции и в первом приближении определяется статистически.

Подчеркнем, что это именно первое приближение, дерево перебора также организовано как онтологическая структура и в принципе может использовать любые знания из базы.

В рассматриваемых задачах на построение для большинства базовых операций результат выполнения непосредственно отображается на чертеже. Это позволяет наглядно представить не только успешно полученный результат, но и неудачные попытки решения. Отметим, что онтологически-ориентированный перебор существенно абстрагирован от концептуальной структуры. Например, поиск на "и/или" графах является более конкретным, информация о том, что вершины в онтологической структуре связаны отношением "и/или", может быть известно только в

онтологии. В принципе возможен случай, что отношения имеет тип "и/или", но какой именно - неизвестно. В задачи решателя онтологии входит выбор действий по определению типа.

2.3. Интерфейс с онтологией

В текущей реализации ПО интерфейс использует достаточно общий метод взаимодействия с онтологией через текстовые файлы. Интерпретация этих файлов выполняется отдельно на уровне графических функций и в программной среде онтологии. При этом онтология выступает в роли системы, принимающей решение, графические функции - в роли выполняющей (эффеторной) системы. В текстовом файле онтология формирует имена и параметры для вызова базовых функций, интерпретация практически сводится к вызову (*call* функции с соответствующими аргументами). Соответствие ЕЯ-наименований базовых функций и объектов визуализации, а также концептуальное описание чертежа содержатся в онтологии. Пример взаимодействия онтологии и графических функций приведен далее.

Целесообразная организация интерфейса сложна не только в технологическом, но и в идейном плане. Интерфейс должен учитывать баланс между такими факторами как эффективность выполнения, удобство восприятия, гибкость настройки на другую программную среду и т.д. Гибкость настройки требует особого внимания, как отмечено выше, в используемой онтологической структуре неизвестные концепты специальным образом отмечены. Квалифицированный подход к гибкости предполагает, что переход от используемой онтологической структуры (и соответствующей аксиоматики) к RDF-триплетам должен быть относительно простым.

Это означает, что будет возможно использовать ресурсы глобальной семантической сети (*semantic web*), в частности, представлять онтологическую структуру задачи в виде запроса на SPARQL. В этом случае неизвестные концепты будут помечены знаком вопроса "?", а сама структура должна быть описана на языке OWL. Разумеется, и аксиоматика задачи должна быть соответствующим образом описана. Вопрос о возможностях такого представления и эффективности использования логического вывода в OWL является предметом специального исследования.

3. Визуализация

3.1. Графические объекты и базовые операции

Для графических объектов онтологии реализовано более 30 базовых операций типа: "создать произвольную прямую", "построить отрезок", "найти точку пересечения прямых", "построить из точки касательную к окружности" и т.д. (см. Таблицу 1.). Отрезки и окружности имеют

внутренние имена (*line x, oval y*), но для диалога важны ЕЯ-наименования таких объектов. Поэтому в число базовых включен и ряд сервисных функций, в частности, функция ЕЯ-наименования отрезков и окружностей (фиксируется в свойстве AlternativeText соответствующего объекта).

К сервисным функциям относятся также раскраска объектов, их перемещение по экрану (или вращение), изменение свойства Visible = msoFalse (или msoTrue) и т.п. Изменение последнего свойства позволяет сделать невидимым указанный объект, что повышает наглядность при сложном чертеже. Важной сервисной функцией является определение выделенных пользователем объектов на чертеже (Selection), что дает возможность организовать диалог с онтологией относительно этих объектов.

Базовые функции, включающие слово "произвольный" (произвольная прямая, окружность, треугольник и т.д.), используют случайный выбор при генерации свойств соответствующих объектов. Это позволяет разнообразить чертеж для одной и той же задачи, а также (в перспективе) формировать в онтологии предположения, используя эмпирические данные чертежа.

3.2. Онтологическая и предметная визуализация

В процессе онтологически-ориентированного перебора выполняется ряд попыток. Эти попытки в онтологической структуре отображаются путем ввода новых экземпляров концептов (*точек, прямых, отрезков, окружностей* и т.п.) и отношений между ними. Визуализацию этой структуры будем называть онтологической, она представляет интерес в основном для инженера по знаниям (или администратора онтологии). Для пользователя важна визуализация найденного решения в терминах базовых операций.

Первоначально на чертеже для нашей задачи с помощью базовых операций типа *создать точку* и *создать прямую*, построены только "две точки (А и В)" и "прямая-1".

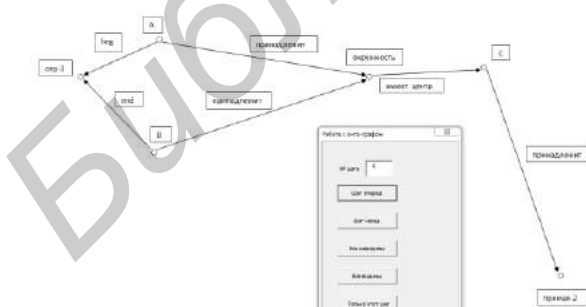


Рисунок 3 – Онтологическая структура после шага 5 (отрезок АВ).

Онтологическая структура, соответствующая промежуточному шагу решения задачи представлена на рис. 3.

Полная концептуальная структура решения этой задачи приведена на рис. 4, где А, В, М, С -

экземпляры концепта *точка*; отр-1,2,3,4 - экземпляры концепта *отрезок*; beg и end - отношения между точкой и отрезком - *является началом* и *является концом*.

Эта структура соответствует онтологической визуализации. С помощью специализированных базовых операций администратор онтологии может проследить по шагам процесс формирования структуры и эвристические основания при выборе каждого шага. Средства фильтрации онтологии позволяют выделить нужный фрагмент структуры и существенно уменьшить громоздкость визуализации.

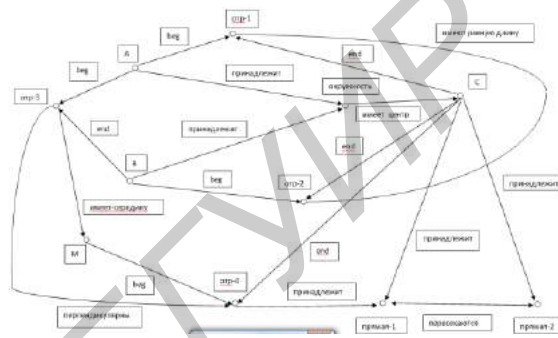


Рисунок 4 – Концептуальная структура решения

Предметная визуализация представляет собой динамически формируемый чертеж, соответствующий выполнению текущей базовой операции. Для нашей задачи базовыми операциями, приводящими к решению, являются:

- построить отрезок А-В;
- найти середину отрезка А-В;
- построить прямую, перпендикулярную отрезку А-В и проходящую через его середину (прямая-2);
- найти точку пересечения прямых прямая-2 и прямая-1 (точка-С);
- построить окружность с центром в точке (точка-С) и радиусом (отрезок А- точка-С).

В процессе применения всех базовых операций найденного решения чертеж модифицируется. Графический результат, соответствующий решению, может иметь вид, приведенный на рис. 5. Именно этот чертеж с комментариями выдается после автоматического нахождения решения.

В дополнение к визуализации вместе с чертежом выводится файл с протоколом выполнения базовых операций. Этот файл имеет вид, практически совпадающий с ЕЯ-описаниями, но возможно дополненный координатами точек, длинами отрезков и т.д., а также эвристическими комментариями по выбору базовой операции.

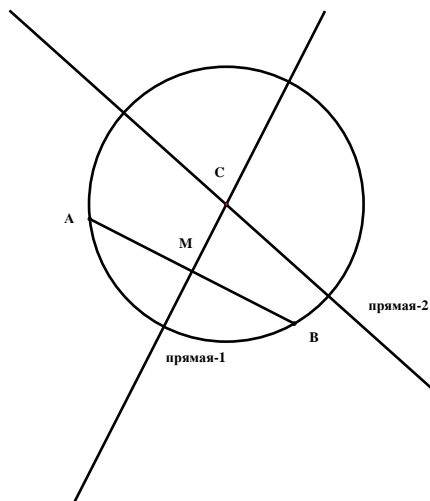


Рисунок 5 – Графический вид решения задачи на построение.

Заключение

В данной работе описывается программное обеспечение, использующее онтологию:

- при лингвистической трансляции условий задачи;
- при автоматическом поиске онтологически-ориентированного решения задачи;
- при демонстрации решения на графическом уровне: онтологический граф и визуальное представление графическими объектами.

Эксперименты подтвердили работоспособность разработанного программного обеспечения, ряд наиболее интересных технических результатов приведен в [Лобзин, 2015].

В прикладном аспекте данное ПО целесообразно развивать для его использования в образовательном процессе. Компьютерная реализация намеченной Пойя концепции [Пойя, 1970] обещает существенно улучшить качество обучения. Эта концепция предполагает формулировку учеником правдоподобных рассуждений и выдвижение догадок и именно на этот процесс сделан акцент в реализованном ПО.

Работа выполнена при поддержке гранта РФФИ 15-07-03847 А «Интерпретация технически-ориентированных текстов на естественном языке с помощью прикладных онтологий».

Библиографический список

- [Khakhalin at al., 2012] Khakhalin G., Kurbatov S., Naidenova K., Lobzin A. Integration of the Image and NL-text Analysis/Synthesis Systems. In book: "Intelligent Data Analysis for Real-Life Applications: Theory and Practice" (Ed. Rafael Magdalena at all). – USA: IGI Global, 2012, p. 160-185.
- [Курбатов и др., 2015] Курбатов С.С., Лобзин А.П., Хахалин Г.К., Лингвистическая поддержка системы решения задач по планиметрии // XX международная научно-практическая конференция "Инновации в науке: применение и результаты". Новосибирск: M&S, 2015, с. 17-23.
- [Апресян и др., 1989] Апресян Ю.Д., Лингвистическое обеспечение системы ЭТАП-2 / Ю.Д. Апресян, И.М. Богуславский, Л.Л. Иомдин и др. – М.: Наука, 1989.

[Давыденко и др., 2011] Давыденко И.Т., Интеллектуальная справочная система по геометрии, OSTIS-2011, / И.Т. Давыденко [и др.]; – Минск. БГУИР, 2011.

[Курбатов, 2010] Курбатов С.С., Высокоуровневые эвристики для автоматизированного формирования базы знаний, / С.С. Курбатов С.С., // 12 национальная конференция по искусственному интеллекту с международным участием КИИ-2010, Труды конференции, М.: Физматлит, Том 2, с. 231-239.

[Геленгер, 1967] Геленгер Г., Реализация машины, доказывающей геометрические теоремы, Г. Геленгер, // Вычислительные машины и мышление, - М.: Мир, 1967, с. 145-165.

[Пойя, 1970] Пойя Д., Математическое открытие, Д. Пойя, - М.: Наука, 1970.

[Лобзин, 2015] http://www.eia--dostup.ru/onto_geom.htm.

[Литвинович, 2012], Литвинович А.В., Язык описания графических объектов GRASP, Литвинович А.В., // Нейрокомпьютеры / №10 за 2012 г., - М.: Радиотехника.

SOFTWARE FOR THE AUTOMATIC SOLUTION OF TASKS ON PLANE GEOMETRY

Curbatow S.S., Lobzin A.P., Khakhalin G.K.

*Research Centre of Electronic Computing
Engineering, Moscow, Russia*

curbatow.serg@yandex.ru

lobzin@rambler.ru

gkhakhalin@yandex.ru

The article describes a system for automatic solution of tasks on plane geometry with the use of ontologies and natural language interfaces

Introduction

This article describes software for automatic solution of tasks on plane geometry with the use of ontologies and natural language interfaces. Integration of linguistic methods, capabilities of the ontology to represent domain knowledge and modern graphics give a fresh look at the role of geometry in research on AI and computer learning.

Main Part

General information about the used ontology tool is provided. The concepts and relationships of the subject area (geometry) are described. For example, specific tasks are demonstrated: the interface with the ontology, the capabilities of the solver. Visualization at the subject and ontological levels is developed for the search solution.

Conclusion

Experiments have confirmed efficiency of the developed software. In applied aspect the software should be developed for use in the educational process.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

ОБ ОНТОЛОГИИ ИНТЕЛЛЕКТУАЛЬНОГО МОДЕЛИРОВАНИЯ

Валькман Ю.Р., Степашко П.В.

*Международный научно-учебный центр информационных технологий и систем НАН и МОН
Украины, г.Киев*

yur@valkman.kiev.ua

pstepashko@gmail.com

В настоящее время объекты и системы моделирования непрерывно усложняются. Поэтому, весьма актуальна разработка средств интеллектуализации процессов моделирования. С нашей точки зрения, ядром системы интеллектуального моделирования является онтологическая база знаний. В статье рассматриваются некоторые принципы построения онтологии моделей, процессов моделирования, моделируемых объектов, решаемых проблем, задач

Ключевые слова: моделирование; онтология; интеллектуальная система; инфраструктура.

Введение

В настоящее время разрабатываются все более сложные объекты и технологии. Без моделирования создание таких систем невозможно. Поэтому, постоянно создаются новые методы моделирования, технологии практического применения которых также усложняются. Модели становятся все более многомерными, и сложноструктурированными. Без построения мультимодельных структур для анализа совместимости, сбалансированности различных комплексов моделей проектирование и исследование таких объектов невозможно.

Специалисты должны хорошо знать специфику своих предметных и проблемных областей, а не особенности технологий компьютерного моделирования. А без использования компьютерных систем в настоящее время, часто, моделирование невозможно. Мы должны накапливать положительный и отрицательный опыт построения и использования моделей различных процессов и объектов. Поэтому разработка методов и технологии интеллектуального моделирования представляются весьма актуальными.

Здесь *объектом исследования* являются процессы синтеза и анализа моделей (и их структур - отношений) сложных систем. *Предмет исследования* – технологии моделирования сложных объектов, процессов. Здесь, *цель исследования* – разработка принципов построения онтологии для разработки интеллектуальных компьютерных технологий моделирования.

1. Некоторые определения и ограничения

Одним из основных исследовательских видов деятельности человека является моделирование, т.е. способ изучения закономерностей изменения структуры различных процессов или явлений под воздействием различных факторов. В основе процесса моделирования лежит понятие модели.

Модель конструируется *субъектом* исследования (или управления) так, чтобы отобразить характеристики объекта (свойства, взаимосвязи, структурные и функциональные параметры и т.п.), существенные для целей исследования.

Поэтому вопрос о качестве такого отображения — адекватности модели объекту — правомерно решать *только относительно определенной цели*.

Модель — всегда искусственно созданный объект, который будучи адекватным исследуемому объекту, отображает и воспроизводит в более простом виде структуру, связи и взаимосвязи между элементами объекта, непосредственное изучение которого связано с какими-либо трудностями, и тем самым облегчает процесс получения информации об интересующем нас объекте.

Конструирование модели на основе предварительного изучения объекта и выделение его существенных характеристик, экспериментальный и (или) теоретический анализ модели, сопоставление результатов с данными об объекте, корректировка модели и т.д. составляют содержание моделирования.

Моделирование – это процесс адекватного отображения наиболее существенных сторон исследуемого объекта с той точностью, которая необходима для решения рассматриваемой задачи.

Введем следующее, удобное нам (для построения онтологии), определение модели [Перегудов и др., 1989].

Модель есть целевое отображение оригинала: абстрактное или реальное; познавательное или прагматическое; статическое или динамическое; конечное, упрощенное, приближенное; имеющее наряду с безусловно истинным, условно истинное и безусловно ложное содержание, ингерентное; адекватное; появляющееся и развивающееся в процессе практического использования.

Далее мы детально рассмотрим эти, базовые для нас, свойства моделей. Именно они (вместе с рассмотренными выше) будут классификаторами в онтологиях моделей и моделирования.

Или, более формально (детальнее см. [Валькман и др., 2012b]).

ОПРЕДЕЛЕНИЕ 1. Под параметром P_i будем понимать формальную характеристику исследуемых свойств исследуемого/проектируемого сложного объекта.

По сути, любой параметр – это модель свойства или характеристики моделируемого объекта.

ОПРЕДЕЛЕНИЕ 2. Моделью M_j будем называть любое отношение между параметрами (свойствами, характеристиками) сложного изделия, представляющее интерес с точки зрения исследования или проектирования.

Таким образом, любая модель формально это:

$$M_j: R(P_{1j}, P_{2j}, \dots, P_{ij}),$$

где: j — индекс (идентификатор) модели,

P_{ij} — i -тый параметр j -той модели,

R — отношение между параметрами.

Отношение R может быть представлено посредством множества нотаций (таблица решений, полином, дифференциальное уравнение, измерительная матрица, система продукций, фреймы, семантические сети и т.д.)

В [2] показана целесообразность и эффективность использования категорий лингвистики «текст» и «контекст» в формальных аппаратах представления знаний проектировщиков и исследователей сложных систем.

ОПРЕДЕЛЕНИЕ 3. Текстом M_T модели M будем называть формальную запись отношения R , отображаемого в модели.

Представляется вполне правомерным определить и категорию контекста модели. И к этому аспекту понятия модели отнести мотивировку ее корректности, описание постановки задачи, используемого формального аппарата и прочие

аспекты, которые не нашли свое отражение в тексте модели, но необходимы для адекватной ее интерпретации.

ОПРЕДЕЛЕНИЕ 4. Контекстом M_K модели M будем называть формализованные условия адекватности модели моделируемому объекту.

Часто, контекст модели определяется посредством областей допустимых значений параметров, но есть и «неявные контексты». Это – значения тех параметров, которые не представлены в тексте моделей. Далее мы будем широко использовать эти понятия.

Рассматриваем мы только знаковые модели (см., например, в [Валькман, 2012a]). Это – всегда абстрактные модели, имеющие символическое представление. Частным случаем таких моделей являются математические, т.е. представленные в форматах математической нотации.

Исчислением символических моделей занимаются специалисты искусственного интеллекта (см. гипотезу «символических моделей» Ньюэла-Саймона, например, в [Саймон, 1972]).

К интеллектуальным технологиям, таким образом, мы приходим с двух сторон. *Во-первых*, сложные проблемы не моделируются без компьютера. *Во-вторых*, технология современного моделирования не может быть не интеллектуальной.

Здесь мы не будем анализировать понятие интеллектуальности каких-либо процессов (см. в [Гаврилова и др., 2000]). Заметим, любой процесс мышления интеллектуален, но, теперь, *интеллектуальность* практически всегда, подразумевает *компьютерность* реализации соответствующих процессов. Мы с этим согласны. Но, здесь нам интересна интеллектуальность моделирования, а нормальных определений этого понятия мы пока не нашли.

Поэтому, здесь, на основании общих соображений, нашего и «чужого» опыта (в создании и практического использования систем ИИ (см., например, [Гаврилова и др., 2000; Guarino, 1992]), мы будем считать, что *интеллектуальность моделирования состоит в автоматизированном (или автоматическом) использовании общих и специальных знаний процессов моделирования, полученных и используемых при решении практических проблем.*

Обратим внимание, что мы здесь не стремимся к строгому определению. Главное, построить систему, которая максимально аккумулирует, систематизирует, классифицирует опыт моделирования различных процессов, объектов, ситуаций и обеспечивает возможность его автоматизированного (или автоматического) использования при решении практических задач.

В настоящее время такие системы называются онтологиями. Широко известны многие интерпретации этого понятия. Поэтому, здесь ведем

рабочее определение (сформулированное на основе дефиниций Т. Грубера, Н. Гуарино, Т. Гавриловой, В. Хорошевского и др.), ориентированное на обеспечение процессов интеллектуального моделирования. *Онтология – специализированная база знаний, в которой эксплицированы все концепты и отношения между ними, необходимые и достаточные для решения задач данной проблемной области.*

2. Метасистемы и инфраструктуры моделирования

На рис. 1 представлена схема метасистемы моделирования. В этой структуре выделяется 3 класса компонент:

- *объекты моделирования,*
- *модели этих объектов,*
- *субъекты (проектировщики моделей).*

Обратим внимание, что отношение моделирования не бинарно (*объект - модель*), а тернарно. У любой модели всегда есть автор. Он определяет цели, исследует решаемые проблемы. Выбирает свойства, которые считает значимыми для данной цели, синтезирует для нее параметры, выбирает методы и средства моделирования.

Заметим, часто, модель синтезируется одним субъектом, а анализируется другим, и, тогда, особое значение приобретает описание контекста модели (особенно неявного) – см. определения 3 и 4 выше.

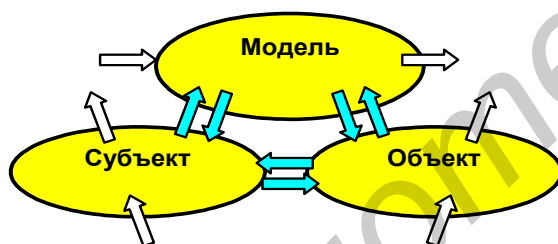


Рисунок 1 - Метасистема моделирования

Все стрелки на рисунке соответствуют отношениям между компонентами метасистемы моделирования и некоторыми процессами.

Соответственно (см. рис. 1) выделяются и три базы знаний. Можно также говорить о трех взаимосвязанных онтологиях.

1. В первой онтологии осуществляется концептуализация знаний о моделях, которые были построены ранее для различных объектов. Для построения структуры моделей необходимо построить классификаторы по различным критериям. В частности, важны: родо-видовые (таксономия) классификаторы, системные отношения (часть-целое, мерология), причинно-следственные, функционально-структурные. Должны в этой онтологии поддерживаться классификаторы свойств моделей, рассмотренные во втором разделе и многие другие. Эти структуры

еще и являются базовыми для ввода новых моделей в базу знаний.

Заметим, в этой и всех других онтологиях выделяются общая и специализированная части. Первая – ориентирована на поддержку любой системы моделирования, а во второй отражаются знания специальных моделирующих систем.

И, конечно, эта онтология должна быть связана со всеми другими онтологиями интеллектуального моделирования.

2. Во второй онтологии производится структуризация знаний о моделируемых процессах, явлениях, объектах. В дальнейшем все моделируемое будем называть объектами моделирования. Эти объекты целесообразно рассматривать как системы и они также классифицируются посредством родо-видовых, системных (часть-целое), причинно-следственных и т.п. отношений.

Конечно, эти онтологии (можно говорить об онтологиях материального (реального) мира) связаны отношениями со всеми другими онтологиями интеллектуального моделирования.

3. В третьей онтологии концептуализируются знания конструкторов моделей. В этих онтологиях отражаются знания целей моделирования, используемые средства, методы, технологии, решаемые задачи и проблемы, отношения «проблемы-задачи». Традиционно эта компонента называется онтологией задач. Но, здесь находятся и знания алгоритмов или технологий их решения.

Структура этой онтологии базируется на классификаторах методов, технологий, целей, средств моделирования. При структуризации этих отношений помимо традиционных структур (таксономия, мерология, и т.п.) используются текстовые описания. Очень важным компонентом этой онтологии является не только статистика (успешные и «неуспешные» проекты моделирования), но и анализ причин успеха и «неуспеха» применений различных методов, технологий в различных конкретных случаях. Это сложные отношения в системе «*решаемая проблема (задача) – технология (метод, средство, алгоритм – объект - модель*»

Метасистема не имеет выделенного объекта, относительно которого ведется исследование. Кроме того, здесь и далее понятие метамодели будет использоваться несколько в другом аспекте. Поэтому, будем говорить об инфраструктуре некоторой выделенной системы.

Инфраструктура — комплекс взаимосвязанных обслуживающих структур или объектов, составляющих и/или обеспечивающих основу функционирования системы

Поскольку компонента «*проблема (задачи)*» занимает особое место в структурах моделирования (она влияет на выбор цели (или наборот), метода,

технологии, структуры исходной модели и т.д.), то целесообразно выделить отдельно эту компоненту в инфраструктуре модели. Мы назвали эту структуру «расширенной» (появилась, новая, нетрадиционная компонента).

На рис. 2 представлена схема этой инфраструктуры. Здесь, помимо традиционной метасистемы моделирования, появилась компонента «проблема (задача)» и, теперь мы все рассматриваем относительно «МОДЕЛИ». Нам интересно, какая проблема решается посредством синтеза данной модели, какой объект отражается в данной модели, кто построил данную модель или кто ее использует при решении данной задачи.



Рисунок 2 - Инфраструктура модели

Мы можем в центр инфраструктуры поставить любой из этих четырех компонент. На рис. 3 представлена инфраструктура РЕШАЕМОЙ ПРОБЛЕМЫ.

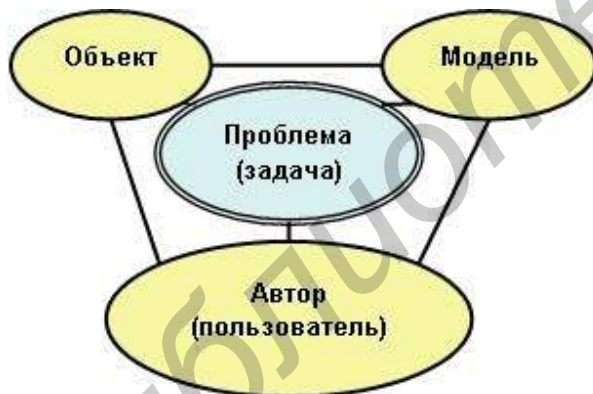


Рисунок 3 - Инфраструктура решаемой проблемы

Теперь мы рассматриваем все относительно *проблемы* (задачи или их множества), которую мы решаем посредством моделирования соответствующего объекта. При этом автор определяет цель, адекватную решаемым задачам, значимые параметры объекта моделирования, выбирает методы, алгоритмы, технологии решения задач и т.д. Информация об объекте, которую субъект должен получить при моделировании, различается в зависимости от задачи. Понять, какая информация существенная, а какая — нет, это отдельная проблема, общего решения которой не существует. Именно на решение этих проблем и

ориентированна онтология интеллектуального моделирования.

Здесь мы предлагаем, по сути, технологию центрирования схем инфраструктур онтологий для обеспечения полноты и согласованности решаемых проблем интеллектуализации моделирования.

3. Классификация моделей систем по Клиру

Предварительно необходимо рассмотреть отношения «система - модель». Представляется вполне очевидным, что любая модель является системой (отношений между параметрами, характеризующими объект моделирования). И любая система является разновидностью модели объекта моделирования. В том смысле, что система – это некоторое представление объекта моделирования (в форме отношений между его компонентами на весьма абстрактном уровне).

С нашей точки зрения, наиболее целесообразной основой классификации моделей для построения онтологии их структур является иерархия систем, исследуемая в [Клир, 1990] для построения ЭС моделирования сложных систем. На рис. 4 представлена полурешетка эпистемологических типов систем Клира. Он рассматривает 5 обобщенных уровней наших знаний о системах:

- *исходные системы* (базовые свойства, характеризующие систему),
- *системы данных* (матрицы значений соответствующих свойствам параметров),
- *порождающие системы* (модели - правила, законы, формулы и т.д. - устройства и/или ее функционирования),
- *структурированные системы* (отношения между построенными моделями для сложных систем),
- *метасистемы* (отношения между построенными ниже отношениями).

Рассмотрим подробнее эти пять уровней.

1. *Исходные системы (S)*, по сути, представляют собой вектора значимых (существенных) параметров, которые характеризуют модели исследуемых систем. В основе формирования этого вектора находятся только наши гипотезы (основанные на опыте и анализе аналогий) о моделируемом объекте. Исходная система – это схема, по которой могут быть сделаны наблюдения отобранных признаков.

2. В *системах данных (D)*, как правило, находятся матрицы некоторой информации. Они могут быть получены, по крайней мере, тремя путями. *Во-первых*, в результате наблюдений или измерений. *Во-вторых*, они могут быть выведены из систем (моделей) более высоких уровней или вычислены посредством каких-либо методов или технологий. *В-третьих*, эти матрицы могут быть

построены проектировщиком каких-либо систем (см., например, в [Валькман и др., 2012b]).

3. *Порождающие системы (F)* (см. рис. 4), часто, представлены полиномиальными уравнениями (в случае применения методов математической статистики к измерительным матрицам). Но, порождающими моделями (системами) могут быть и дифференциальные уравнения, и производные системы, деревья решений, семантические сети и т.д. Основная функция моделей этого типа является возможность порождения информации о поведении или структуре моделируемого объекта.

Следующие два уровня эпистемологии Клира целесообразно рассматривать на основе схемы полурешетки, представленной на рис. 4. Эти системы соответствуют отношениям «часть-целое» и «род-вид».

4. *Структурированные системы (SS, SD, SF)*, если говорить в общих чертах, представляют собой набор исходных систем, систем данных или порождающих систем, имеющих общее параметрическое представление. Системы, образующие структурированную систему называются ее *компонентами*. Некоторые параметры у них будут общими. Их называют *связывающими* параметрами. Они представляют взаимодействия между компонентами. В [Клир, 1990] рассматривается розовый куст в горшке. В этой структурированной системе выделяется шесть компонент (подсистем - моделей): *почва, корни, стебель, сок растения, листья и цветы*.

В [Валькман и др., 2012b] представлена специальная методология построения и исследования модельно-параметрического

пространства моделей объектов военного кораблестроения. Это пространство соответствует многоуровневой структурированной системе.

В базе знаний этой системы поддерживается синтез и анализ около 2000 моделей, которые построены на 700 параметрах (см. в [Валькман и др., 2012b]).

ОПРЕДЕЛЕНИЕ 5. Под *модельно-параметрическим* ($\langle M, P \rangle$ -) пространством будем понимать множество всех моделей, параметров, отношений между ними, характеризующих свойства (проектируемого и/или исследуемого) объекта.

В общем случае $\langle M, P \rangle$ -пространство можно трактовать как двудольный граф (понимая под его вершинами - различные M и P , а под дугами — отношения между ними) или семантическую сеть со специальными свойствами. Структура $\langle M, P \rangle$ -пространства детально рассмотрена в [Валькман и др., 2012b].

5. *Метасистемы (MS, MD, MF)* в приложении к системам включают три смысла:

- 1) метасистема может быть определена только *после* того, как определены другие типы систем;
- 2) эта система описывает *изменение* – замену одной систему другой;
- 3) она выше отдельных систем – процедура замены делает ее чем-то большим, чем набор отдельных процедур.

В качестве примера метасистемы Клир в [Клир, 1990] приводит интегрированную систему с данными (D) работу светофора: *обычно, ночью и в час пик*.

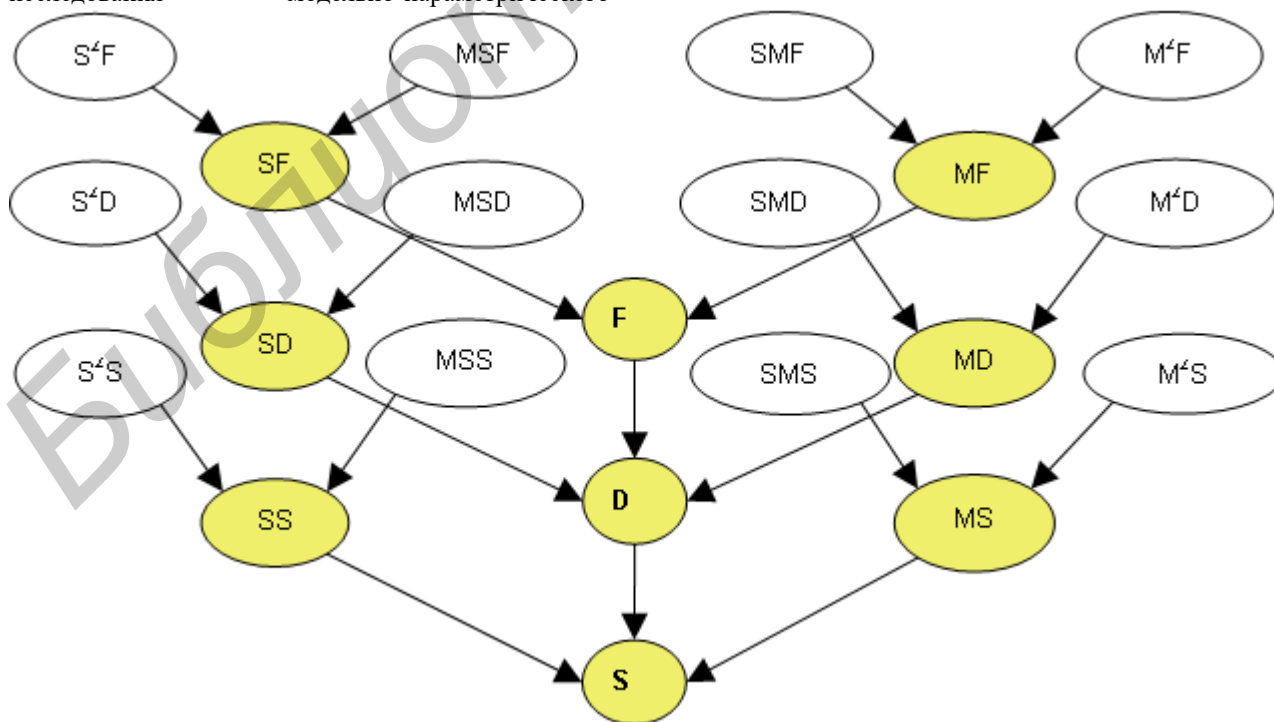


Рисунок 4 - Полурешетка эпистемологических типов систем

Мы в метасистемах видим не только отражение динамики, но и представление структур статических отношений «род-вид».

Последние два класса систем могут применяться многократно друг к другу и/или рекурсивно (см. *SMS, SMD, SMF, MSS, MSD, MSF, M²S, M²D, M²F, S²S, S²D, S²F*, в полурешетке на рис. 4). Так например, для корабля одна из структурированных исходных систем второго уровня *S²S* - модели всех видов качек (суперпозиция бортовой, килевой, вертикальной и др. видов качек структуры *SS*) в интегрированной модели (*<M, P>*-пространстве) корабля *S⁴S*.

Рассмотренные здесь классификации систем (для нас, *моделей*) и принципы построения *<M, P>*-пространства являются базой для разработки онтологий интеллектуального моделирования.

Заключение

В статье рассматриваются некоторые принципы построения онтологии моделей, процессов моделирования, моделируемых объектов, решаемых проблем, задач для разработки и использования интеллектуальных систем моделирования. В дальнейшем планируется создание компьютерной системы, обеспечивающей создание и использование онтологических БЗ, некоторые принципы создания которых здесь изложены.

Разработанные программно-информационные системы могут использоваться в комплексе. Но, могут применяться и локально в различных приложениях. Так различные онтологические БЗ могут использоваться в качестве основной компоненты при создании ЭС моделирования в разнообразных предметных и проблемных областях. Могут они применяться и в качестве инструментальных комплексов при разработке различных интеллектуальных систем и технологий.

Библиографический список

- [Валькман, 2012a] Валькман Ю.Р. Когнитивная семиотика // Сб. научн. тр. I Междунар. симпозиума "Гибридные и синергетические интеллектуальные системы: теория и практика", Москва, Калининград, "Наука. Физматлит", май, 2012, с. 98-116.
- [Валькман и др., 2012b] Валькман Ю. Р., Гриценко В.И., Рыхальский А. Ю. Модельно-параметрическое пространство. Теория и приложение. Киев: Наукова думка, 2012. – 257 с.
- [Гаврилова и др., 2000] Гаврилова Т. А., Хорошевский В.Ф. Базы знаний в интеллектуальных системах. – Санкт-Петербург, Питер, 2000 г. – 384 с.
- [Клир, 1990] Клир Дж. Системология. Автоматизация решения системных задач. - Москва: Радио и связь, 1990. - 544 с.
- [Перегудов и др., 1989] Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ — М.: Высш. школа, 1989. – 376 с.
- [Саймон, 1972] Саймон Г. Науки об искусственном. - Москва: Мир, 1972. – 242 с.
- [Guarino, 1992] Guarino N. Concepts, attributes and arbitrary relations // Data Knowledge Engineering.1992. 8. pp. 249-261.

ON ONTOLOGIES OF INTELLECTUAL MODELING

Valkman Y.R., Stepashko P.V.

International Research and Training Center for Information Technologies and Systems of the National Academy of Sciences of Ukraine and Ministry of Education and Science of Ukraine, Kiev

yur@valkman.kiev.ua

pstepashko@gmail.com

The core of the intellectual modeling system is an ontology. Some principles of building an ontology of models, modeling processes, objects being modeled, problems are discussed.

Introduction

Models are becoming more complicated. Specialists have to be closely familiar with the specifics of their problem areas, not the specifics of the computer modeling technologies. Therefore, the development of methods of intellectual modeling is considered of current interest.

Main Part

The *intellectuality of modeling* will be considered to comprise automated usage of common and specific knowledge of the modeled processes, acquired and used while solving practical problems. The basis of building such systems would be ontologies. They are built on the basis of classifiers of appropriate objects and processes.

A few principles of development of classification are reviewed. In the modeling system three components can be separated: modeled objects, models of the objects, subjects (the model designers). In the semilattice of epistemological system types of J. Klir, 5 classes of models are discussed. The first three (*source systems, data systems and generating systems*) are sets. The other two classes (*structured models and metamodels*) are recursive sets.

These structures provide the possibilities to accumulate successful and unsuccessful experience in modeling of complicated objects for its practical application.

Conclusion

The principles of constructing metasystems, infrastructures, model classifiers that were considered here are the basis of development of ontologies of intellectual modeling.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822: 004.912

МОДЕЛИ ПРЕДМЕТНЫХ ОБЛАСТЕЙ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ МОНИТОРИНГА ИНФОРМАЦИОННОГО ПРОСТРАНСТВА

Додонов А.Г., Ландэ Д.В., Коваленко Т.В.

*Институт проблем регистрации информации НАН Украины,
г. Киев, Украина*

dodonov@ipri.kiev.ua

dwlande@gmail.com

2005ste@ukr.net

Описываются подходы к построению моделей предметных областей в системах поддержки принятия решений. Как источники знаний рассматриваются текстовые корпуса и ресурсы современных наукометрических сервисов. Представлена архитектура системы информационной поддержки принятия решений, идеология создания и использования онтологий для построения сюжетов информационной поддержки принятия решений. Предложена методология аналитического исследования, которая базируется на использовании инструментальных средств анализа и визуализации информационных потоков и сетевых структур.

Ключевые слова: поддержка принятия решений, онтология, модель предметной области, информационное пространство, контент-мониторинг.

Введение

В настоящее время все большее применение находят знание-ориентированные информационные системы с онтолого-управляемой архитектурой [Палагин, 2012], которые обеспечивают компьютерную обработку естественно-языковых объектов, описывающих предметные знания: объекты, их свойства и понятия предметной области [Ландэ, 2013], [Добров, 2009].

В частности, системы с онтолого-управляемой архитектурой находят свое применение в системах информационной поддержки процессов противодействия информационным кампаниям, деструктивным внешним и внутренним информационным воздействиям, что является актуальной проблемой современности [Горбулін, 2009], [Шульц, 2011].

В качестве основных задач такой системы рассмотрим:

- построение сценариев противодействия информационным деструктивным воздействиям на основе некоторой онтологии понятий;
- контент-мониторинг (непрерывный содержательный анализ) информационного пространства на основе знаний экспертов;

- выявление закономерностей (трендов) и аномалий путем анализа динамики изменения значений отдельных факторов;
- выявление информационных воздействий и информационных операций;
- прогнозирование развития информационных сюжетов и ситуаций;
- оценка эффективности процедур поддержки принятия решений.

Соответственно, для реализации такой системы информационной поддержки процессов, связанных, в частности, национальной безопасностью необходимо:

- создать онтологию понятий предметной области (узлов – факторов безопасности и соответствующих причинно-следственных (каузальных) связей – зависимости факторов), определить вид целевой функции безопасности объектов-узлов этой онтологии в зависимости от значений факторов безопасности;
- постоянно актуализировать значение факторов безопасности и связей в зависимости от результатов мониторинга информационного пространства и знаний экспертов;
- определять возможные сценарии на основе анализа онтологии и выявления соответствующих частных онтологий;

- анализировать динамику изменения значений отдельных факторов и связей с целью выявления закономерностей, прогнозирования;
- постоянно проводить оценку эффективности проводимой информационной поддержки.

Для решения приведенных выше задач системы информационной поддержки принятия решений, предполагается, что система информационной поддержки должна состоять из трех основных подсистем (рис. 1): подсистемы создания онтологии понятий, подсистемы мониторинга информационного пространства, подсистемы аналитической обработки, и соответствующих интерфейсов с администраторами и пользователями.



Рисунок 1 – Архитектура системы информационной поддержки

1. Задача формирования онтологии

Онтологический подход логично интегрируется со сложными динамическими системами. Вместе с тем, сами онтологии, представляемые в виде семантических сетей, когнитивных карт, каузальных сетей и т.п., являются динамическими объектами, состав их узлов и ребер постоянно изменяется, причем эксперты не всегда в состоянии оперативно отслеживать эти изменения. Таким образом, современная онтологическая система, необходимая для принятия решений, должна содержать характеристики как предметной области, так и внешней информационной среды, с которой система должна взаимодействовать.

Онтология в данном случае представляет собой функциональный аналог базы знаний, отражающей знания экспертов о предметной области, т.е. в качестве узлов графа онтологии выбираются важнейшие факторы предметной области обеспечения безопасности, а в качестве связей - причинно-следственные связи между факторами (с математической точки зрения – граф с направленными ребрами) [Шульц, 2014]. Узлам и связям приписываются числовые значения, которые в дальнейшем могут корректироваться. Связи также могут иметь разный вес (силу воздействия) и быть как положительными (увеличение значения первого фактора приводит к увеличению значения второго фактора), так и отрицательными (увеличение значения первого фактора приводит к уменьшению значения второго фактора).

При формировании предметных онтологий, применяемых в системах поддержки принятия решений, решается несколько содержательных задач, среди которых:

- выявление узлов – основных понятий и соответствующих им слов и словосочетаний из соответствующей предметной области;
- выявление различных семантических связей между узлами и соответствующих им понятий;
- ранжирование понятий, выявление главных понятий в предметной области;
- ранжирование связей, выявление главных связей.

Построение онтологии в процессе изучения некоторой системы может рассматриваться как прямая задача, где заранее известна схема управления, основные объекты и связи, в соответствии с которыми формируется сеть понятий.

Задача автоматического формирования модели предметной области еще на этапе формирования реальной системы, в процессе принятия решений об ее структуре и функциях, когда не определены объекты и связи, может рассматриваться как обратная задача, в которой сам состав понятий и связей, выбираемых при создании предметной области, определяют дальнейший состав (а иногда и функции) системы.

Именно в решении такой обратной задачи заключается проблема формирования онтологий предметных областей. Онтология может формироваться экспертами (что и происходит чаще всего), однако, перспективным кажется автоматическое формирование онтологий, которое может базироваться на знаниях, заложенных учеными, специалистами, экспертами в таких источниках, как:

- фактографические базы данных/знаний;
- текстовые корпуса;
- ресурсы современных социальных, наукометрических, библиографических сетей.

Конечно, если существуют фактографические базы данных или знаний, то проблема автоматического формирования онтологий на их основе может показаться несложной. Но при этом задача предварительного формирования таких баз данных и знаний, сама по себе, является проблемной, хотя и уже традиционной.

2. Источники формирования онтологий

Рассмотрим случаи использования двух других источников для формирования моделей предметных областей, а точнее, для частичной задачи – формирования терминологической основы предметной области систем организационного управления, связанных со свойствами живучести и надежности [Додонов, 2011]. В начале рассмотрим

построение сетей иерархий терминов на основе анализа корпуса текстов по выбранной проблематике, которое базируется на применении компактифицированных графов горизонтальной видимости для терминов – отдельных слов, биграмм и триграмм, а также установлении связей между терминами.

Как терминологическая основа для формирования терминологических онтологий используется сеть естественной иерархии терминов (СЕИТ), которая базируется на информационно-значимых элементах текста, опорных словах и словосочетаниях, методология выявления которых приведена в [Ландэ, 2014-1], [Ландэ, 2014-2]. Данная методика предусматривает реализацию шагов, охватывающих предварительную обработку исходного текста, выявление терминов, выбор из них необходимого количества наиболее весомых, непосредственное построение сети и ее отображение. На первом этапе формируется исходный текстовый корпус. Как пример такого корпуса рассматриваются полные тексты научных статей, посвященных проблематике живучести в информационных и технических системах, представленных на русском языке. В состав текстового корпуса было в качестве примера включено около 50 научных статей общим объемом около 1 млн. символов. Предварительная обработка такого текстового корпуса предусматривала выделение фрагментов текстов (отдельных статей, абзацев, предложений, слов), исключение нетекстовых символов, отсечение флективных окончаний (стемминг). На втором этапе каждому отдельному термину из текста (слову-униграмме, биграмме или триграмме) ставится в соответствие оценка их "дискриминантная сила", а именно TFIDF, которая в каноническом виде равна произведению частоты соответствующего термина (Term Frequency) в фрагменте текста на двоичный логарифм от величины, обратной к количеству фрагментов текста, в которых этот термин встретился (Inverse Document Frequency). Для последовательностей терминов и их весовых значений по TFIDF затем строились компактифицированные графы горизонтальной видимости (CHVG) и выполнялось переопределение весовых значений слов уже по этому алгоритму. Данная процедура позволяет учитывать в дальнейшем кроме терминов с большой дискриминантной силой также высокочастотные термины, которые имеют большое значение для общей тематики. На последнем этапе формирования СЕИТ осуществляется ее отображение программными средствами анализа и визуализации графов. Для загрузки СЕИТ в базы данным формируется матрица инцидентности общепринятого формата csv. На рис. 2 представлена небольшая сеть естественной иерархии терминов.

Для работы с третьим типом источников предлагается методика построения моделей предметных областей на основе зондирования информационных сетей [Ландэ, 2015]. Как такая

сеть рассматривается сеть понятий, соответствующих тегам сервиса Google Scholar Citations (<http://scholar.google.com/citations>). Множество тегов (понятий) в этой сети образуют сеть, узлы которой соответствуют понятиям, а связи – некоторую семантическую связь между ними.

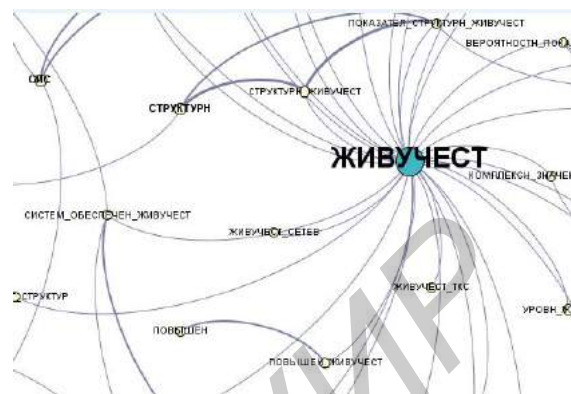


Рисунок 2 – Визуализация связанного фрагмента СЕИТ

Зондирование выбранной опорной сети Google Scholar Citations осуществляется по такому алгоритму:

1. Выбирается определенное количество узлов опорной сети, определяемых как базовые для новой сети, соответствующей результатам зондирования.
2. Для каждого из рассматриваемых узлов опорной сети определяются смежные с ним узлы ("соседи"), которые добавляются к создаваемой сети с результатами зондирования.
3. От текущего узла опорной сети осуществляется переход к соседнему узлу, имеющему наибольшую степень.
4. Если имеет место "заикливание" (выбирается узел, к которому уже был осуществлен переход по этому алгоритму), происходит переход к следующему по степени соседнему узлу. Если таких узлов не осталось – осуществляется переход к пункту 2. Если перечень базовых узлов завершен, считается, что сеть, соответствующая результатам зондирования, построена.
5. Формирование базового стартового перечня узлов-понятий и правил отбора «конечных» узлов выполняется экспертами в предметной области.
6. Для построения модели предметной области (в рассматриваемом примере для области искусственного интеллекта) экспертным путем были определены базовые теги на английском языке: *survivability*, *reliability*, *dependability*, *resilience*, *fault tolerance*, *availability*, *safety*, *durability*, *security*, *resiliency* др.

На рис. 3 приведен пример архитектуры сети понятий предметной области, построенной в соответствии с приведенным алгоритмом по указанным базовым тегам.

Построенная сеть понятий оказалась связанной. При количестве базовых тегов 20, общее количество узлов-тегов, которые были охвачены алгоритмом, составили 125, а количество нетерминальных узлов – 1450.

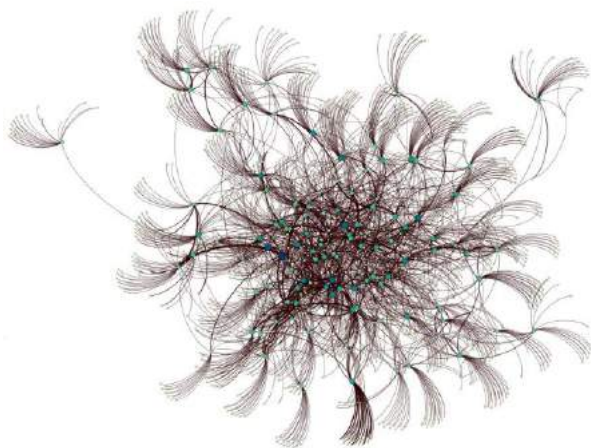


Рисунок 3 – Фрагмент архитектуры сети понятий

3. Интеграция сценарного анализа и мониторинга информационного пространства

Задача создания онтологии понятий уже рассмотрена выше, остановимся подробнее на средствах формирования сценариев и задачах мониторинга и последующей аналитики информационного пространства.

3.1. Определение возможных сценариев

Сценарии информационной поддержки, как правило, связывают с определенными факторами безопасности (чаще объектами и уязвимостями). После выбора целевых факторов сценария в графе онтологии оказываются подграфы (частичные онтологии), тесно связанные с выбранными факторами. Далее решается задача частичной оптимизации целевой функции на избранных подграфах, то есть вычисляется целевая функция в зависимости от изменений факторов безопасности, соответствующих возможным сценариям.

Существует несколько подходов к вычислению взаимных влияний между узлами семантической сети [Робертс, 1986], которая может трактоваться как сеть причинно-следственных связей между объектами в заданной предметной области. В частности, в случае задачи, связанной с обеспечением безопасности некоторого объекта, соответствующего узлу, из общей каузальной сети выбираются узлы-объекты, имеющие наибольший вес связи с целевым объектом. Выделение подсетей из данных узлов и целевого узла, а также связей между ними, определение планов очередности влияния на эти подсети, задает сценарии информационной поддержки, в частности, в задачах принятия решений в области безопасности.

Детально рассмотрим подход, предложенный в [Снарский, 2015]. При решении этой задачи решающим фактором оказывается определение силы связи между отдельными узлами онтологии – между отдельными факторами безопасности, в рассматриваемом случае.

В рассматриваемой сети узлы связаны между собой связями, отражающими влияние одного узла на другой. Каждой связи приписано две характеристики: первая – направление влияния, вторая – величина влияния (рис. 4а).

Так например, узел 1 напрямую влияет на узел 3, но узел 3 напрямую на узел 1 не влияет.

Кроме того, каждой связи, задающей влияние, соответствует ее весовая величина. Например, для связи между узлами 1 и 3 это влияние задается как число ε_{13} , которое может быть как положительным, так и отрицательным.

Для решения задачи определения взаимного влияния узлов кажется естественным применение законов электротехники. Вместе с тем, подход опирающийся на непосредственное применение закона Кирхгофа приводит к некоторому логическому противоречию. В некоторых схемах влияние узла 1 на узел 2 $\varphi(1 \rightarrow 2)$ зависит от влияния узла 2 на узел 1 $\varphi(2 \rightarrow 1)$.

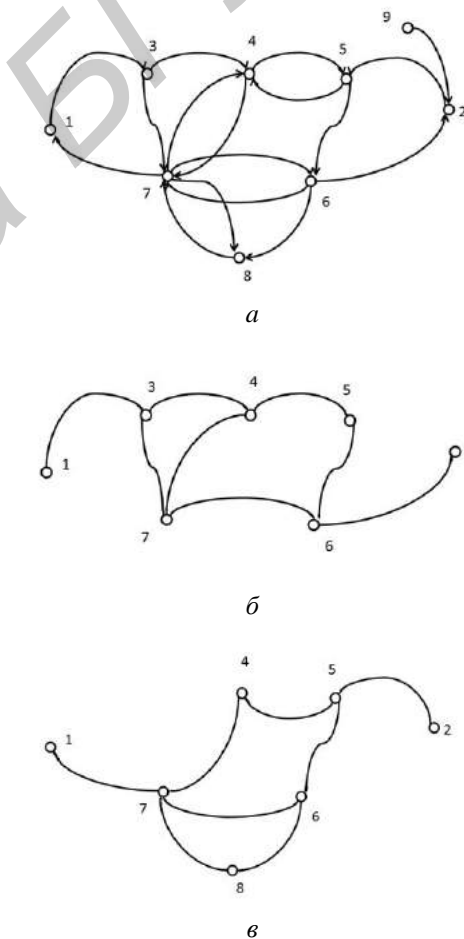


Рисунок 4 – Пример сети влияния: а – общая направленная сеть; б – подсеть влияния узла 1 на узел 2; в – подсеть влияния узла 2 на узел 1. Источник – [Снарский, 2015]

Поэтому авторами был предложен алгоритм определения величины влияния одного узла (например, 1-го) на другой (например, 2-й), состоящий из двух частей. Вначале необходимо выделить ту часть сети, по которой осуществляется

влияние первого узла на второй. Для этого из начальной сети (рис. 4а) нужно удалить все связи, оставляя только те, по которым можно пройти из первого узла во второй (рис. 4б). Аналогично выделяется сеть влияния второго узла на первый (рис. 4в). Отметим, что в общем случае сети, изображенные на рис. 4б и 4в, не совпадают.

После выделения подсетей (подсети 1→2 и 2→1) используем аналогию с электрическими сетями. Величина влияния i -го узла на j -й ε_{ij} в данной аналогии – это ЭДС. Сопротивление каждой связи в рамках данного допущения принимается равным 1. Распределение токов – J_{ij} , текущих по связям ij , определяется решением уравнения Кирхгофа:

$$\forall k: \sum_{i \rightarrow k \rightarrow j} J_{ij} = 0, \quad \forall z: \sum_{i, j \in z} J_{ij} = \sum_{i, j \in z} \varepsilon_{ij}, \quad (1)$$

где первое уравнение требует (для стационарного случая), чтобы алгебраическая сумма токов в каждом узле k была равной нулю, а второе, чтобы для любого замкнутого контура z сумма токов (с учетом того, что сопротивление всех связей равно единице) было равно сумме всех ЭДС ε_{ij} в этом же контуре.

После решения системы уравнений (1) и нахождения токов J_{ij} выбирается любой контур, соединяющий узлы 1 и 2, например, контур 1-3-4-5-6-2 и из уравнения Кирхгофа для незамкнутого контура:

$$\sum \varepsilon + \varphi_1 - \varphi_2 = \sum J, \quad (2)$$

находится разность потенциалов $\varphi_1 - \varphi_2$ (суммирование производится по связям выбранного контура).

Аналогично происходит вычисление для любых других пар узлов.

Именно поэтому для нахождения влияния одного узла на другой расчет приводится в соответствии с несколько измененной схемой, соответствующей рис. 4а–4в. На первом шагу от направленной сети (рис. 4а) происходит переход к ненаправленным подсетям (рис. 4б и 4в). И так, на первом шаге происходит модификация направленной сети связи. После выбора узлов, для которых изучается взаимное влияние, остаются только связи, по которым можно пройти от одного узла к другому (рис. 4б и 4в).

На втором шаге происходит расчет $\varphi(1 \rightarrow 2)$ происходит по правилам Кирхгофа (рис. 4б). Естественно, при расчете влияния узла 2 на узел 1 расчет происходит по другой схеме (рис. 4в). При этом связи считаются обычными проводниками с единичным сопротивлением, и в $\varphi(\alpha \rightarrow \beta) = \varphi(\beta) - \varphi(\alpha)$ принимается $\varphi(\alpha) = 0$. Заметим, что при таком методе расчета влияние узла

1 на 2 не зависит от влияния узла 2 на 1.

Анализ динамики изменения значений отдельных факторов безопасности и связей во времени (как по их отражению в информационном пространстве, так и внесенных экспертами) позволяет выявлять некоторые закономерности изменения этих факторов (периодичности, тренды, аномалии) путем применения современных средств цифровой обработки сигналов (регрессионный, дисперсионный, вейвлет-анализ и т.д.), выявлять возможные информационные операции путем сравнения с соответствующими шаблонами их динамики, а также осуществлять прогнозирование [Додонов, 2014].

3.2. Определение Контент-мониторинг и аналитика

Сегодня Интернет образует значимый динамичный сегмент информационного пространства, информационные потоки, содержание и объемы которых необходимо учитывать при проведении аналитических исследований. Основным объектом анализа при этом является событийные или тематические срезы этих потоков – массивы информационных сообщений, документов, соответствующих определенным событиям или тематикам.

Задача подсистемы мониторинга информационного пространства следующие:

- мониторинг целевых объектов;
- нахождение релевантных тематических сообщений в информационном пространстве;
- контроль медиаприсутствия и медиаактивности целевых объектов;
- выявление новых объектов мониторинга;
- формирование ретроспективных фондов для последующего анализа.

Подсистема аналитической обработки представляет собой аналитический блок системы информационной поддержки, обеспечивает решение следующих задач:

- определение динамики тематических сюжетов;
- определение критических точек в динамике тематических сюжетов;
- отслеживание сюжетных цепочек, соответствующих событий, процессов;
- выявление основных событий и объектов с тематического сюжета;
- выявление и визуализация взаимосвязей событий и объектов мониторинга, а также объектов мониторинга между собой.

В соответствии со своим назначением данная подсистема, вместе с подсистемой мониторинга информационного пространства, позволяет реализовать следующие этапы информационно-аналитического исследования:

- формирование запроса в среде выбранной системы. Нахождение тематических публикаций по запросу с помощью систем контент-мониторинга;
- определение динамики тематических публикаций по запросу;
- определение критических точек в динамике тематических публикаций;
- определение основных событий в критических точках;
- выявление объектов мониторинга;
- выявление и визуализация взаимосвязей;
- прогноз развития событий.

Заключение

Таким образом, представлена архитектура системы информационной поддержки принятия решений, идеология создания и использования онтологий для построения сюжетов информационного противодействия. Также рассмотрена методика аналитического исследования, основанная на использовании инструментальных средств анализа и визуализации информационных потоков, соответствующих временных рядов и сетевых структур.

Автоматически сформированные терминологические основы онтологий и соответствующие, пусть и простейшие, семантические связи по выбранным свойствам (живучести, надежности) могут использоваться, в частности, в качестве «единого» для всех участников разработки языка предметной области, для обучения, тренингов, организации семантического поиска (организации контекстных подсказок информационно-поисковых систем), навигации пользователей в соответствующих информационных ресурсах.

Предложенные архитектурные решения можно использовать при реализации систем информационной поддержки принятия решений, основанных на контент-мониторинге информационного пространства и сценарном анализе, а так же в качестве базы для проведения аналитической и прогнозной деятельности.

Авторы благодарны коллегам и соавторам А.А. Снарскому, С.М. Брайчевскому, В.Г. Пуятину, В.А. Додонову и В.Н. Фурашеву за внимание, поддержку и интерес, проявляемые при обсуждении рассматриваемых подходов.

Библиографический список

- [Горбулін, 2009] Горбулін В.П., Додонов О.Г., Ланде Д.В. Інформаційні операції та безпека суспільства: Загрози, протидія, моделювання: монографія. – К.: Інтертехнологія, 2009. – 164 с.
- [Добров, 2009] Добров Б.В., Соловьев В.Д., Лукашевич Н.В., Иванов В.В. Онтологии и тезаурусы. Модели, инструменты, приложения / Б.В. Добров, В.Д. Соловьев, Н.В. Лукашевич, В.В. Иванов. – М: Бином, 2009. – 173 с.
- [Додонов, 2011] Додонов А.Г. Живучесть информационных систем / А.Г. Додонов, Д.В. Ландэ. – К.: Наук. думка, 2011. – 256 с.
- [Додонов, 2012] Додонов А.Г., Ландэ Д.В. Методика аналитического исследования динамики событий на основе

мониторинга веб-ресурсов сети Интернет // Информационные технологии и безопасность: основы обеспечения информационной безопасности: Материалы международной научной конференции ИТБ-2014. – К.: ИПРИ НАН Украины, 2014. – С. 3-17.

[Ланде, 2014] Ланде Д.В. Элементы компьютерной лингвистики в правовой информатике. – К.: НДПП НАПрН України, 2014. – 168 с.

[Ландэ, 2014-1] Ландэ Д.В. Построение терминологической сети предметной области / Д.В. Ландэ, А.А. Снарский, В.Г. Пуятин // Реєстрація, зберігання і обробка даних, 2014. – Т. 14. – № 2. – С. 114-121.

[Ландэ, 2014-2] Ландэ Д.В. Применение КГТВ-алгоритма для научных текстов / Д.В. Ландэ, А.А. Снарский, Е.В. Ягунова // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2014): материалы IV междунар. науч.-техн. конф. (Минск 20-22 февраля 2014 года) / – Минск: БГУИР, 2014. – С. 199-204.

[Ландэ, 2015] Ландэ Д.В. Построение модели предметной области путем зондирования сервиса Google Scholar Citations // Онтология проектирования, 2015. - N 3(17). - С. 328-335.

[Палагин, 2012] Палагин А.В. Онтологические методы и средства обработки предметных знаний: монография / А.В. Палагин, С.Л. Кривый, Н.Г. Петренко. – Луганск: изд-во ВНУ им. В. Даля, 2012. – 324 с.

[Робертс, 1986] Робертс Ф.С. Дискретные математические модели с приложениями к социальным, биологическим и экономическим задачам. - М.: Наука, 1986.

[Снарский, 2015] Снарский А.А., Ландэ Д.В. Метод выделения подсетей в каузальных сетях в задачах сценарного анализа // Информационные технологии и безопасность. Материалы XV Международной научно-практической конференции ИТБ-2015. - К.: ИПРИ НАН Украины, 2015. - С. 212-215.

[Шульц, 2011] Шульц В.Л., Кульба В.В., Шелков А.Б., Чернов И.В. Сценарный анализ эффективности управления информационной поддержкой государственной политики России в Арктике. // Национальная безопасность / nota bene. – 2011. – № 6. – С. 104-137.

[Шульц, 2014] Шульц В.Л., Кульба В.В., Шелков А.Б., Чернов И.В. Структурно-динамический подход к сценарному анализу процессов информационного противоборства в Арктике // Труды XII Всероссийского совещания по проблемам управления (ВСПУ 2014). – М.: ИПУ РАН, 2014. – С. 8889-8901.

MODELS OF SUBJECT DOMAINS IN DECISION-MAKING SUPPORT SYSTEMS ON THE BASIS OF INFORMATION SPACE MONITORING

Dodonov A.G., Lande D.V., Kovalenko T.V.

*Institute for Information Recording NAS of
Ukraine, Kiev, Ukraine*

dodonov@ipri.kiev.ua

dwlande@gmail.com

2005ste@ukr.net

Approaches to creation of models of subject domains in decision-making support systems are described. As sources of knowledge text cases and resources of modern scientometric services are considered. The architecture of system of information support of decision-making, ideology of creation and use of ontologies for creation of plots of information support of decision-making is presented. The methodology of analytical research which is based on use of tools of the analysis and visualization of information streams and network structures is offered.

Keywords: *decision support, ontology, domain model, information space, content-monitoring*



УДК 004.822:514

ИСПОЛЬЗОВАНИЕ ОДНОРОДНОЙ СЕМАНТИЧЕСКОЙ СЕТИ ДЛЯ КЛАССИФИКАЦИИ РЕЗУЛЬТАТОВ ГЕНЕТИЧЕСКОГО АНАЛИЗА

Куликов А.М.* , Харламов А.А.**

*Институт биологии развития им. Н.К. Кольцова РАН, г. Москва
amkulikov@gmail.com

**Институт высшей нервной деятельности и нейрофизиологии РАН, г. Москва
kharlamov@analyst.ru

В работе показано использование механизма сравнения семантических сетей текстов в задаче диагностики заболеваний с использованием сигнальных сетей. Выявление степени пересечения семантических сетей текстов позволяет говорить о степени их смыслового подобия. Однородная семантическая сеть как множество узлов, связанных дугами, имеет численные характеристики – частоты появления слов, а также пар слов в тексте, которые перенормируются с использованием n-граммной модели текста. Такие сети как смысловые портреты текстов могут служить для сравнения (и, следовательно, для классификации) текстов. Генетический квазитекст может быть представлен, в том числе, в виде сигнальной или генной сети. Сигнальные сети разных классов генетических событий могут быть использованы для классификации этих текстов. В этом случае концентрации белков, выявленные в процессе эксперимента, используются для вычисления числовых характеристик узлов сети. Приведены примеры сравнения сетей генетических квазитекстов, соответствующих норме и патологии.

Ключевые слова: однородные семантические сети, сигнальные сети, сравнение текстов, классификация текстов

Введение

Предположение о сходстве текстов естественно-языковых и текстов генетических кодов (в дальнейшем будем их называть генетическими квазитекстами) оказывается правомерным при более подробном сравнении. Семантические сети как смысловые портреты естественно-языковых текстов [Харламов, 2006] имеют свою параллель в виде сетей, представляющих некоторые предметные области в генетике (например, определенную патологию). Рассмотрим это сравнение более подробно.

Семантическая сеть естественно-языкового текста – это граф, вершинами которого являются ключевые понятия этого текста, а дуги описывают взаимосвязи ключевых понятий в тексте. Типичный вид такого графа для, например, текста «Нейросетевая среда (нейроморфная ассоциативная память) для преодоления информационной сложности. Поиск смысла в слабо структурированных массивах информации. Часть I.

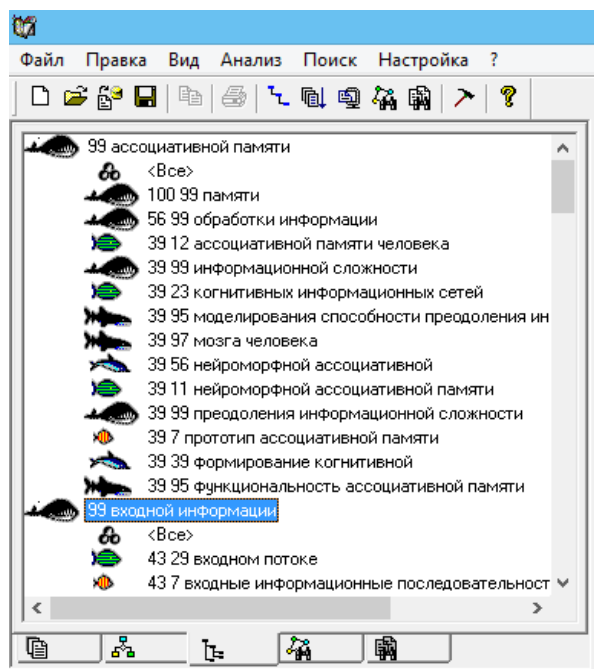


Рисунок 1 – Фрагмент семантической сети текста. Здесь перечислены вершины и показаны их связи. И вершины и связи имеют числовые характеристики

Структурная обработка информации в коре», опубликованный в 11 номере журнала «Информационные технологии» за 2015г. представлен (с использованием графического интерфейса программного продукта для автоматического смыслового анализа текстов TextAnalyst) на рисунке 1.

Такая сеть представляется как множество пар слов, встречающихся в тексте. И сеть они составляют как раз потому, что некоторые пары слов оказываются связанными между собой через промежуточное слово. Другими словами, такая сеть – это перечень пар слов.

Сеть, описывающая генетическое событие (см. как пример сеть, представленную на рисунке 2), как правило, получается при анализе активности работы генов (или экспрессионной активности) клеток, органов, тканей или целого организма на определенной стадии развития и/или под действием тех или иных факторов. В этом случае концентрации тех или других веществ позволяют делать предположение о протекании определенных генетических процессов, каждый из которых имеет свое начало и свой конец. Эти процессы могут иметь общие промежуточные компоненты, то есть суммарный граф, представляющий результаты эксперимента, также разбивается на пары событий (которые можно условно называть словами).

В приведенном примере на схеме изображена сеть межбелковых взаимодействий, представляющая передачу сигналов от толл-подобных рецепторов, участвующих в клеточном иммунном ответе, на определенный набор транскрипционных факторов, активирующих работу

соответствующих генов. Набор собственно толл-подобных рецепторов, вспомогательных белков и факторов, участвующих в активации этих рецепторов, расположен в верхней части схемы. В нижней части представлены транскрипционные факторы, т.е. регуляторы активности генов-мишеней. Синие и зеленые стрелки показывают отрицательные обратные связи между регулирующими транскрипционными факторами и центральными узлами данной сети, желтые и красные – аналогичные положительные обратные связи. Характер связей в представленном графе позволяет сделать вывод, что активация одних узлов сети приводит к формированию циклов последовательного усиления активности генов-мишеней, тогда как активация других приводит к последовательному снижению активности генов-мишеней. Количественные оценки состава белков и РНК в клетке или ткани позволяет, при наложении этих оценок на сигнальную сеть, сделать вывод об относительной активности различных участков такой сети, или подграфов, и об активности биологических процессов, определяемых данными подграфами.

Учитывая последовательность и дискретность актов передачи сигнала между белками-партнерами, весь набор передаваемых сигналов от вершин входа графа до набора вершин – «мишеней» за некоторый промежуток времени можно представить как набор одновременно и/или последовательно идущих предложений, где пары слов представлены парами взаимодействующих молекулярно-генетических объектов.

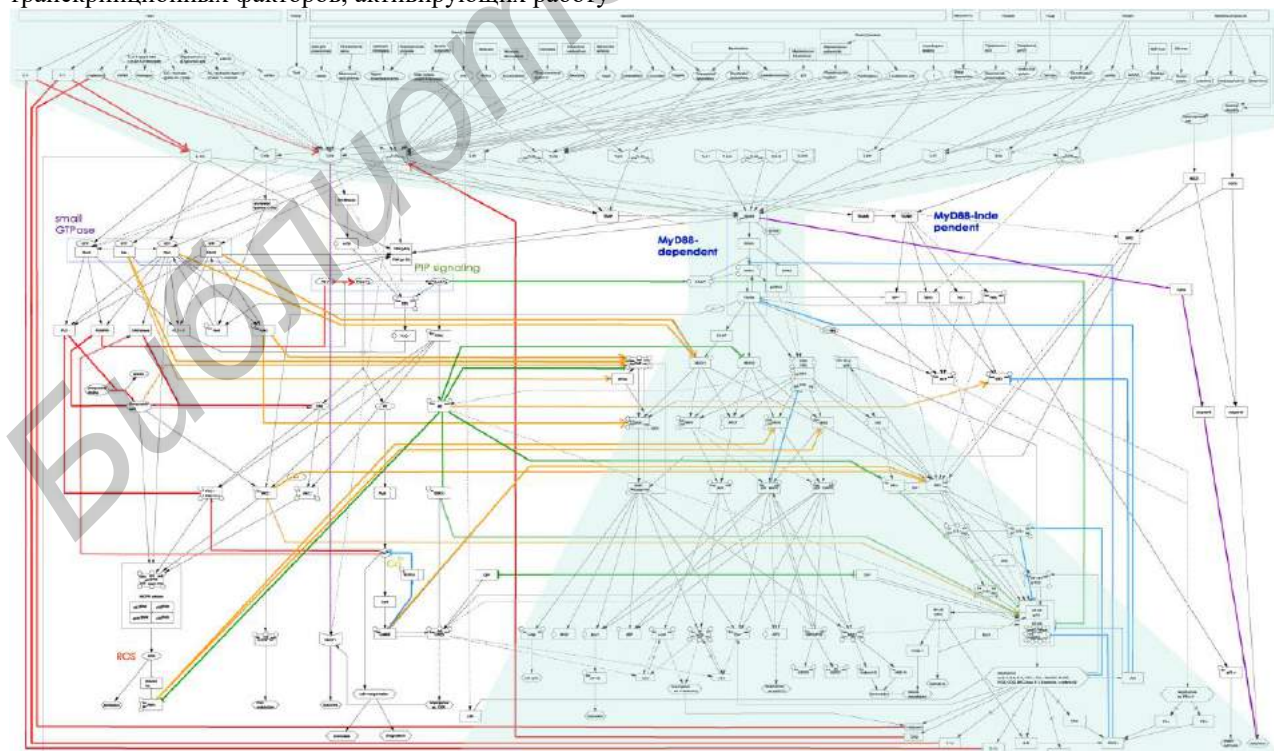


Рисунок 2 – Иллюстрация из статьи К. Oda и Н Kitano: A comprehensive map of the toll-like receptor signaling network. (Mol Syst Biol , 2006. 2:0015): схема сигнального пути toll-like рецепторов

Анализ семантических сетей – смысловых портретов – естественно-языковых текстов позволяет осуществлять сравнение текстов по структуре (по смыслу) [Харламов, 2006]. Аналогия с генетическими квазитекстами позволяет предположить возможность использования сравнения сетей для выявления степени подобия между ними – для классификации генетических событий.

1. Молекулярно-генетические данные

В настоящий момент экспериментальной биологией для основных модельных объектов – человека, мыши, крысы, дрозофилы и некоторых других видов получены строгие оценки десятков тысяч взаимодействий между веществами белковой и небелковой природы в организме изучаемого объекта. Многие из этих взаимодействий объединены в составе цепей или каскадов взаимодействий, имеющих направление, положительные и отрицательные эффекты взаимодействий, различной степени пересечения. Все вместе эти взаимодействия образуют чрезвычайно сложный высокосвязный граф. Тем не менее, такой граф достаточно строго делится на подграфы, представляющие собой каскады метаболических или сигнальных путей. Это связано с тем, что в клетке передача информации зависит преимущественно от «входных» узлов подграфа, и направлена к ограниченному количеству узлов «мишеней» на выходе подграфа, а различные латеральные связи узлов подграфа с другими подграфами определяют интенсивность передачи сигнала по каскаду.

Общий граф, определяющий все возможные варианты взаимодействий веществ в данном организме, существует только гипотетически. Реально в разных типах клеток и тканей, на разных стадиях развития организма, в норме и при патологии работают разные наборы генов. В сравнимых выборках часть генов отличается качественно, т.е. уникальна только для одной из сравниваемых групп, а среди совпадающих генов активность работы может различаться. Такие различия выявляются в количественных показателях состава и/или активности белков или РНК в образце. Суть генетического эксперимента заключается в сравнении наборов данных по альтернативным выборкам, выявление общих для сравниваемых выборок наборов и наборов со значимым изменением активности работы (экспрессии) генов в сторону снижения или увеличения. Анализ структур графов, построенных из таких наборов, позволяет выявить неслучайные процессы роста или падения активности сигнальных или метаболических каскадов.

Упомянутый выше граф, описывающий генетическое событие, может быть представлен перечнем пар «слов» - названий участвующих в эксперименте веществ.

2. Однородная семантическая сеть

Автоматический смысловой анализ естественно-языковых текстов заключается в выявлении ключевых понятий и их взаимосвязей, и ранжировании понятий и связей, то есть в формировании однородной семантической (ассоциативной) сети [Харламов, 2006].

Под семантической сетью N понимается множество несимметричных пар событий $\{<c_i c_j>\}$, где c_i и c_j – события, связанные между собой отношением ассоциативности (совместной встречаемости в некоторой ситуации):

$$N \cong \{<c_i c_j>\} \quad (1)$$

с весовыми характеристиками w_i и w_{ij} , соответственно, ключевого понятия и связи между ключевыми понятиями. В данном случае отношение ассоциативности

несимметрично: $<c_i c_j> \neq <c_j c_i>$.

$$<c_i c_j> \neq <c_j c_i>. \quad (2)$$

Особенностью анализа является итеративная процедура переранжирования частот встречаемости слов текста в их ранг w_i :

$$w_i(t+1) = \left(\sum_{i \neq j} w_i(t) w_{ij} \right) \sigma(\bar{E}). \quad (3)$$

Здесь $w_i(0) = z_i$; $w_{ij} = z_{ij}/z_i$ и $\sigma(\bar{E}) = 1/(1 + e^{-k\bar{E}})$ – функция, нормирующая на среднее значение энергии всех вершин сети \bar{E} ; z_i – частота встречаемости i -го слова в тексте, z_{ij} – частота совместной встречаемости i -го и j -го слов в фрагментах текста; t – номер итерации. Полученная числовая характеристика слов – их смысловой вес – характеризует степень их важности в тексте.

3. Классификация текстов с помощью сравнения сетей

Семантическая сеть N , описанная таким образом (1), может быть представлена как множество так называемых звездочек $<c_i c_j>$ [Харламов, 2015]. Это все пары $\{<c_i c_j>\}$, у которых первое слово одинаковое:

$$N \cong \{z_i\} = \{<c_i c_j>>. \quad (4)$$

Под звездочкой $<c_i c_j>$ понимается конструкция, включающая главное событие c_i , связанное с множеством событий – ближайших ассоциантов в сети $<c_j>$, которые являются семантическими признаками главного события, отстоящими от главного события на одну связь. Связи направлены от главного события к событиям-ассоциантам.

Звездочка с единичными значениями весов событий-ассоциантов называется единичной

звездочкой (звездочкой-ортом). Звездочкой-подпространством называется звездочка, полученная на единичной звездочке введением весов событий w_j :

$$Z \cong \langle c_i \langle w_j c_j \rangle \rangle. \quad (5)$$

Семантическая сеть в терминах этих определений представляет собой декартово произведение подпространств, порождаемых всеми звездочками, входящими в семантическую сеть, полученными на единичных звездочках за счет введения весовых характеристик понятий-ассоциантов:

$$N = Z_1 \times Z_2 \times \dots \times Z_f. \quad (6)$$

Введем скалярное произведение на векторах \bar{c}_i и \bar{c}_j , где угол между векторами понятий соответствующих c_i и c_j : пропорционален весу связи от c_i к c_j : $w_{ij} \in (0 \dots 90^\circ)$.

Площадь треугольника s_i , построенного на векторах \bar{c}_i , \bar{c}_j , развернутых на угол w_{ij} относительно друг друга, будет использована для вычисления степени пересечения сначала звездочек, а потом семантических сетей как совокупностей звездочек.

Под пересечением двух звездочек понимается сумма по всем событиям-ассоциантам данного главного события звездочки пересечений площадей двух треугольников, построенных в плоскости векторов \bar{c}_i и \bar{c}_j , один из которых построен на векторах, развернутых на угол, пропорциональный связи $(w_{ij})_1$ между событиями в одной звездочке, а другой – на угол, пропорциональный связи $(w_{ij})_2$ между теми же событиями в другой, сравниваемой с первой, звездочке. В случае если в одной из звездочек пары, для которой считается пересечение:

$$\begin{aligned} s_{12} &= \langle c_{i_1} \langle c_{j_1} \rangle \rangle \cap \langle c_{i_2} \langle c_{j_2} \rangle \rangle \\ &= \sum_{j=1}^{\max(N_1 N_2)} (s_{j_1} \cap s_{j_2}). \end{aligned} \quad (7)$$

Если не нашлось соответствующего события-ассоцианта, пересечение считается равным 0. Здесь N_1 , N_2 – число ассоциантов в звездочках соответственно 1 и 2.

Тогда под пересечением семантических сетей понимается сумма пересечений звездочек, включенных в эти сети (считая по главным понятиям):

$$S_{12} = \sum_{k=1}^{\max(M_1 M_2)} (s_{j_1} \cap s_{j_2}), \quad (8)$$

где M_1 , M_2 – число звездочек, входящих соответственно в семантические сети N_1 , N_2 .

Под классификацией входного текста можно понимать отнесение семантической сети входного текста N к сети N_n (где $n=1..N$ – число предметных

областей) одной из предметных областей, описываемых соответствующими корпусами текстов. В идеальном случае семантическая сеть текста вкладывается в сеть соответствующей предметной области.

Используя операцию пересечения сетей N_1 и N_2 , определенную выше, мы можем оценивать степень подобия двух сетей $N_1 \cap N_2$ и, тем самым, сравнивать по смыслу (по структуре) тексты. Имея модели предметных областей в виде ассоциативных семантических сетей, мы можем классифицировать входные тексты вычислением степени совпадения (вложения) сети входного текста и сетей предметных областей, относя входной текст к той предметной области, у которой степень совпадения сети с сетью предметной области окажется выше.

Поскольку выше мы показали аналогию между текстами естественного языка и генетическими квазитекстами, представляется возможным использовать приведенный механизм сравнения сетей для классификации генетических квазитекстов.

4. Результаты генетических экспериментов как квазитекст

Аналогия между естественно-языковыми и генетическими текстами позволила предположить возможность использования механизмов анализа естественно-языковых текстов для анализа генетических квазитекстов. Для анализа были использованы так называемые сигнальные или генные сети – сети переходов, описывающие процессы передачи сигналов внутри живой клетки.

Результаты нескольких молекулярно-генетических исследований были использованы для выяснения возможности применения подхода для классификации генетических квазитекстов. В качестве экспериментального материала были использованы результаты анализа экспрессионной активности в ткани опухоли (саркома) и в нормальной ткани.

Так же как и в случае анализа естественно-языковых текстов, представленных однородными семантическими (ассоциативными) сетями, генетические квазитексты, представленные в виде сигнальных сетей – графов, вершинами которых являются названия некоторых веществ, участвующих в генетических процессах, а дуги указывают на взаимосвязи этих веществ в этих процессах – представлялись в виде пар слов $\langle c_i c_j \rangle$, далее – в виде звездочек $\langle c_i \langle c_j \rangle \rangle$.

Сигнальные сети, использованные в эксперименте, были представлены в виде пар «слов», где в качестве слов рассматривались названия белков, концентрации которых анализировались в генетическом эксперименте. Поскольку эти «слова» составляют цепочки при описании генетических процессов (как на рисунке 2), результаты генетического эксперимента могут

быть представлен в виде перечней пар «слов», каждая из которых имеет свой вес – концентрацию, выявленную в результате эксперимента.

Фрагмент одного из таких перечней (саркома) представлен ниже.

(MAPK1 ELK1), (MAPK1 FOS), (MAPK1 MAPK3), (MAPK1 RPS6KA1), (PPP1R3B PPP1R3C), (HDAC9 HSPA5), (TIRAP TRAF6), (STAT5B FOS), (RAP1A BRAF), (RAP1A RASA1), (RAP1A KRIT1), ...

Для того чтобы использовать эти перечни в качестве входных данных для анализа, обычно применяемого для естественно-языковых текстов, необходимо осуществить некоторую их интерпретацию, и некоторое преобразование. Основой механизма анализа естественно-языковых текстов является переранжирование весовых характеристик слов, результат которого зависит от внутренней структуры текста, представленной в виде семантической сети.

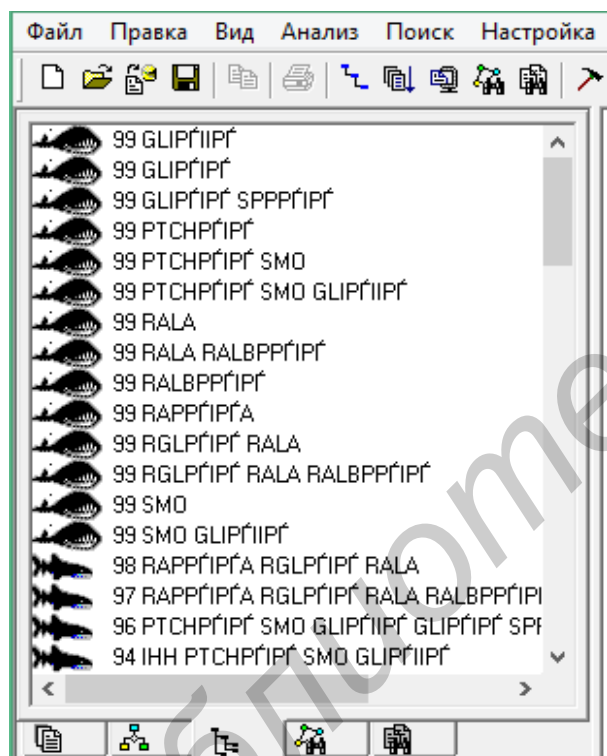


Рисунок 3 – Фрагмент семантической сети генетического квазитекста. Здесь перечислены вершины и показаны их связи. И вершины и связи имеют числовые характеристики. Символы с диакритическими знаками кодируют цифровую информацию при обработке в программе TextAnalyst

Другими словами, необходимо привести исходные данные, полученные в генетическом эксперименте к виду, близкому к виду естественно-языкового текста. Для этого полученные в генетическом эксперименте пары «слов» собираются в «предложения» - цепочки пар слов, описывающие некоторые генетические процессы, от их начала – появления на входе рецептора, до их конца – достижения мишени. Такая сборка осуществляется с помощью «шаблонов» - цепочек пар «слов», стандартных для некоторых групп генетических процессов, представленных в

подсетях, описывающих норму/патологию. Отнормированная величина концентрации используется как характеристика частоты появления пар «слов» в «тексте» - сигнальной сети. «Текст» представлен выявленными вышеописанным способом «предложениями», перечисленными в случайном порядке.

Тогда мы можем анализировать полученный «текст» как в случае естественно-языкового анализа. При этом выявляется частота встречаемости «слов», частота встречаемости пар «слов», строится семантическая сеть «текста», переранжируются весовые характеристики «слов» «текста».

Ну и далее полученные для трех разных случаев генетического эксперимента такие семантические сети (их пример представлен на рисунке 3) сравниваются между собой с целью выявления степени их пересечения.

5. Обсуждение и анализ результатов

Ниже в таблице 1 представлены результаты сравнения трех семантических сетей, соответствующих трем генетическим «текстам», описывающим соответственно, норму и два вида патологии (саркома), которые были получены в результате генетических экспериментов.

Таблица 1 – Результаты сравнения семантических сетей трех разных генетических квазитекстов, соответствующих один – норме, а два других – патологии (саркома).

	Норма	Патология 1	Патология 2
Норма	1,000	29,144	30,334
Патология 1	29,144	1,000	23,150
Патология 2	30,334	23,150	1,000

Конечно, интерпретация сравнения всего трех текстов не очень убедительна, но результаты сравнения, тем не менее, говорят за себя: мы видим разницу в сравнении сетей нормального и двух патологических случаев, и большую степень пересечения при сравнении сетей, представляющих патологию. Невысокие разницы возникают за счет большой общности так называемого house keeping, имеющего место как в норме, так и патологии.

Заключение

Использование механизма сравнения семантических сетей естественно-языковых текстов позволяет сравнивать и другие похожие сетевые структуры, в том числе – сигнальные сети (генетические квазитексты) различных генетических заболеваний, что можно использовать для классификации таких сетей, а следовательно, и для диагностики заболеваний.

Использование данного подхода для сравнения, следовательно, и классификации генетических «текстов» позволит автоматизировать обработку результатов генетических экспериментов, объем которых в известных хранилищах (например, GeneNet) очень велик. Что, в свою очередь, облегчит и улучшит интерпретацию результатов генетических экспериментов.

На самом деле представление генетических квазитекстов в виде перечней пар слов не совсем корректно, кстати, как и при анализе естественно-языковых текстов. Некоторые пары слов различаются видом связи. В дальнейшем вместо пар слов квазитекстов можно будет использовать тройки, включающие помимо пары слов еще и тип связи между ними.

Библиографический список

[Харламов, 2006] Харламов А.А. Нейросетевая технология представления и обработки информации (естественное представление знаний) / А.А. Харламов // М.: Радиотехника, 2006. – 89 с.

[Харламов, 2015] Харламов А.А., Ермоленко Т.В. Нейросетевая среда (нейроморфная ассоциативная память) для преодоления информационной сложности. Поиск смысла в слабоструктурированных массивах информации. Часть II. Обработка информации в гиппокампе. Модель мира. / А.А. Харламов, Т.В. Ермоленко // Информационные технологии, 2015, № 12, С. 883—889

HOMOGENOUS SEMANTIC NETWORK FOR GENETIC ANALYSIS RESULT CLASSIFICATION

Kulikov A.M. *, Kharlamov A.A. **

**Koltzov Institute of Developmental Biology of Russian Academy of Sciences, Moscow*

amkulikov@gmail.com

***Institute of Higher Nervous Activity and Neurophysiology of Russian Academy of Sciences, Moscow*

kharlamov@analyst.ru

In the work mechanism of text semantic networks comparison in task of disease diagnostic on signalling network using is shown. One can calculate texts semantic similarity by calculating volume of their networks crossection. Homogenous network consists of nodes and their connections with their weight characteristics. The characteristics may be more exact by their renormalization on the basis of text n-gram model. The examples of such genetic quasi-texts network crossection calculation of two different diseases.

Key words: homogenous semantic networks, signaling networks, texts comparison, text classification.

Introduction

There is an analogy between natural language texts and genetic code texts (genetic quasi-texts) because of

natural language text semantic network and genetic quasi-text signaling network similarity. Signaling network also can be represented as list of quasi-word pares.

Main Part

Currently, estimations of the interactions between compounds of protein and non-protein nature in the organism of the studied object have been obtained in the field of experimental biology. Many of these interactions are integrated as part of chains or cascades of interactions. These chains have a direction, positive and negative effects of the interactions of varying degrees of intersection. Together, these interactions form a highly complex connected graph, which is divided into subgraphs representing cascades of metabolic and signaling pathways.

A full graph defining all possible interactions of the substances in the organism exists only hypothetically. As a matter of fact, in different types of cells and tissues at different stages of development of an organism, in the normal and pathologic different sets of genes are working. In comparing samples, a part of genes differs qualitatively, i.e. it is unique for only one of the groups being compared. The essence of the genetic experiment is to compare data sets of alternate samples, identification of the common sets for compared samples and sets with a significant change in the activity of gene work (expression). An analysis of the graph structure constructed from such data sets allows revealing nonrandom processes of growing or falling down in activity of signaling or metabolic cascades.

The above mentioned graph describing genetic events may be represented by a list of pairs of "words" - the names of substances participating in the experiment.

Automatical semantic natural language text analysis extract key words and their relationships from the text. That is why the key words can be integrated into homogenous semantic network. The nodes and arcs of the network have their semantic weights.

Such networks can be compared by their crossection calculation – calculation of the nodes and relationships weighting coefficients in the crossection.

Because of natural language text and genetic quasi-text analogy one can use the mechanism for genetic quasi-texts comparing (and for their classification). In the paper results of such comparing signaling networks of two diseases represented.

Conclusion

One can use a natural language text semantic network comparison mechanism and for another network structures comparing also, genetic quasi-texts signaling networks different diseases for example. That is why one can use it and for their diagnostic also. The exactness of the analysis may be can be shifted by using not word pares networks but word triple ones where besides the word pare has a type of relationship too.



УДК 004.822:514

КОНСТРУИРОВАНИЕ ОТНОШЕНИЯ ПСЕВДОХРОНОЛОГИЧЕСКОГО ПОРЯДКА В ТЕРМИНОЛОГИЧЕСКИХ СЕТЯХ

Мальковский М.Г., Соловьев С.Ю.

*Факультет вычислительной математики и кибернетики,
Московский государственный университет имени М.В. Ломоносова,
г. Москва, Россия*

malk@cs.msu.su

soloviev@glossary.ru

Рассматривается жизненный цикл терминов научной и деловой лексики. Вводится понятие хронологической шкалы терминов. Рассматривается задача восстановления хронологических связей на заданном структурированном множестве терминов. Предлагается для восстановления использовать приемы научного познания, которые с одной стороны используются для конструирования терминосистем, а с другой стороны позволяют зафиксировать порядок появления терминов в научно-деловом обороте. Приводятся образцы продукционных правил, ориентированных на выявление хронологических связей. Предлагается метод конструирования хронологической шкалы по результатам функционирования продукционной системы.

Ключевые слова: термин, терминологическая сеть, хронология, продукционная система.

Введение

В научном и деловом обращении термины, хотя и существуют относительно самостоятельно, главным образом используются для именованя понятий. Известно, что с течением времени¹ термины:

- вводятся в обращение (появляются);
- замещаются синонимами;
- изменяют смысл;
- выбывают из обращения (устаревают).

Так в 1990-е годы появились новое понятие и новый термин “Евро”. Примером типичного замещения является вытеснение из научно-технических текстов термина “ЭВМ” в пользу термина “Компьютер”. Если под изменением смысла понимать изменение объема понятия, то термин “Еврозона” изменялся несколько раз. За последние 20 лет в категорию устаревших перешли термины “Ленинград”, “Единый социальный налог” и “Евродоллар”, а также устарела целая терминосистема “Научный коммунизм”. Заметим, что со временем устаревшие термины (если не принять меры) могут либо породить проблему неоднозначного понимания накопленных электронных текстов, либо

загромождать веб-пространство, в котором отсутствует хронология документов.

Разработка глобальных терминологических баз знаний существенно затрудняется необходимостью моделировать хронологические связи и временные характеристики терминов. В последующих рассуждениях мы исходим из того, что каждый термин однажды изобретается человеком и, после определенного периода “борьбы за признание”, включается в активный научно-деловой оборот. В некоторых случаях возникновение термина удается датировать весьма точно. Так, термин “Наркоманическая зависимость” как “зависимость больного от присутствия или отсутствия в его организме алкоголя” введен ВОЗ в 1964 году [Дьяченко, 2002]. В других случаях, как это произошло, скажем, с “Ноосферой” В.И.Вернадского, история зарождения термина растягивается на десятилетия.

Настоящая работа посвящена задаче восстановления по косвенным признакам хронологии появления терминов, вопросы изменения смысла и устаревания терминов практически не рассматриваются.

1. Терминологические сети

Терминологическая сеть [Мальковский и др., 2012] есть продукт деятельности коллектива

¹ Жизненный цикла термина в пределах одной терминосистемы называется терминогенезом [Кокорина, 2011].

научных редакторов, занимающихся систематизацией терминов и их определений. С формальной точки зрения терминологическая сеть представляет собой ориентированный граф, узлы которого соответствуют терминам², а дуги – экземплярам бинарных отношений из заранее фиксированного набора допустимых отношений. В терминологических сетях:

- каждый узел A можно рассматривать как четверку $\langle N, T, S, D \rangle$, где

$N = N(A)$ – необязательное понятийное имя,

$D = D(A)$ – определение термина,

$T = T(A)$ – выделенный термин из множества синонимов $S = S(A)$.

- каждая дуга $(A, B)_R$ есть упорядоченная пара узлов A и B , помеченная символом отношения R ; если для дуги не оговаривается тип отношения, то в записи такой дуги метка опускается;

- набор допустимых отношений обязательно содержит родовидовое отношение P , которому соответствуют дуги $(A, B)_P$, где A – вид, B – род;

- понятийным узлом называется узел, в который заходит хотя бы одна дуга;

- потомками понятийного узла B называются узлы A , связанные с B дугами (A, B) ;

- предками понятийного узла A называются узлы B , связанные с узлом A дугами (A, B) ;

- каждый понятийный узел A имеет уникальное имя $N(A)$, которое также служит наименованием понятия; как правило, наименование понятия есть общее наименование составляющих объектов.

Выявление терминов, пригодных для образования понятийных узлов, а также установление отношений между терминами составляет основу интеллектуальной деятельности научных редакторов. В своей работе редакторы используют определения терминов, справочную литературу и собственные знания.

При анализе терминологических сетей существенно используются специальные отношения между терминами, узлами и дугами. Приведем эти отношения.

Будем говорить, что (многословный) термин x подчинен (многословному) термину y , если термин x является развитием термина y . Примерами отношения подчиненности являются следующие пары терминов:

$x =$ “Промышленные аварии” и $y =$ “Аварии”;

$x =$ “Централизованная библиотечная система” и

$y =$ “Библиотечные системы”;

$x =$ “Скорость света в вакууме” и $y =$ “Вакуум”.

Отношение подчиненности для пары терминов, связанных дугой, отвечает синтаксическому способу терминоподчинения [Гринев-Гриневиц, 2008]. С формальной точки зрения термин x , состоящий из слов x_1, \dots, x_g , подчинен термину y , состоящему из

слов y_1, \dots, y_h , если существует однозначная функция

$$f : \{y_1, \dots, y_h\} \rightarrow \{x_1, \dots, x_g\} \quad (1)$$

такая, что для всех $i = 1, \dots, h$ слова y_i и $f(y_i)$ отличаются формальными суффиксами.

2. Хронологическая шкала

Будем полагать существование порядковой хронологической шкалы терминов, позволяющей судить о порядке их появления. Строго говоря, шкала моделируется симметричным и транзитивным бинарным отношением предшествования на множестве узлов терминологической сети: если A и B два узла, то запись $A \leq B$ означает, что термин A появился не позднее термина B . Соответственно отношение $A = B$ определяется как $(A \leq B) \ \& \ (B \leq A)$.

Понимая под отношением множество упорядоченных пар, будем называть хронологической связью одну пару из отношения. Примерами хронологических связей между терминами являются:

“Здание” \leq “Кровля”,

“Алкоголь” \leq “Наркоманическая зависимость”,

“Облигация” \leq “Дисконтная облигация”,

“Дисконтная облигация” \leq

\leq “Облигация с глубоким дисконтом”,

“Облигация” \leq “Облигация с глубоким дисконтом”.

Хронологическая шкала терминов открывает новые возможности для конструирования, анализа и моделирования эволюционных процессов в терминосистемах.

3. Подход к восстановлению хронологических связей

По сути дела терминологическая сеть представляет собой частично формализованную структуру, в которой бинарные связи между терминами фиксируются дугами семантической сети, а остальная информация содержится в определениях терминов (в словарных статьях). Кроме того, принятый способ организации данных позволяет относить к формальной части структуры синонимы и аббревиатуры терминов.

Терминологическая сеть, будучи существенно статической структурой данных, тем не менее позволяет с определенной достоверностью установить некоторые связи на хронологической шкале терминов. Основной подход к реконструкции хронологических связей состоит в том, что при конструировании [новых] понятий и соответствующих им терминов существенно используются общенаучные методы анализа и абстрагирования, а также приемы сравнения и аналогии [Степин, 2003]. Применительно к понятиям упомянутые методы и приемы имеют особенности, позволяющие делать выводы о хронологических связях.

² Термины в свою очередь являются наименованиями понятий.

Анализ понятий – метод конструирования новых понятий в результате разложения известного понятия (α -понятия) на его составляющие. В терминологической сети новым подпонятиям соответствуют потомки α -узла. С точки зрения хронологических связей новые понятия появляются позже α -понятия.

Абстрагирование – метод конструирования новых понятий (β -понятий) посредством отвлечения от несущественных деталей, присущих известным понятиям (γ -понятиям). В терминологической сети β -понятиям соответствуют предки γ -узлов. С точки зрения хронологических связей β -понятия появляются позже γ -понятий.

Сравнение понятий – прием выявления общих признаков заданных понятий.

Вывод по аналогии – прием выявления новых признаков понятий, при котором на основе сходства понятий по одним признакам заключается их сходство по другим признакам.

4. Правила восстановления хронологических связей

В качестве исходных данных в задаче восстановления хронологических связей используется связный фрагмент терминологической сети отвечающей некоторой терминсистеме. Для выявления отдельных хронологических связей предлагается использовать продукционные правила вывода, некоторые из которых приводятся далее.

Правило 1. ЕСЛИ (а) $(A, B)_P$ – дуга, И (б) множества $S(A)$ и $S(B)$ содержат только термины $T(A)$ и $T(B)$, И (в) $T(A)$ подчинен $T(B)$, ТО $B \leq A$.

Правило 2. ЕСЛИ (а) B – узел, И (б) для всех A_i (таких, что $(A_i, B)_P$ дуга терминологической сети) множество $S(A_i)$ содержит термин s_i подчиненный термину $T(B)$, ТО $B \leq A_i$.

Правило 3. ЕСЛИ (а) $(A, B)_P$, $(B', B)_X$, $(A', A)_Y$, $(A', B')_P$ – дуги, причем $X = Y \neq P$, И (б) множества $S(A)$ и $S(B)$ содержат только термины $T(A)$ и $T(B)$, И (в) $T(A)$ не подчинен $T(B)$, ТО $A \leq B$ (см. рис.1 левая часть).

Правило 4. ЕСЛИ (а) $(A, B)_P$ – дуга, И (б) множества $S(A)$ и $S(B)$ содержат только термины $T(A)$ и $T(B)$, И (в) $N(A)$ не подчинен $N(B)$, ТО $A \leq B$.

Правило 5. ЕСЛИ (а) $(A, B)_P$ – дуга, И (б) существует цепочка дуг $(B, B^{(1)})_P, \dots, (B^{(n-1)}, B^{(n)})_P$, И (в) $T(B^{(n)}) = \text{“Идеализированный объект”}$, ТО $A \leq B$.

Правило 6. ЕСЛИ (а) (A, B) – дуга, И (б) для любой пары терминов (s_A, s_B) из $S(A) \times S(B)$ термин s_A не подчинен s_B , ТО $A \leq B$. (рис.1 правая часть).

Правило 7. ЕСЛИ (а) $(A, B)_X$, $(A', B)_X$ – дуги, с одинаковыми метками, И (б) в сети отсутствуют иные дуги, исходящие из A и A' , И (в) $A \leq B$, ТО $A' \leq B$.

Правило 8. ЕСЛИ (а) $(A, B)_X$, $(A', B)_X$ – дуги с одинаковыми метками, И (б) в сети отсутствуют иные дуги, исходящие из A и A' , И (в) $B \leq A$, ТО $B \leq A'$.

Замечание 1. Приведенные правила являются примерами продукций, ориентированных на формирование хронологической шкалы терминов для выбранного фрагмента терминологической сети. Естественно, правила такого рода должны иметь числовые коэффициенты уверенности (CF), значения которых в конечном итоге зависят от выбранной схемы правдоподобных рассуждений [Моросанова и др., 2012], а на выбранном уровне описания можно зафиксировать лишь общие свойства этих коэффициентов:

- правила 1-8 не являются абсолютно достоверными,
- CF правила 3 больше CF правила 1 и т.д.

Результатом функционирования продукционной системы является матрица коэффициентов уверенности фактов вида $A \leq B$. Матрица попарных оценок открывает возможность использовать для построения искомой порядковой шкалы алгоритмы многомерного шкалирования [Терехина, 1986]. По построению в матрице содержатся незаполненные позиции, количество которых можно сократить посредством транзитивного замыкания:

ЕСЛИ $A_1 \leq A_2$ [CF=a] И $A_2 \leq A_3$ [CF=b],
ТО $A_1 \leq A_3$ [CF=comb(a,b)],

где *comb* – операция комбинирования коэффициентов уверенности.

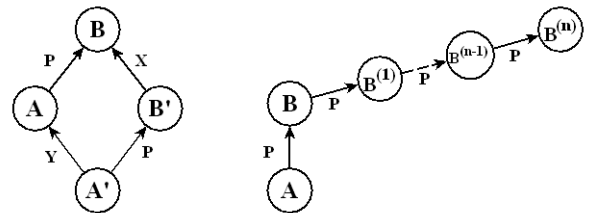


Рисунок 1 – Условия правил 3 и 5

Замечание 2. Правила 1 и 2 исходят из того, что понятия A и A_i появились в результате анализа ранее существовавшего α -понятия B . Предполагается, что анализ состоит в выделении подпонятий и связанных с ними сервисными понятиями.

Замечание 3. Правила 3-6 исходят из того, что β -понятие получено в результате абстрагирования как новое знание, обобщающее ранее известные γ -понятия A и B' . Особое место занимает правило 5, в котором примененный метод познания идентифицируется смыслом узла $B^{(n)}$.

Замечание 4. Правила 7 и 8 соответствуют выводам по аналогии, в которых хронологическая связь, полученная тем или иным способом для узлов дуги $(A, B)_X$, распространяется на узлы дуги $(A', B)_X$.

Замечание 5. Некоторые дуги вида $(A, B)_P$, удовлетворяют условиям правил 4-6. При этом в

сети возможна ситуация, зафиксированная в следующем правиле:

Правило 9. ЕСЛИ $(A, B)_P$ И $(A', B')_P$ – различные дуги, удовлетворяющие условиям правил 4-6, ТО $A \leq B$ и $A' \leq B'$.

Условие правила 9 фиксирует прием сравнения понятий A и A' . Особенность правила 9 состоит в том, что его коэффициент уверенности выше коэффициентов каждого из правил 4-6.

Замечание 6. Поскольку введенное отношение хронологического предшествования является симметричным, то для любого фрагмента терминологической сети искомая хронологическая шкала может быть построена как шкала одновременно возникших терминов (терминов, связанных отношением $=$). Естественно, такое решение не является информативным, однако оно открывает подход к построению нетривиальных хронологических шкал дивизимным методом [Жамбю, 1988] разделения множества терминов на хронологически упорядоченные и более мелкие составляющие.

Замечание 7. Некоторые термины (и связанные с ними понятия) имеют точные дату возникновения. Такие термины представляют особую ценность, поскольку их можно использовать для отбраковки неверно построенных хронологических связей.

Заключение

1. Описанный подход не влияет на топологию терминологической сети: все узлы и дуги сети остаются без изменений, что ограничивает область применения описанной методологии. Моделирование процессов замещения, изменения и устаревания терминов требует дополнительных исследований. Представляется, что в первую очередь следует определиться с понятийным аппаратом терминогенеза.

2. Предложенный подход в лучшем случае позволяет конструировать псевдохронологические связи, лишь отчасти совпадающие с истинными связями. Исчерпывающий подход к решению задачи конструирования хронологической шкалы состоит в поддержке специальной структуры, описывающую эволюцию каждого конкретного термина. Однако это невозможно по двум причинам: (а) задача нуждается в точной и плодотворной постановке, и (б) задача является слишком трудоемкой.

Библиографический список

- [Гринев-Гриневиц, 2008] Введение в терминографию / С.В.Гринев-Гриневиц. – М.: ЛИБРОКОМ, 2009.
- [Дьяченко, 2002] Словарь авторских терминов, понятий и названий в медицине / А.П.Дьяченко. – М.: Триада-Х, 2002.
- [Жамбю, 1988] Иерархический кластер-анализ и соответствия / М.Жамбю. – М.: Финансы и статистика, 1988.
- [Мальковский и др., 2012] Мальковский М.Г., Терминологические сети / М.Г.Мальковский, С.Ю.Соловьев // Материалы конференции OSTIS-2012. – Мн.: БГУИР, 2012. С. 77-82

[Моросанова и др., 2012] Моросанова Н.А., Формальные свойства схемы Шортлиффа / Н.А.Моросанова, С.Ю.Соловьев // Управление большими системами, выпуск 36. - М.: ИПУ РАН, 2012. С. 5-38

[Кокорина, 2011] Терминосистема концептуальной области «Археология» / Ю.Г.Кокорина. – М.: ЛИБРОКОМ, 2011.

[Степин, 2003] Теоретическое знание / В.С.Степин. – М.: Прогресс-Традиция, 2003.

[Терехина, 1986] Анализ данных методами многомерного шкалирования / А.Ю.Терехина. – М.: Наука, 1986.

THE DESIGNING OF PSEUDO CHRONOLOGICAL ORDER RELATION IN THE TERMINOLOGICAL NETWORKS

Malkovsky M.G., Soloviev S.Y.

Lomonosov MSU, Moscow, Russia

malk@cs.msu.su

soloviev@glossary.ru

In this article we

- consider we consider the terms life cycle;
- introduce the notion of term chronological scale;
- consider the problem of reconstructing the historical ties in a given structured sets of terms;
- propose to use for the restoration techniques of scientific knowledge, which on one hand is used for the construction of terminological, on the other hand allow you to fix the order of appearance in terms of scientific and business turnover;
- give examples of production rules, aimed at identifying historical ties;
- propose a method of constructing a chronological scale according to the functioning of a production system.

Introduction

The problem of the evolution of terms is seen as a difficult task, which has a subtask of designing a term chronological scale.

Main Part

Object of research are terminology network, which reflect the relationship between terms. The terms chronological scale must satisfy the axioms of symmetric, and transitive. The method of building chronological ties between the terms is based on the recognition methods of scientific knowledge, which were the basis of terminological structure. These methods are called: the method of analysis of concepts, the method of abstraction, the method of comparison and analogy method.

Methods of scientific knowledge generate production rules, some of which are presented in the article. The production system builds a matrix of comparative assessments of terminology ties. Justification production rules boil down to different variants of the method of scientific knowledge. Ultimately, the task of designing terminology scale is reduced to the hierarchical clustering.

Conclusion

We give a brief overview of the problem of modeling subtasks of terms evolution.



УДК 004.822:514

АЛГОРИТМ ВЫДЕЛЕНИЯ БИНАРНЫХ ОТНОШЕНИЙ В ТЕРМИНОЛОГИЧЕСКОЙ СЕТИ

Тузикова А.В.

*Факультет вычислительной математики и кибернетики,
Московский государственный университет имени М.В. Ломоносова,
г. Москва, Россия*

kabaylova@yandex.ru

В работе рассматривается возможность частичной автоматизации технологии построения терминологических сетей. Предлагается метод выявления семантических связей, основанный на использовании метрики редактирования для вычисления расстояний между понятийными вершинами. Работа алгоритма иллюстрируется на модельной терминологической сети.

Ключевые слова: терминологическая сеть; понятие; метрика редактирования.

Введение

В современном мире ощущается потребность в структурном представлении данных, удобном для их исследования. Одним из способов структуризации данных является "Универсальное терминологическое пространство" (УТП) [Мальковский и др., 2002], проект которого подразумевает построение семантической сети из проблемно-ориентированных взаимосвязанных глоссариев. Развитие УТП обеспечивается редакторами-экспертами, пополняющими терминологическую сеть путем обработки глоссариев, выделения понятий и установления бинарных семантических связей.

Проблема качественного редактирования терминологической сети порождает задачу автоматизации этой деятельности.

1. Введение в терминологические сети

Терминологическая сеть [Мальковский и др., 2012] относится к классу семантических сетей [Sofa, 1992] и представляет собой граф, вершинами которого являются определения терминов, а ребрами – бинарные отношения между ними. Фактически, вершина представляет собой структуру данных, содержащую информацию о термине: синонимы, определение, а так же дополнительные сведения о термине или о связанных с ним вершинах. Ребрам графа соответствуют два допустимых типа отношений: "это-есть" и

"относится-к", к которым сводятся все остальные бинарные отношения.

При добавлении новых данных в существующую терминологическую сеть редактор обрабатывает определенное количество новой информации, постоянно взаимодействуя с сетью значительных размеров. Для облегчения деятельности редактора предлагается алгоритм выделения потенциально возможных связей между включаемой в сеть понятийной вершиной и вершинами расширяемой терминологической сети.

2. Алгоритм автоматического выявления связей

В основу алгоритма выявления связей положено предположение о наличии в определении или в словарной статье достаточной информации для определения возможных связей с понятийными вершинами терминологической сети.

Исходное определение можно рассматривать как строку (см. раздел 2.1). Понятийную вершину так же можно рассматривать как особым образом структурированную строку. Рассматривая определение и понятийную вершину как строки, можно определить расстояние между ними с использованием метрики редактирования [Деа и др., 2008]. В свою очередь, если рассматривать строку как набор слов, тогда в общем наборе операций редактирования достаточно рассматривать лишь операции замены и вставки-удаления символов, что соответствует метрике Левенштейна [Левенштейн, 1965].

Метрика Левенштейна применяется неоднократно для вычисления расстояний между парами слов из сравниваемых строк. Для принятия решения о значимости вычисленного расстояния используется порог, который задается величиной Val , устанавливаемой опытным путем и являющейся единственным параметром при определении близости понятийных вершин.

Таким образом, задача выявления связей сводится к задаче определения расстояний между строками. Упомянутое расстояние предлагается вычислять как:

$$Dist = \sum_{succNum} 2^{k_{succNum}} + \frac{imp}{10^2} - \frac{unimp}{10^5}. \quad (1)$$

$$k_{succNum} = Val - \max\{1, pDist\}. \quad (2)$$

В формуле (1) суммирование ведется по всем новым расстояниям, являющимся существенными в соответствии с порогом Val ; imp – число пар слов, расстояние между которыми влияет на результат сравнения; $unimp$ – число пар слов, расстояние между которыми не влияет на результат сравнения; 10^2 и 10^5 – постоянные коэффициенты, выбранные таким образом, чтобы можно было свести в одно число количество существенных результатов, количество несущественных результатов, а также вычисленное с использованием метрики Левенштейна расстояние; $pDist$ – вычисленное значение расстояния для пары слов. Взятие максимума позволяет рассматривать идентичные слова как просто близкие, что дает возможность комбинациям слов конкурировать с совпадающими словами при вычислении их вклада в расстояние $Dist$. Следует отметить, что $k_{succNum} \geq 0$, так как к моменту вычисления гарантируется существенность значения переменной $pDist$.

Предложенный подход к определению близости двух понятийных вершин позволяет учесть не только влияние результатов отдельных сравнений, но и их качество. Так, словосочетания “свободные места” и “свободными местами” окажутся более близкими, чем “свободные места” и “свободные стулья”, что соответствует сути дела. Если же установить значение параметра $Val = 0$ (что соответствует абсолютной идентичности сравниваемых слов), сложится обратная ситуация.

В силу выбора упомянутого числового коэффициента 10^5 , число несущественных сравнений оказывает влияние лишь на порядок перечисления выявленных кандидатов, предлагаемых редактору.

2.1. Входные и выходные данные

Алгоритм выявления связей позволяет модифицировать первоначально заданную терминологическую сеть. В качестве входных данных алгоритм использует выделенное

пользователем определение. Приведем типичные примеры исходных определений, взятых из музыкального словаря [Риман, 2008] и в модифицируемой сети не представленных.

Пример 1

Бассанелло – ныне вышедший из употребления деревянный духовой инструмент, родственник фаготу. У него был двойной язычок, заключенный в воронкообразный мундштук и изогнутая шейка (в виде буквы S). Бассанелло делался трех различных величин (басовый, теноровый и дискантовый). В старинных органах были также регистры Bassanelli.

Пример 2

Чибыза (чебызга) – дудка из камыша или из дерева; народный музыкальный инструмент киргиз-кайсаков.

Пример 3

Магадис – струнный инструмент древних греков, похожий на арфу, число струн коего доходило до 40; одно место в псевдо-аристотелевых проблемах указывает на то, что на магадисе играли мелодию октавами.

Выходными данными алгоритма являются термины, соответствующие понятийным вершинам терминологической сети, которые могут быть связаны с новой понятийной вершиной, формируемой на основе введенного определения, связью одного из типов “это-есть” и “относится-к”.

2.2. Идея алгоритма выявления связей

Алгоритм выявления связей состоит из двух этапов.

Этап 1. [Определение расстояний]. Определить расстояния между введенным термином, его определением и соответствующей информацией из понятийных вершин с помощью метрики Левенштейна. Сравнения элементов, представленных в виде строк, следует провести между парами “определение” – термин, соответствующий понятийной вершине; и “термин” – информация, полученная из понятийной вершины. Остальными способами комбинирования пренебречь вследствие малой вероятности возникновения связей “это-есть” и “относится-к” между практически идентичными понятиями.

Этап 2. [Добавление кандидатов]. Добавить подходящие вершины, а так же вычисленные для них значения к числу кандидатов на добавление связей. Новые “варианты” во множество кандидатов добавлять лишь взамен самого худшего из текущих.

По результатам выполнения этапов 1 и 2 формируется набор из ограниченного постановкой конкретной задачи числа кандидатов. После обработки всех вершин сети работа алгоритма завершается, результатом является набор понятийных вершин терминологической сети,

которые могут быть связаны отношениями “это-есть” или “относится-к” с новой, сформированной на основе предложенных для обработки данных, понятийной вершиной.

После завершения поиска кандидатов полученные результаты предоставляются редактору для принятия и документирования в сети соответствующих решений, так же как информация о предлагаемых понятийных вершинах. Таким образом, алгоритм облегчает выявление связей между вершинами, хотя и оставляет задачу выбора редактору.

Результатом взаимодействия пользователя с системой является расширенная терминологическая сеть, в состав которой включены сформированная вершина и выбранные для добавления связи.

Следует отметить, что алгоритм не предусматривает усечение или нормализацию слов, то есть сравнения могут проводиться между различными формами одного и того же слова.

3. Применение к конкретной предметной области

Для иллюстрации работы метода выявления связей используется сеть, соответствующая области “Музыка”, содержащая 847 понятийных вершин и входящая в состав glossary.ru [Мальковский и др., 2002].

Для модельной области проведено расширение терминологической сети включением в нее новых понятийных вершин. В таблицах 1-3 приведены полученные численные результаты для небольшого числа новых данных, добавленных в сеть с использованием алгоритма. Входные данные приведены в примерах 1-3. В таблице 4 предложены к рассмотрению результаты анализа данных из таблиц 1-3 редактором-экспертом.

В таблицах 1-3 столбец “Dist” содержит вычисленное значение семантической близости введенного определения и соответствующей понятийной вершины; в столбце “Вершины” приведены по два предлагаемых варианта для рассматриваемых входных данных.

Таблица 1 – Расчетные значения расстояний для модельной области “Музыка” для термина “бассанелло” – Пример 1

Вершина	<i>imp</i>	<i>unimp</i>	<i>Dist</i> , до 10^{-3}
Двойной язычок	2	80	16.019
Духовые музыкальные инструменты	2	121	12.019

Для термина “бассанелло” выбор вершины “двойной язычок” обусловлен наличием соответствующего словосочетания в тексте определения в именительном падеже, что делает его

более ценным с точки зрения алгоритма, чем понятие “духовые музыкальные инструменты”.

Таблица 2 – Расчетные значения расстояний для модельной области “Музыка” для термина “чибыза” – Пример 2

Вершина	<i>imp</i>	<i>unimp</i>	<i>Dist</i> , до 10^{-3}
Народные музыкальные инструменты	3	30	24.030
Духовые музыкальные инструменты	2	31	16.020

Для термина “чибыза” можно наблюдать аналогичную ситуацию. Выбор вершины “Народные музыкальные инструменты” основан на непосредственном наличии в тексте определения словосочетания “народный музыкальный инструмент”. “Духовые музыкальные инструменты” имеют меньшее значение в связи с меньшим числом слов в названии вершины, встречающихся в тексте определения.

Таблица 3 – Расчетные значения расстояний для модельной области “Музыка” для термина “магадис” – Пример 3

Вершина	<i>imp</i>	<i>unimp</i>	<i>Dist</i> , до 10^{-3}
Струнные музыкальные инструменты	2	79	16.019
Струнные щипковые музыкальные инструменты	2	106	16.019

Для термина “магадис” имеет место влияние количества информации (числа слов) в понятийных вершинах на упорядоченность результатов в соответствии с (1).

В таблице 4 обозначения “О” и “Э” соответствуют типам связей “относится-к” и “это-есть”.

Таблица 4 – Результаты работы эксперта-редактора

Обрабатываемые данные	Вершина	Тип связи	Выбор
Пример 1	Духовые музыкальные инструменты	Э	Система
	Народные музыкальные инструменты	Э	Система
Пример 2	Духовые музыкальные инструменты	Э	Система
	Древнегреческая	О	Редактор

	культура		
	Струнные музыкальны е инструменты	Э	Система

При анализе результатов, предлагаемых алгоритмом, для выборки в 40 определений новых терминов, 72,5% случаев были оценены как удовлетворительные и не требующие глубокого вмешательства редактора-эксперта. Таким образом, работа редактора при использовании представленного алгоритма в большинстве ситуаций сводится к выбору одного из предложенных вариантов.

Заключение

Предложен метод предварительной настройки терминологической сети, основанный на использовании метрики редактирования для выделения потенциальных связей между новыми данными и вершинами сети.

Использование метрики редактирования позволяет свести процесс поиска связей между понятийными вершинами к определению расстояния между строками.

Описанный метод дает релевантные результаты и позволяет за счет частичной автоматизации снизить нагрузку на редакторов терминологической сети при ее расширении и повысить объективность.

Предложенный подход может быть использован в качестве основы при создании программы для введения в сеть новых данных порциями большего объема.

Библиографический список

[Деза и др., 2008] Деза Е.И., Деза М.-М. Энциклопедический словарь расстояний / Елена Деза, Мишель-Мари Деза – пер. с англ. В.И. Сычева; Моск. гос. пед. ун-т; Нормальная высш. шк., Париж. – М.: Наука, 2008. – с. 178-186.

[Левенштейн, 1965] Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академии Наук СССР, т. 163, №4 – М.: Наука, 1965. – с. 845–848.

[Мальковский и др., 2002] Мальковский М.Г., Соловьев С.Ю. Универсальное терминологическое пространство. // Труды Международного семинара Диалог'2002 "Компьютерная лингвистика и интеллектуальные технологии", т.1. М.: Наука, 2002, с. 266-270.

[Мальковский и др., 2012] Мальковский М.Г. Терминологические сети / М.Г.Мальковский, С.Ю.Соловьев // OSTIS-2012. Материалы конференции. С. 77-82.

[Риман, 2008] Риман Г. Музыкальный словарь [Электронная версия] / Пер.: Б. Юргенсон. - М.: Директмедиа Паблишинг, 2008. - 10440 с.

[Sofa, 1992] John F. Sowa Semantic Networks / Encyclopedia of Artificial Intelligence, second edition, Wiley, New York, 1992.

ALGORITHM OF BINARY RELATIONS EXTRACTION FROM TERMINOLOGICAL NETWORK

Tuzikova A.V.

Lomonosov MSU CS department, Moscow, Russia

kabaylova@yandex.ru

In the article the possibility of partial automatization of terminological networks forming process is considered. The algorithm for identification semantic relations is proposed. It is based on using editing metrics for determining distances between concept nodes of the terminological network.

Introduction

There is lot of unstructured information in the world around us nowadays. It is necessary to make it structured. One of the ways to do that is the universal terminological space (UTS). This project implies creation of the semantic network over the set of problem-oriented glossaries. In its semantic network two types of relations are used, they are "it-is" and "belongs-to". The extension of the existing network by including new concept nodes and binary relations in it makes it necessary to automate the activity of scientific editors. As a result, the solution described at the article appeared.

Main Part

The main idea of the suggested algorithm is to find distances between the concept node, created by processing new data, and nodes of the existing terminological network that is being extended. The output of the algorithm is the information about several nodes that can be connected with the new one.

The algorithm is based on using editing metrics. The Levenshtein distance was chosen.

Let's think about new data and existing nodes as about strings. Then let's look at strings (that obviously are sentences) as a set of words. That approach allows to calculate distances between new data and nodes as distances between separate words that are summed by several rules to find required value.

In the article there are results concerning the data connected with the Music area.

Conclusion

The proposed method allows to find semantic relations between new data and nodes of the extensible terminological network. Thus, it contributes to decrease loading on the editors and to increase the objectivity.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891

АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ ДЕРЕВА КЛАССОВ ОНТОЛОГИИ И ИХ ЭКЗЕМПЛЯРОВ НА ОСНОВЕ ТЕКСТА ПО ПРЕДМЕТНОЙ ОБЛАСТИ

Андреев И.А., Бексаева Е.А., Желепов А.С., Клейн В.В., Серков И.П.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

ares-ilya@ya.ru

ekaty@clukva.ru

a.zheleпов@yandex.ru

vikklein93@gmail.com

bivo@bivo.ru

В данной статье рассматривается алгоритм формирования дерева классов онтологии и их экземпляров на основе текста предметной области. Описывается техническая реализация программной системы, входные и выходные данные, оценивается результат работы реализованного алгоритма. Предлагаемый подход основан на применении лингвистических шаблонов и тезаурусов по предметным областям.

Ключевые слова: онтология; автоматическое формирование онтологии; дерево классов; экземпляры классов.

Введение

Задача формирования онтологии предметной области связана с задачами информационного поиска [Можжерина, 2011], извлечения терминологии [Андреев и др., 2014а], интеграции разнородных баз данных [Соловьев и др., 2006]. Использование онтологии в качестве инструмента при решении задачи предполагает привлечение эксперта по предметной области, его обучению работе с онтологиями с последующим процессом ручного формирования самой онтологии.

Недостатками данного процесса является увеличение материальных и временных затрат на решение задачи, возможные неточности в результате работы эксперта. Таким образом, автоматизация процесса формирования онтологии, имеющая целью избавление от упомянутых недостатков, является актуальной задачей.

В данной статье приводится алгоритм автоматизации процесса формирования онтологии предметной области на основе текстов по соответствующей предметной области. В основу предлагаемого алгоритма положены преимущественно лингвистические методы.

В качестве базы знаний используется тезаурус по соответствующей предметной области. Особенностью алгоритма является зависимость

результата от содержания, как тезауруса, так и текста, подаваемого на вход. Тезаурусы по предметным областям предварительно составляются экспертом-лингвистом, что делает их содержание зависимым от субъективного фактора, уровня подготовки эксперта, а также актуальности использованных при составлении источников. Для получения качественного результата подаваемый на вход алгоритму текст должен содержать термины, релевантные рассматриваемой предметной области, иначе в результате онтология не будет в полной мере представлять текст.

Предлагаемый алгоритм решает задачу построения дерева классов, а также выделяет экземпляры классов, но это не является полным решением задачи автоматического формирования онтологии. Классы в онтологии вместе с экземплярами являются основой для последующего извлечения связей, атрибутов и их значений. В данной статье рассматривается только начальный этап автоматического формирования онтологии из текстов по определенной предметной области.

Таким образом, целью данной статьи является описание разработанного алгоритма формирования дерева классов и их экземпляров и некоторые особенности его технической реализации.

1. Онтология, классы и их экземпляры

1.1. Понятие и модель онтологии

Большой спектр решаемых с использованием онтологии задач породил множество определений понятия «онтология», большинство которых рассматриваются в [Клещев и др., 2000].

В данной статье онтология понимается как формальное представление предметной области, представимое, согласно [Гаврилова и др., 2000] в виде тройки:

$$O = \langle T, R, F \rangle, \quad (1)$$

где T – конечное множество терминов рассматриваемой предметной области; R – конечное множество отношений между понятиями рассматриваемой предметной области; F – конечное множество функции аксиоматизации, заданных на множестве T и R . При этом присутствует обязательное требование непустоты множества T .

1.2. Классы и их экземпляры

Классы (или понятия) онтологии вместе с их экземплярами (или индивидами) являются основой онтологии любого типа, применяются и строятся одинаково практически в любом языке описания онтологий.

Классы онтологии представляют собой коллекции объектов, сформированные по определенному принципу. Классы могут включать в себя как другие классы, так и экземпляры. Возможно содержание одним классов как классов-потомков, так и собственных экземпляров. Таким образом, классы представляют собой иерархию понятий по отношению вложения.

Экземпляры классов в онтологии являются компонентами нижнего уровня. Выражают как физические, так и абстрактные объекты. Формально онтология может не включать в себя ни одного экземпляра класса.

Кроме классов и их экземпляров онтология содержит отношения и атрибуты. Извлечению этих компонентов из текста предшествует формирование дерева классов и их экземпляров.

2. Формирование дерева классов и экземпляров

Разработанный алгоритм автоматического формирования онтологии включает в себя этапы первичной обработки текста, извлечения терминов, извлечения классов и извлечения экземпляров.

2.1. Первичная обработка текста

На вход реализованной информационной системе подается текст на естественном языке. Для определения части речи слова в тексте и правильного составления термина по

лингвистическому шаблону используются инструменты Mystem [Андреев и др., 2013] и LingvoNET [Торгашов, 2015].

Mystem — морфологический анализатор русского языка с поддержкой снятия морфологической неоднозначности. Программа работает на основе словаря и способна формировать морфологические гипотезы о незнакомых словах.

LingvoNET - библиотека для склонения и спряжения слов русского языка для .NET Framework. Библиотека содержит встроенные словари глаголов, существительных, прилагательных и местоимений русского языка на основе словаря Зализняка.

Текст, который был размечен программой Mystem, обрабатывается информационной системой с помощью библиотеки LingoNET и заносится в базу данных.

2.2. Извлечение терминов

Для построения понятийного аппарата из текстов предметной области используется поиск и выделение субстантивных именных словосочетаний, выражаемых схемой: согласуемое слово + существительное. В этой модели существительное является главным словом, а согласуемое слово — зависимым и может выражаться как прилагательным, так и существительным. Словосочетания могут включать в свой состав также предлоги и сочинительные союзы.

Количество слов в именных словосочетаниях колеблется от двух до семи и в среднем составляет три слова. В ходе работы были выделены шаблоны именных словосочетаний, используемых для выделения терминов предметной области.

Для возможности выделения терминов из текстов предметной области были разработаны лингвистические шаблоны, с помощью которых удастся выделить основные термины. В русском языке синтаксическая структура терминов предметной области более чем в 90 процентов случаев соответствует следующим пяти шаблонам:

- одиночные существительные;
- существительное + существительное в родительном падеже;
- прилагательное + существительное;
- прилагательное + прилагательное + существительное;
- существительное + прилагательное + существительное в родительном падеже.

Вместе с тем существуют сложные словосочетания, используемые для обозначения понятий и терминов, состоящих из трех и более значимых слов.

Выражение понятий и терминов словосочетаниями в пять и более слов, с использованием союзов и предлогов встречается

редко, особенно такими словосочетаниями, в которых части речи не чередуются (например, прилагательное + прилагательное + прилагательное + существительное + существительное в родительном падеже).

2.3. Автоматическое формирование дерева классов из текстов по предметной области

На начальном этапе построения онтологии необходимо построить дерево классов. Для разработки инструмента, осуществляющего формирование дерева классов, был разработан алгоритм автоматического построения дерева классов на основе данных из тезауруса, а также применялась методика слабоиерархической онтологии [Загоруйко и др., 2006]. Данные алгоритмы представляют собой последовательность шагов по обработке характеристик, их структурированию, а также выделению классов на их основе.

Рассмотрим данные алгоритмы более детально. На первом этапе анализа текстов выделяются термины по лингвистическим шаблонам рассмотренных в параграфе 2.2. После выделения списка терминов они проходят отбор по стоп-словам. Стоп-слова были выделены экспертом лингвистом и в основном включают в себя: местоимения, междометия, частицы, вводные слова и т.п.

На втором этапе происходит непосредственно построение дерева классов. Формирование классов происходит при сопоставлении терминов, выделенных из текста, с тезаурусом по предметной области предварительно составленный экспертом-лингвистом, который представляет следующие данные:

- Лемма термина;
- Отношение термина с другими терминами в тезаурусе (в качестве основного отношения рассматривается гипероним-гипоним).

Сам процесс построения тезауруса состоит из нескольких взаимосвязанных этапов:

Первый этап - формирование словаря. Словарь – первоначальные множества ключевых слов. При этом рассматривается представительный массив наиболее информативных для данной предметной области документов.

Выбираются слова, употребляемые в этих источниках, при всем этом устанавливается частота употребления слов, и учитываются все формы, которые могут иметь слова.

Второй этап – формирование множества ключевых слов. Из словаря формируется множество ключевых слов. При отборе ключевых слов учитывается информативность слова, которая определяется исходя из частоты встречаемости слова, роли слова в этой предметной области. Процесс выбора ключевых слов достаточно сложно формализовать.

Например, такой критерий, как частота встречаемости не может быть абсолютным. Если слово встречается в текстах очень часто, это может означать, что оно выражает чрезмерно широкое понятие, либо недостаточно четко определено, т.е. неинформативно.

Если ключевое слово встречается очень редко, это может означать, что оно выражает новое понятие и таким образом является информативным.

Третий этап – формирование классов эквивалентности. Выделение дескрипторов. Автоматические информационно-поисковые тезаурусы являются составным элементом автоматического индексирования документов и запросов.

В словарной статье автоматического тезауруса, как правило, зафиксированы отношения условной эквивалентности, отношения подчинения и ассоциативные отношения.

Данные по тезаурусу хранятся в базе данных и представлены в следующем виде:

Id	Lemma	Topic
1	механика	Физика
2	сила	Физика
3	сложение сил	Физика
4	параллелограмм сил	Физика

Рисунок 1 – Тезаурус

Сопоставление терминов происходит по полному совпадению данных с леммой термина из тезауруса. При полном совпадении выделенный термин применяет на себя свойства тезауруса.

Данные по свойствам тезауруса хранятся в таблице данных ThesaurusProperty и представлены в следующем виде:

Id	Property	Subject	Lemmalid
10	hyperonym	сила	7
11	hyperonym	сила	8
12	hyperonym	сила	9
13	hyperonym	сила	10

Рисунок 2 – Свойства тезауруса

Данные по выделенным терминам хранятся в таблице данных FilteredWord и представлены в следующем виде:

Id	Lemm	Template
20	согласование характеристики дизеля	SSS
21	история создания тепловоза	SSS
22	история создания передачи	SSS
23	сгорание жидкого топлива	SAS

Рисунок 3 – Выделенные термины

На следующем этапе происходит выделение классов по следующему алгоритму:

1. У выделенной Lemma из таблицы ThesaurusLemma рассматривается свойство hyperonym из таблицы ThesaurusProperty;

2. Subject из строки, которая имеет свойство hyperonym у выделенной Lemma, будет являться классом, а Lemma подклассом;

3. Ищем выделенный Subject из таблицы ThesaurusProperty в столбце Lemma из таблицы ThesaurusLemma. Если Subject и Lemma полностью совпадают, то мы возвращаемся к шагу 1 и начинаем выделять надклассы у Subject.

4. Если у выделенного Subject с Lemma совпадений не найдено. То на этом шаге выделение классов останавливается.

Например, у нас есть следующие данные в таблице ThesaurusLemma:

Таблица 1 – Пример данных в базе данных ThesaurusLemma

Id	Lemma	Topic
1	блок цилиндров	машины
2	бензиновый двигатель	машины

Они соотносятся со следующими данными в таблице 2.

1. Был выделен термин “блок цилиндров”;
 2. В таком случае “бензиновый двигатель” из строки, которая имеет свойство hyperonym у выделенной Lemma, будет являться классом, а “блок цилиндров” подклассом;

3. Ищем выделенный класс “бензиновый двигатель” из таблицы ThesaurusProperty в столбце Lemma из таблицы ThesaurusLemma на полное совпадение. При анализе было выделено совпадение. “Бензиновый двигатель” имеет Id равное 2 в таблице ThesaurusLemma. Повторяем шаги 1, 2 и 3 еще раз:

1. Был выделен термин “бензиновый двигатель”;

2. В таком случае “двигатель” из строки, которая имеет свойство hyperonym у выделенной Lemma, будет являться классом, а “бензиновый двигатель” подклассом;

3. Ищем выделенный класс “двигатель” из таблицы ThesaurusProperty в столбце Lemma из таблицы ThesaurusLemma на полное совпадение. При анализе совпадений не найдено;

4. Выделение классов остановлено. Получена следующая структура:

Класс: Двигатель

Подкласс1: Бензиновый двигатель

Подкласс2: Блок цилиндров

Таблица 2 – Пример данных в базе данных ThesaurusProperty

Id	Property	Subject	LemmaId
1	hyperonym	бензиновый двигатель	2
2	hyperonym	двигатель	3

На третьем этапе рассматриваются термины, которые отсутствуют в тезаурусе. К ним применяется методика слабоиерархической онтологии. Выделенные термины объединяются в слабоиерархическую онтологию.

Основанием для объединения типа родовидового было вхождение термина из среднечастотной зоны в словосочетание в качестве главного компонента, например: воздействие => антропологическое воздействие; изотоп => тяжелый изотоп. Термины, которые входят в выделенные словосочетания в качестве зависимых, позволяют объединять словосочетания в ассоциативные группы, например: почва <= эрозия почвы / плодородие почвы / дефляция почвы.

2.4. Автоматическое формирование экземпляров

Выделение экземпляров можно разделить на два этапа. На первом этапе происходит определение последнего элемента в цепочке дерева классов по следующему правилу: если Lemma из таблицы ThesaurusLemma не имеет совпадений в столбце Subject в таблице ThesaurusProperty, то термин из столбца Lemma является последним в цепочке классов.

На втором этапе происходит непосредственно определение экземпляров. Для возможности выделения экземпляров из текста были разработаны следующие лингвистические шаблоны, с помощью которых удастся выделить экземпляры: существительное + цифра, существительное + буква + цифра, существительное + буква, существительное + существительное + буква, существительное + существительное + цифра.

На начальном этапе термины по выше выделенным лингвистическим шаблонам из таблицы FilteredWord сопоставляются с последним элементом в цепочке классов. Сопоставление происходит по полному совпадению существительных.

Например, есть два правила существительное + существительное + буква и прилагательное + существительное, если существительное из правила прилагательное + существительное совпадает с существительным из правила существительное + существительное + буква, то термин существительное + существительное + буква является экземпляром в цепочке классов прилагательное + существительное.

3. Техническая реализация проекта

Для данного проекта был выбран язык C# (Microsoft Visual Studio 2015), на платформе .NET 4.5.2 и СУБД Microsoft SQL Server 2012.

При разработке инструмента была использована технология Entity Framework. В Entity Framework модель данных реляционной базы данных сопоставляется объектной модели, выраженной в языке программирования разработчика.

При запуске приложения Entity Framework преобразует запросы LINQ из объектной модели в SQL и отправляет их в базу данных для выполнения. Когда база данных возвращает результаты, Entity

Framework преобразует их обратно в объекты, с которыми можно работать на собственном языке программирования.

Основное предназначение разработанной базы данных заключается в хранении сущностей выбранной пользователем онтологии и предоставлении данных для методов, осуществляющих работу алгоритма.

После написания программы, система должна выполнять следующие функции с БД:

- Отображать содержимое базы данных;
- Выводить информацию из базы данных на экран;
- Удалять информацию из базы данных;
- Предоставлять данные для методов, осуществляющих подсчет критериев.

Таким образом, программа использует БД, которая хранится в двух файлах: с расширением mdf и ldf, которые используются запускаемым в операционной среде экземпляром СУБД.

Взаимодействие с системой осуществляется через web-интерфейс, расположенный по адресу <http://auto-ontology.ru>. Доступ пользователя к сайту осуществляется при помощи любого современного браузера.

На стороне сервера работают службы ИС, обеспечивающие выполнение скомпилированных исходных файлов системы. Для автоматической первичной обработки текста на сервере должен содержаться исполняемый файл Mystem, который выполняется при соответствующем запросе пользователя. Кроме того, на сервере должна быть размещена СУБД MS SQL Server 2012.

Данная конфигурация системы имеет 2 основных преимущества: независимость от платформы и облачная обработка данных. Таким образом, система доступна почти с любого современного устройства и не использует его ресурсов для собственной работы.

Из недостатков следует отметить зависимость от подключения к интернету.

4. Оценка качества построенной онтологии

Проблема оценки качества онтологии является одной из актуальных проблем современного онтологического инжиниринга. Распространенным критерием оценки качества онтологии является оценка работы приложения, использующего онтологию.

В настоящее время за методикой оценки качества онтологии можно рассматривать системы семантического поиска, которые используют онтологии.

Применяя онтологию, поисковые системы могут улучшить качество поиска за счет динамического

расширения запросов пользователя. Для таких систем необходимо обрабатывать большое количество текстовой информации и переработка данных текстов в онтологию при помощи экспертов. Но экспертная оценка трудозатратна.

Построение качественной онтологии без помощи эксперта усложняется отсутствием некоторого эталона онтологии. В связи с этим, для проверки качества построенной онтологии имеет смысл проводить начальную оценку работы алгоритма, только в одной тематике (например, машиностроение).

Именно по этой теме был построен тезаурус терминов, что позволит оценить работу метода на первых этапах.

Заключение

В статье рассматривается подход к автоматизации процесса построения онтологии по текстам предметной области, на основании лингвистических шаблонов, тезаурусов и методики слабоиерархической онтологии.

В результате работы был создан инструмент автоматизированного создания онтологии предметной области. Полученная в результате онтология представляет собой структурированную систему базы знаний на основе текста предметной области.

Использование данного инструмента позволит значительно сократить затраты времени на составление и редактирование онтологии и будет полезен во многих областях знаний, таких как компьютерная и корпусная лингвистика, лексикография, библиотечное дело, семантический поиск и многие другие.

Разработанный инструмент находится в свободном доступе и расположен по адресу: <http://auto-ontology.ru>

Библиографический список

- [Можжерина, 2011] Можжерина Е. С. Автоматическое построение онтологии по коллекции текстовых документов // Электронные библиотеки: Перспективные Методы и Технологии, Электронные коллекции-RCDL. – 2011. – С. 293-298.
- [Соловьев и др., 2006] Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В. Онтологии и тезаурусы. Учебное пособие. – Казань, Москва. – 2006.
- [Андреев и др., 2014] Андреев И.А., Башаев В.А., Клейн В.В., Мошкин В.С., Ярушкина Н.Г. Семантическая метрика терминологичности на основе онтологии предметной области // Автоматизация процессов управления. – 2014. – № 4 (38). – С. 76 – 84.
- [Гаврилова и др., 2000] Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем: учеб. для вузов. — СПб.: Питер, 2000. — 384 с.
- [Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В.Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.
- [Клещев и др., 2000] Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия «онтология».

// Научно-техническая информация, серия 2 «информационные процессы и системы». — М.: ВИНТИ, 2001. — №2. — С. 20–27.

[Андреев и др., 2013] Андреев И. А., Башаев В. А., Клейн В. В. Разработка программного средства для извлечения терминологии из текста на основании морфологических признаков, определяемых программой Mystem //Интегрированные модели и мягкие вычисления в искусственном интеллекте». — М.: Физматлит. — 2013. — С. 1227-1236.

[Торгашов, 2015] NuGet Gallery | LingvoNET - Библиотека для склонения и спряжения слов русского языка 1.1.0 [Электронный ресурс] // NuGet Gallery [Интернет-портал]. URL: <https://www.nuget.org/packages/LingvoNET/> (Дата обращения: 29.11.2015)

[Загорюлько и др., 2006] Загорюлько Ю. А. и др. Подход к построению предметной онтологии для портала знаний по компьютерной лингвистике //Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог». — 2006. — С. 148-151.

[Ярушкина и др., 2014] Ярушкина Н.Г., Мошкин В.С. Применение онтологического подхода к анализу состояния локальной вычислительной сети. Радиотехника. — 2014. — № 7. — С. 120-124.

AUTOMATIC CREATION OF CLASS HIERARCHY AND SEARCH INSTANCES FROM TEXT FOR ONTOLOGY

Andreev I.A., Beksaeva E.A., Zheleпов A.S,
Klein V.V., Serkov I.P.

*Ulyanovsk State Technical University, Ulyanovsk,
Russia*

ares-ilya@ya.ru

ekaty@clukva.ru

a.zheleпов@yandex.ru

vikklein93@gmail.com

bivo@bivo.ru

The article deals with the algorithm of automatic creation of class hierarchy and search instances from text of subject domain for ontology.

Introduction

Over the past decade, ontologies and knowledge bases have gained popularity due to their high potential benefits in a number of applications including data/knowledge organization and search applications. Though ontology integration is beneficial, it is very well known that ontology creation is an expensive process.

The modeling of non-trivial domain ontologies is difficult, and is time and resource intensive. The knowledge acquisition bottleneck problems in ontology creation and maintenance have resulted in expensive procedures for maintaining and expanding the ontology library available to support the growing and evolving needs of analysts in various domains.

Ontology generation is the automatic creation of ontologies, including extracting the corresponding domain's terms and the relationships between those concepts from a corpus of natural language text. As building ontologies manually is extremely labor-

intensive and time consuming, there is great motivation to automate the process.

Main Part

This article provides the algorithm of automatic creation of ontology from text. The proposed algorithm is put mainly linguistic methods. The algorithm is based mainly linguistic processors such as phrase chunking.

We described a method to automatically extract key terms, and relationships between the key terms.

For automatic building class hierarchy and searching instances used specialized thesaurus and concept hierarchy derivation. Specialized thesaurus are made expert and submit the following information: lemma of term and the relationship between terms in the thesaurus.

In the concept hierarchy derivation step, the system tries to arrange the extracted concepts in a taxonomic structure. This is mostly achieved by unsupervised hierarchical clustering methods.

Conclusion

Knowledge intensive applications require extensive domain-specific knowledge in addition to general-purpose knowledge bases.

However, domain-specific ontology creation and maintenance is an expensive process and hence is referred to as the knowledge acquisition bottleneck. Thus, you need an expert to be involved either in the process of creating the ontology itself and to evaluate the effectiveness of the developed algorithm.

In this paper, we presented a generalized and improved procedure to automatically extract information from text resources and rapidly create ontologies while keeping the manual intervention to a minimum.

We also defined evaluation metrics to assess the quality of the ontologies created using our methodology. The results show that a decent amount of knowledge can be accurately extracted while keeping the manual intervention in the process to a minimum.

Another conclusion we came into is that you need to use the developed algorithm in several real tasks before evaluating its effectiveness.

Using the developed algorithms, a tool that uses C# and Microsoft SQL Server 2012 has been created for automatic creation of ontology from text. This tool is available at auto-ontology.ru. Using this tool you can make an ontology of your own, based on a text you choose.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК004.8

ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА ЗНАНИЙ ХРАНЯЩИХСЯ В МНОГОМЕРНЫХ ДАННЫХ ПОСРЕДСТВОМ РАСШИРЕННОГО БАЗОВОГО СЕМАНТИЧЕСКОГО ГИПЕРГРАФА

Шарипбай А.А., Барлыбаев А.Б., Сабыров Т.С.

*Евразийский Национальный Университет им. Л.Н.Гумилева,
г. Астана, Республика Казахстан*

Sabyrov.Talgat@mail.ru

В данной работе рассматриваются вопросы представления и обработки знаний. Главной новизной работы является вложенность знаний, представление знаний посредством расширенного базового семантического гиперграфа (РБСГ). Построена математическая модель представления знаний. Описаны технология хранения, обработки знаний. Представлена клиентская часть представления знаний.

Ключевые слова: глобалы; многомерные массивы; гиперграфы; семантическая сеть; дейкстра.

Введение

Одним из самых важных и сложных вопросов искусственного интеллекта является вопрос представления и обработки знаний. Основные модели представления знаний являются: продукционные, фреймовые, логические и семантические сети. Используя любую из этих моделей можно разработать интеллектуальную систему. Поэтому перед началом разработки стоит вопрос об выборе необходимой модели [Paolo Giudici, David Heckerman, Joe Whittaker].

Первую продукционную модель предложил Пост в 1943 году. Основанна на правилах, позволяет представить знание в виде предложений типа «Если (условие), то (действие)». Продукционная модель обладает тем недостатком, что при накоплении достаточно большого числа (порядка нескольких сотен) продукций они начинают противоречить друг другу [Ishida, Toru.].

Фрейм – способ представления знаний в искусственном интеллекте, представляющий собой схему действий в реальной ситуации. Первоначально термин «фрейм» ввёл Марвин Минский в 70-е годы XX века для обозначения структуры знаний для восприятия пространственных сцен. Является моделью абстрактного образа, минимально возможное описание сущности какого-либо объекта, явления, события, ситуации, процесса [Matthias Strobbе, Olivier VanLaere, Bart Dhoedt, Filip DeTurck, Piet Demeester].

Основная идея подхода при построении логических моделей представления знаний – вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода. В основе логических моделей представления знаний лежит понятие формальной теории [CHRISTINE W.CHAN].

Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа. В семантической сети роль вершин выполняют понятия базы знаний, а дуги (причем направленные) задают отношения между ними. Таким образом, семантическая сеть отражает семантику предметной области в виде понятий и отношений [Isaac Lera, Carlos Juiz, Ramon Puigjaner].

При обработке знаний обычно используются методы поиска решений на основе исчисления предикатов (правила модус-понус и т.д., конъюнкции дизъюнкции и отрицания и т.п.) [Yuksel Uckan]. Также используются прямой и обратный вывод в экспертных системах продукционного типа (стратегия поиска в глубину, стратегия поиска в ширину, разбиение на подзадачи, a-b-алгоритм и т.п.) [Birgit Vogel-Heuser, Alexander Fay, Ina Schaefer, Matthias Tichy]. Все чаще стали использоваться метод обработки знаний в интеллектуальных системах с фреймовым представлением (демоны, присоединенные

процедуры, механизм наследования) [JahangirKarimi, M.K.Zand].

Очень интересным подходом было бы использование многомерных массивов как хранилища для базы знаний. Многомерные массивы это массивы, элементами которых являются массивы. Определение многомерного массива должно содержать информацию о типе, размерности и количестве элементов каждой размерности. Элементы многомерного массива располагаются в памяти в порядке возрастания самого правого индекса.

Переменные как локальные, так и глобальные могут существовать в виде простых или индексированных структур. Глобальные переменные или глобалы, являясь хранимыми данными, создают основу так называемого прямого доступа [Иванчева Н.А., Иванчева Т.А.]. Глобалы это структуры данных, как правило, многомерные, которые хранятся в базе данных и могут обрабатываться в многопользовательской среде различными процессами. Многомерность данных реализована через индексы, поэтому говорят об индексированных переменных.

Глобальные данные сохраняются в В*-деревьях. Дерево, которое имеет одинаковое число подуровней в каждом своем поддереве, называется сбалансированным (balancedtree, отсюда и В-дерево). Дерево, у которого каждый ключ указывает на блок данных, содержащий требуемую запись, называется В*-деревом. Оно делает возможной интеграцию области указателей и области данных. В*-деревья можно рассматривать как сеть состоящая из графов.

Согласно [ЗЫКОВА.А.] гиперграф, $H(V, E)$ есть пара, где V – множество вершин $V = \{v_i, i \in I = \{1, 2, \dots, n\}\}$, а E – множество ребер $E = \{e_j, j \in J = \{1, 2, \dots, m\}\}$; каждое ребро представляет собой подмножество V . Вершина v и ребро e называются инцидентными, если $v \in e$. Для $v \in V$ через $d(v)$ обозначается число ребер, инцидентных вершине v ; $d(v)$ называется степенью вершины v . Степень ребра e – число вершин, инцидентных этому ребру, обозначается через $r(e)$. Гиперграф H является r -однородным, если все его ребра имеют одинаковую степень r .

Онтология – это комплексный и детально формализованное представление о определенном домене с использованием концептуальных фреймворков состоящего из экземпляров, концептов (классов), атрибутов (свойств), функций (операции), аксиом (фактов) и связей.

Для построения онтологической модели будет использоваться расширенный базовый семантический гиперграф (РБСГ). Узлы в этих графах представляют семантические признаки (свойства и функции) объектов или субъектов, а дуги представляют отношения между ними.

Отметим, что структура РБСГ аналогична парадигме объектно-ориентированного программирования. Поэтому РБСГ могут быть использованы для описания прикладного программного обеспечения, которое может ответить на вопросы пользователей о базе знаний. Концепты в структуре гиперграфа описаны на деревьях, которые преобразуются в математические формулы.

Есть несколько статей, где описывается семантический граф [Lian R., Goertzel B., Ke S., O'Neill J., Sadeghi K., Shiu S., Wang D., Watkins O., Yu G.] описывает модель семантического гиперграфа как «гипер-граф основанный на семантической сети», который может представить более сложные семантические сети и более эффективные структуры данных для хранения знания в репозиториях.

Весы вершин БСГ: $K = \{k_a\}, a \in A = \{1, 2, \dots, b\}$, где $k_a = \{S_a, V_a, E_a\}$, $S_a = \{s_{j=1, 2, \dots}^a\}$ – множество свойств класса, $V_a = \{v_{j=1, 2, \dots}^a\}$ – множество экземпляров класса, $E_a = \{e_{j=1, 2, \dots}^a\}$ – множество семантических дуг, которые инцидентны классу, k_s – количество свойств класса, k_v – количество экземпляров класса, k_e – количество семантических дуг, которые инцидентны классу. Вершина-экземпляр класса может быть представлена в виде тройки $v_i = \{k_i, S_i, E_i\}$, где k_i – родительский класс, т.е. $v_i \in k_i$, где $S_i = \{s_{j=1, 2, \dots}^i\}$ – множество экземпляров, $E_i = \{e_{j=1, 2, \dots}^i\}$ – множество семантических дуг, которые инцидентны экземпляру, k_s – количество свойств экземпляра класса, k_e – количество семантических дуг, которые инцидентны экземпляру классу.

Эту тройку, $k_a = \{S_a, V_a, E_a\}$, мы поменяли на пятерку $k_a = \{S_a, F_a, I_a, V_a, E_a\}$, где F_a – множество функции класса, I_a – множество инкапсуляции в классе. Вершина-экземпляр класса может быть представлена в виде пятерки $v_i = \{k_i, S_i, F_i, I_i, E_i\}$, где F_i – множество функций экземпляров, I_i – множество инкапсуляций экземпляра.

Расширение семантического гиперграфа имеет тот же смысл, что и расширение контекстно-свободной грамматики до атрибутивной грамматики. То есть если $G = (N, T, P, S)$ это контекстно-свободная грамматика, то атрибутивная грамматика, определяется как $AG = (N, T, P, S, AS, AI, R)$, где N – множество нетерминалов, T – множество терминалов (непересекающихся с N), P – множество правил, S – начальный нетерминал, AS – конечное множество синтезированных атрибутов, AI – конечное множество наследственных атрибутов (непересекающихся с AS), R – конечное множество семантических правил.

Здесь атрибутивная грамматика совершенно новый математический инструмент, который позволяет описывать не только структуру языковых единиц, но их атрибуты (семантические свойства). Это хорошо известный классический научный результат Дональда Эрвина Кнута. Вот как следует понимать расширение семантического гиперграфа, мы

добавили два новых символа, которые позволяют описать множество функций класса и набор классов инкапсуляции. Но мы не имеем имя для расширенного семантического гиперграфа.

Знания записанные, в глобале, мы будем структурировать и обрабатывать как онтологию, которая основана на РБСГ.

1. Представление знаний

1.1. Описание использованных технологий и структур данных

Глобалы это структуры данных в ООСУБД IntersystemsCache, как правило, многомерные, которые хранятся в базе данных и могут обрабатываться в многопользовательской среде различными процессами [WolfgangKirsten, MichaelIhringer, MathiasKühn, BernhardRöhrig.]. Для оценки всех значений глобалов, разработаем каталог электронных устройств. Для этого введем глобал ^device, который на первом уровне имеет артикул ^device (article), а на втором уровне глобал имеет уровень свойств устройств ^device (article,property), уровень свойства устройств может содержать n-количество подуровней свойств устройств ^device (article,property,subproperty,...,n). Также на третьем уровне артикула могут быть функции ^device(article,*func*). Этот уровень определяется ключевой строкой *func*. Все подуровни *func* содержат в себе функции объекта (в данном случае устройства) ^device(article,*func*,function). Аналогично уровню свойства устройств, уровень функций может содержать n-количество подуровней функций устройств ^device(article,*func*,function,subfunction,...,n). Тем самым мы использовали многомерность данных предоставленную глобалами.

Таблица 1 – Примеры записей в глобале device

Узел	Значение
^device(111587)	""
^device(111587,"name")	"ASUS GL752VW"
^device(111587,"type")	"pc"
^device(111587,"type","subtype")	"laptop"
^device(111587,"processor")	"Intel Core i7"
^device(111587,"ram")	"12gb"
^device(111587,"os")	"Windows 10"
^device(111587,"video")	"GeForce GTX960M"
^device(111587,"*func*","run")	w 'Hello'"
^device(111587,"*func*","run1")	w 'Hello 1'"

^device(100997)	""
^device(100997,"display")	"sensory,IPS"
^device(100997,"name")	"3Q Meta RC7802F"
^device(100997,"os")	"Android"
^device(100997,"price")	37000
^device(100997,"ram")	"1gb"
^device(100997,"type")	"mobile"
^device(100997,"type","subtype")	"tablets"
^device("rel",111587,10097)	"SameCompany/MadeOfIron"

После того как мы указали свойства и функции устройств, нам необходимо связать эти устройства по артикулам. Для этого используется ключевая строка "rel" ^device(rel, article1, article2). В уровнях этой строки содержатся артикулы связанных устройств, а в значениях этих уровней будут содержаться названия связей, разделенные символом "/".

^device("rel",111587,111588)="SameCompany/MadeOfIron"

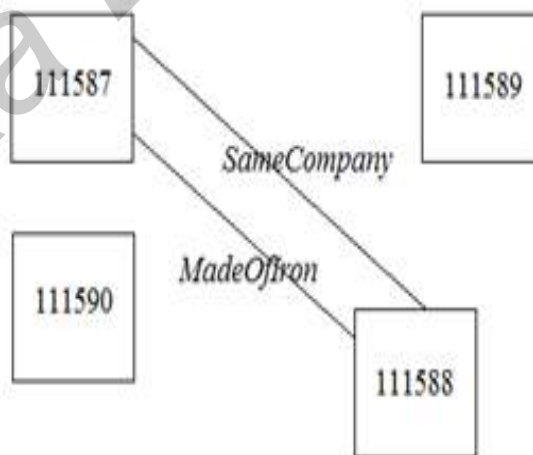


Рисунок 1 – Связи между устройствами

Функции представляются в глобале в виде записи:

^device(111587,"*func*","run") = "w 'Hello'"

Значения функций описаны ниже в таблице 2.

Команда	Действие	Сокращение
set	присвоить значение	s
if ...	выполнить все команды справа по строке - если ...	i ...
do	выполнить подпрограмму	d

for ...	выполнять все команды справа по строке многократно (цикл)	f ...
quit	остановитьциклиипрограмму	q
kill	уничтожитьпеременнуюилиузел	k
new	объявить новые переменные с ограниченной областью	n
execute	выполнит строку текста как отдельную программу	x
merge	объединениеузлов и глобалей	m
\$data	0 или 1 или 10 или 11	\$d
\$get	замену, если не существует такойпеременной	\$g
\$select	одинизвариантов	\$s
\$length	(длинуиличисло_входов)+1	\$l
\$piece	возвращаетчастьстрок и	\$p
\$extract	возвращаетчастьстрок и	\$e
\$find	(номерсимвола в строке)+1	\$f
\$translate	перекодированнуюстроку	\$tr
\$justify	округляетчисла	\$j
\$order	индекс узла дерева справа или слева на этом уровне	\$o
\$query	полный адрес ветви дерева справа/слева от текущей	\$q
\$qs	частьадресаветви (индекс)	\$qs
\$ql	длину полного адреса ветви дерева (размерность адреса)	\$ql

Таким образом, мы создали глобалы, в которых хранятся знания об устройствах. Эти глобалы мы будем собирать в структурированные фреймы в семантической сети в нашей базе знаний об

устройствах. На рисунке 2 мы соберем наши глобалы в сеть.

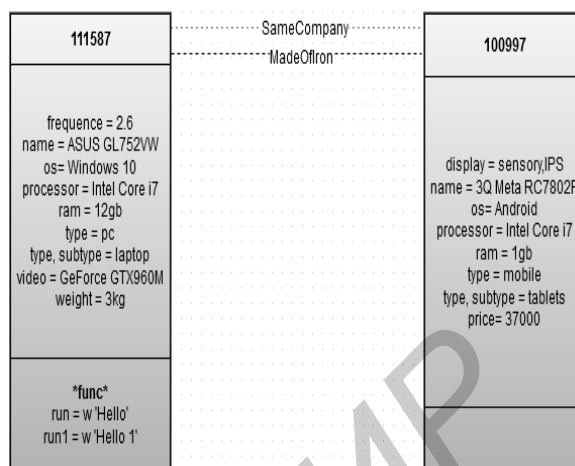


Рисунок 2 – Представление знаний об устройствах в структурной форме.

1.2. Описание гиперграфа и математической модели данных

Гиперграф является парой $H(V, E)$, но так как мы в знаниях используем помимо свойств и отношений функций, а также имеет свою внутреннюю вложенность данный вид гиперграфа нам не подходит. Поэтому мы предлагаем расширить гиперграф за счет добавления функций до расширенного базового семантического гиперграфа (РБСГ) [AltynbekSharipbayevand AlibekBarlybayev].

Будем использовать $H(A, P, F, R)$, где H – РБСГ, A – Id концепта, P – свойства, F – функции, R – отношения, семантические дуги. $A = \{A_1, A_2, \dots, A_n\}$, $P = \{P_1, P_2, \dots, P_n\}$, $F = \{F_1, F_2, \dots, F_n\}$, $R = \{R_1, R_2, \dots, R_n\}$.

$$P_{nm} = P_1^{1,2,\dots,m} + P_2^{1,2,\dots,m} + \dots + P_n^{1,2,\dots,m} \quad (1)$$

$$F_{nm} = F_1^{1,2,\dots,m} + F_2^{1,2,\dots,m} + \dots + F_n^{1,2,\dots,m} \quad (2)$$

Для поиска в графах мы использовали алгоритм Дейкстры.

A – множество вершин графа

R - множество рёбер графа

w - вес (длина) всех ребер. В данном случае константа 1

a - вершина, расстояния от которой ищутся

U - множество посещённых вершин

$d[u]$ - по окончании работы алгоритма равно длине кратчайшего пути из a до вершины u

$l[u]$ - по окончании работы алгоритма содержит кратчайший путь из a в u

Присвоим $d[a] \leftarrow 0, p[a] \leftarrow 0$

Для всех $u \in A$ отличных от a

присвоим $d[u] \leftarrow \infty$

Пока $\exists v \notin U$

Пусть $v \notin U$ – вершина с минимальным $d[u]$

занесем v в U

Для всех $u \notin U$ таких, что $vu \in R$

если $d[u] > d[v] + w$ то

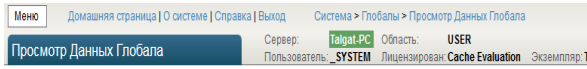
изменим $d[u] \leftarrow d[v] + w$

изменим $l[u] \leftarrow l[v], u$

2. Программная реализация

2.1. Обработка данных

Для программной реализаций мы записали тестовые данные устройств в глобал ^device. Отображение можно увидеть в портале управления системой



Просмотр глобала в области USER:

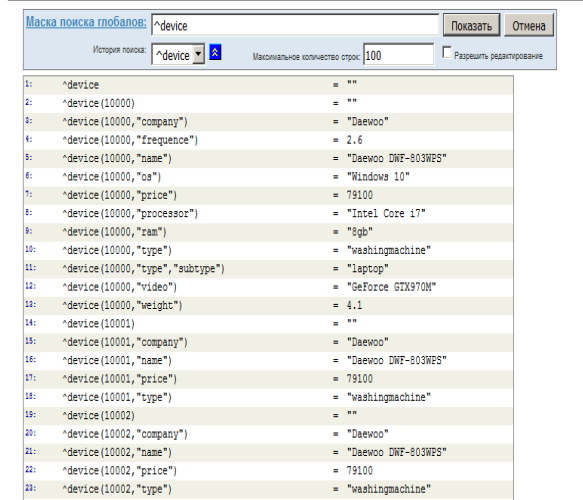


Рисунок 3 – Отображение глобалов

С помощью языка программирования Cache ObjectScript (COS) мы собрали данные глобала во фреймы и перевели их в данные формата JavaScriptObjectNotation (JSON) для вывода данных в клиентскую часть [JSON].

```

{
  "^device(58783)": {
    "^device(58783,\"name\")": "Data switch KVM D-Link DKVM-4K",
    "^device(58783,\"price\")": "10460",
    "^device(58783,\"type\")": "network"
  },
  "^device(75009)": {
    "^device(75009,\"interface\")": "usb",
    "^device(75009,\"name\")": "Defender Accent 930, Black, USB",
    "^device(75009,\"price\")": "2059",
    "^device(75009,\"type\")": "keyboard"
  },
  "^device(77115)": {
    "^device(77115,\"frequency\")": "20-16000h",
    "^device(77115,\"length\")": "1.37m",
    "^device(77115,\"name\")": "ACME MK-200",
    "^device(77115,\"price\")": "803",
    "^device(77115,\"type\")": "earphone"
  }
}

```

Рисунок 4 – Фрагмент данных JSON

Данные в виде фреймов были выведены в браузере. Для этого мы использовали язык программирования JavaScript с фреймворком AngularJS (рисунок 5)



Рисунок 5 – Отображения данных в виде фреймов и их связи

2.2. Реализация поиска

Как мы упомянули ранее, мы использовали поиск методом Дейкстры. Блок схема алгоритма работы нашей программы показан на рисунке 6.

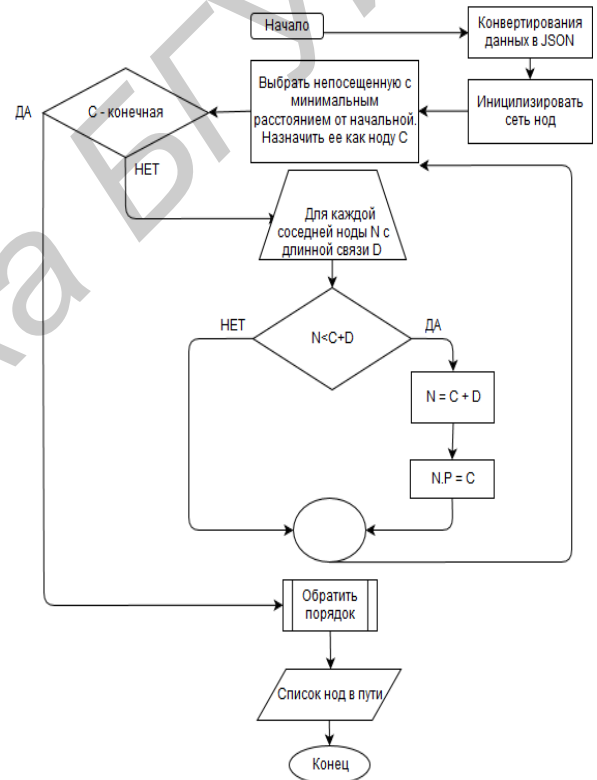


Рисунок 6 – Блок схема алгоритма поиска

В программе как входная данная пишется название вершины. Программа находит кратчайшие пути на каждые связанные вершины по вышеуказанному алгоритму. Результат показан на рисунке 7

```

^a(10000) : ^a(10000)=0
^a(10000) : ^a(10001)=1
^a(10000) : ^a(10002)=2
^a(10000) : ^a(10003)=3
^a(10000) : ^a(10004)=4
^a(10000) : ^a(10005)=5
^a(10000) : ^a(10006)=6
^a(10000) : ^a(10007)=7
^a(10000) : ^a(10008)=8
^a(10000) : ^a(10009)=9
нет пути до вершины ^a(77150)
^a(10000) : ^a(10010)=10

```

Рисунок 7 – Результат работы программы

Заключение

Описан метод представления и обработки знаний посредством РБСГ. Дальнейшая работа будет посвящена nр, новому алгоритму обработки вложенных знаний, будут проведены эксперименты над производительностью обработки знаний.

Библиографический список

- [Paolo Giudici, David Heckerman, Joe Whittaker.] Statistical Models for Data Mining. Data Mining and Knowledge Discovery. July 2001, Volume 5, Issue 3, pp 163-165.
- [Ishida, Toru.] An optimization algorithm for production systems. Knowledge and Data Engineering, IEEE Transactions on (Volume:6, Issue: 4). 549 – 558.
- [Matthias Strobbe, Olivier Van Laere, Bart Dhoedt, Filip De Turck, Piet Demeester.] Hybrid reasoning technique for improving context-aware applications. Knowledge and Information Systems. June 2012, Volume 31, Issue 3, pp 581-616
- [CHRISTINE W. CHAN.] FROM KNOWLEDGE MODELING TO ONTOLOGY CONSTRUCTION. International Journal of Software Engineering and Knowledge Engineering Vol. 14, No. 06, pp. 603-624 (2004)
- [Isaac Lera, Carlos Juiz, Ramon Puigjaner.] Performance-related ontologies and semantic web applications for on-line performance assessment of intelligent systems. Science of Computer Programming, Volume 61, Issue 1, June 2006, Pages 27-37
- [Yuksel Uckan.] Knowledge representation using views in relational deductive data bases. Journal of Systems and Software, Volume 15, Issue 3, July 1991, Pages 217-232.
- [Birgit Vogel-Heuser, Alexander Fay, Ina Schaefer, Matthias Tichy.] Evolution of software in automated production systems: Challenges and research directions. Journal of Systems and Software, Volume 110, December 2015, Pages 54-84.
- [Jahangir Karimi, M.K. Zand.] Asset-based system and software system development – A frame-based approach. Information and Software Technology, Volume 40, Issue 2, 1998, Pages 69-78.
- [Зыков А.А.] Гиперграфы // Успехи математических наук. – 1974. – Т. 29. – Вып. 6. – С. 89–154.
- [Иванчева Н.А., Иванчева Т.А.] Постреляционная СУБД Caché. Новосибирск 2004, Новосибирский государственный университет, Высший колледж информатики, 83-93
- [Wolfgang Kirsten, Michael Ihringer, Mathias Kühn, Bernhard Röhrig.] Object-Oriented Application Development Using the Caché Postrelational Database. Springer verlag. ISBN-13: 978-3540009603. 2003. 191-215
- [Altynbek Sharipbayev and Alibek Barlybayev.] Formal Models of the Intelligent Electronic University. International Journal of Information Technology & Computer Science (IJITCS) ISSN : 2091-1610. Volume 21 : Issue No : 1 / August 2015

[Lian R., Goertzel B., Ke S., O'Neill J., Sadeghi K., Shiu S., Wang D., Watkins O., Yu G.] Syntax-Semantic Mapping for General Intelligence: Language Comprehension as Hypergraph Homomorphism, Language Generation as Constraint Satisfaction. Artificial General Intelligence. Lecture Notes in Computer Science – 2012, Volume 7716 – P. 158-167.

[Zhen L., Jiang Z. Hy-SN:] Hyper-graph based semantic network. Knowledge-Based Systems – 2010, Vol 23, Issue 8 – P. 809-816.

[JSON] <http://www.json.org/json-ru.html>

PRESENTATION AND PROCESSING KNOWLEDGE STORED IN MULTIDIMENSIONAL DATA THOUGHT ENHANCED BASE SEMANTIC HYPERGRAPH

Sharipbay A.A., Barlybayev A.B., Sabyrov T.S.

*Eurasian National University L.N. Gumilyov,
Astana, Kazakhstan*

Sabyrov.Talgat@mail.ru

In this work we consider the presentation and processing knowledge. The main novelty of this research is the nesting of knowledge which is presented through enhanced base semantic hypergraph (EBSH). The mathematical model of presentation of knowledge is built. The storage technology and knowledge processing are described. The client side of knowledge presentation is presented.



УДК 004.89

ПОДХОД К АВТОМАТИЗАЦИИ СОЗДАНИЯ БАЗ ЗНАНИЙ НА ОСНОВЕ ТРАНСФОРМАЦИИ КОНЦЕПТУАЛЬНЫХ МОДЕЛЕЙ

Дородных Н.О., Юрин А.Ю.

*Институт динамики систем и теории управления имени В.М. Матросова СО РАН,
г. Иркутск, Россия*

tualatin32@mail.ru

iskander@icc.ru

Рассмотрена концепция подхода к созданию программных компонентов, обеспечивающих автоматизированную разработку баз знаний (OWL и CLIPS) на основе трансформации концептуальных моделей, представленных в формате XML. Предлагается архитектура сервис-ориентированной программной системы (и ее основных элементов), реализующая данный подход. Описана модель типового программного компонента, включая модель трансляции, на основе которой создаются (специализируются) компоненты. Описан метод (алгоритм) трансформации концептуальных моделей в программный код баз знаний с возможностью уточнения (моделирования) продуктов в нотации RVML.

Ключевые слова: получение знаний; сервис-ориентированная система; концептуальная модель; базы знаний; трансформация моделей.

Введение

В настоящее время разработка новых методов и подходов к созданию интеллектуальных систем (ИС) остается перспективной областью научных исследований. При этом особый интерес представляет процесс формирования баз знаний (БЗ), который всегда считался традиционно «узким местом». На этапе разработки БЗ решаются задачи моделирования предметной области, идентификации (получения), концептуализации (структурирования) и формализации (представления) знаний и их описание на определенном языке представления знаний (ЯПЗ, или языке программирования баз знаний – ЯПБЗ) [Гаврилова и др., 2000]. Дополнительные сложности возникают, если этот процесс носит удаленный и распределенный характер и требует согласования мнений экспертов.

Эффективность создания БЗ может быть повышена путем автоматизации анализа и повторного использования (трансформации) концептуальных моделей предметных областей, построенных при помощи программных средств концептуального, когнитивного, онтологического моделирования или CASE-средств. Существующие в данной области решения обладают рядом недостатков, в частности: отсутствие возможности совместной, распределенной и одновременной

работы пользователей; отсутствие или ограниченность генерации кода БЗ на различных ЯПБЗ; ограниченный набор поддерживаемых форматов концептуальных моделей, а также сложность описания самих моделей для генерации кода.

В связи с этим актуально создание веб-ориентированной (сервис-ориентированной) программной системы, обеспечивающей возможность проектирования БЗ и синтеза кода на различных ЯПБЗ путем трансформации концептуальных моделей.

Таким образом, целью работы является разработка моделей, методов и средств, обеспечивающих автоматизированное создание продукционных БЗ на основе концептуальных моделей, используя промежуточное их представление в виде онтологии и уточнение с использованием специальной графической нотации Rule Visual Modeling Language (RVML) [Грищенко и др., 2013]. Для достижения поставленной цели требуется разработать концепцию сервис-ориентированной программной системы, включая метод трансформации исходных концептуальных моделей в код БЗ на целевом ЯПБЗ.

В качестве источников концептуальных моделей предлагается использовать модели, синтаксис которых выражен на XML (наиболее распространенном формате хранения моделей).

Например, могут быть использованы диаграммы классов UML (Unified Modeling Language) [Booch et al., 2005], представленные в формате XML в соответствии со стандартом XMI (XML Metadata Interchange) [XMI] или концепт-карты, представленные с использованием стандарта XTM [XTM] и др. В качестве целевых ЯПБЗ выбраны CLIPS [Частиков и др., 2003] и OWL [Grau et al., 2008], как одни из широко используемых языков при создании ИС различного назначения.

1. Концепция

На основе результатов анализа моделей, методов и средств автоматизации создания БЗ на основе концептуальных моделей разработана концепция специализированной сервис-ориентированной программной системы.

Основными элементами концепции являются:

- архитектура сервис-ориентированной системы и ее основных элементов, основанные на принципах SaaS;
- модель типового программного компонента, включая модель трансляции и предметно-ориентированный (декларативный) язык для представления и хранения модели трансляции (ЯПМТ);
- модели онтологии и продукций, как основного информационного «ядра» сервис-ориентированной системы;
- метод (алгоритм) создания программных компонентов на основе «клонирования» типового программного компонента и его настройки (специализации), путем установления соответствий (правил трансформации) между элементами метамodelей;
- метод трансформации исходных концептуальных моделей в код БЗ на целевом ЯПБЗ, включающий моделирование (уточнение) продукций в нотации RVML.

1.1. Архитектура системы

Для разработки концептуальной архитектуры сервис-ориентированной системы исследовалась и специфицировалась модель «Программное обеспечение как услуга» (Software-as-a-Service, SaaS) облачного сервиса (системы).

Приведем формальную постановку первой основной задачи. В общем виде SaaS-модель облачного сервиса может быть представлена в виде:

$$SaaS = \langle R, S, A \rangle, \quad (1)$$

где R – набор предоставляемых сервисом ресурсов (услуг) пользователям; S – программное обеспечение, предоставляющее возможность повсеместного и удобного сетевого доступа к ресурсам R ; A – аппаратное обеспечение, как вычислительные узлы (основа) для размещения программного обеспечения S .

Решая задачу разработки архитектуры, специализируем SaaS-модель (1) для сервис-ориентированной системы:

$$SaaS^* = \langle R^*, S^*, A \rangle, \quad (2)$$

где R^* – набор предоставляемых сервис-ориентированной системой ресурсов (услуг) пользователям; S^* – набор сервисов (подсистем), предоставляющий возможность повсеместного и удобного сетевого доступа к ресурсам сервис-ориентированной системы R^* . При этом:

$$S^* = \langle S_{SYS}^*, S_{USR}^*, I \rangle, \quad (3)$$

где S_{SYS}^* – множество системных сервисов обеспечивающих базовое взаимодействие пользователей с сервис-ориентированной системой и предоставляющих возможность проектирования, разработки и регистрации (развертывания) пользовательских прикладных сервисов (программных компонентов) синтеза БЗ; S_{USR}^* – множество разработанных и зарегистрированных прикладных сервисов (программных компонентов) в составе сервис-ориентированной системы; I – интерфейс взаимодействия с внешними ИС.

В свою очередь:

$$I = \{i_1, \dots, i_n\}, i_j = \langle name_j, command_j \rangle, j \in \overline{1, n}, \quad (4)$$

где $name_j$ – наименование j метода взаимодействия; $command_j$ – управляющая команда метода.

Используя (2), уточним множество предоставляемых сервис-ориентированной системой ресурсов (функций) R^* для набора системных сервисов S_{SYS}^* :

- разработка программного компонента (веб-сервиса), на основе типового программного компонента, включая визуальное и текстовое построение соответствий (правил трансформации) элементов исходной и целевой метамodelей;
- хранение информации с использованием специальной онтологической модели и продукций;
- визуальное отображение и редактирование понятий предметной области и их отношений в виде графа (онтологической модели);
- визуальное отображение и редактирование (моделирование) продукций в нотации RVML.

Используя (2), уточним множество предоставляемых сервис-ориентированной системой ресурсов (функций) R^* для набора прикладных сервисов S_{USR}^* :

- формирование модели онтологии на основе автоматизированного анализа концептуальных моделей предметных областей;
- генерация кода БЗ на целевом ЯПБЗ, путем автоматической трансформации модели онтологии;
- генерация кода БЗ на целевом ЯПБЗ на основе прямой автоматической трансформации концептуальных моделей предметных областей.

Таким образом, концептуальная архитектура сервис-ориентированной системы (рисунок 1) позволяет описать ее структуру, включая состав и типы элементов, а также принципиальные особенности функционирования.

Основные типы элементов:

- информационные ресурсы – предназначены для хранения служебной информации, которая используется системными и прикладными компонентами (подсистемами) сервис-ориентированной системы, как для обеспечения собственного функционирования, так и для решения задач автоматизированного формирования БЗ;
- системные сервисы – представляют собой набор всех предлагаемых пользователям сервисов (подсистем), обеспечивающих базовое взаимодействие пользователей с сервис-ориентированной системой и предоставляющие инструментариум для создания прикладных программных компонентов на основе типового;
- прикладные сервисы – представляют собой набор разработанных пользователями программных компонентов, обеспечивающих возможность автоматического синтеза БЗ путем трансформации концептуальных моделей. Программные компоненты создаются на основе типового программного компонента, путем его «клонирования» и настройки (специализации).



Рисунок 1 – Концептуальная архитектура сервис-ориентированной системы

Для унифицированного хранения и представления знаний, извлеченных из концептуальных моделей, разработана специальная онтологическая модель. Разработанная модель позволяет абстрагироваться от особенностей описания знаний в различных ЯПБЗ, используемых при реализации БЗ (например, CLIPS, Jess, Drools,

RuleML, OWL, SWRL и др.), и позволяет хранить знания в собственном независимом формате.

Подробное описание метамодели онтологии приводится в работе [Дородных и др., 2015].

Для поддержки возможности работы со знаниями в виде правил разработана специальная продукционная модель. Главной особенностью продукционной модели является использование специализированной графической нотации Rule Visual Modeling Language (RVML) [Грищенко и др., 2013].

В зависимости от решаемых задач и типов входных данных (исходных концептуальных моделей или модели онтологии) на основе типового программного компонента могут быть созданы разные виды программных компонентов, приведем их классификацию по различным основаниям.

По отношению к информационным ресурсам сервис-ориентированной системы:

- автономные – обеспечивают выполнение операций трансформации и кодогенерации, без использования модели онтологии и продукций (трансформация происходит без использования информационных ресурсов сервис-ориентированной системы);
- интегрируемые – обеспечивают выполнение операций с использованием информационных ресурсов сервис-ориентированной системы.

Интегрируемые программные компоненты имеют доступ к системным сервисам (средствам) визуального проектирования (редактирования) БЗ. Данными средствами являются два специализированных редактора: редактор онтологической модели (визуальное представление и редактирование онтологии) и редактор продукционной модели (визуальное представление и редактирование продукций в нотации RVML).

По типу выполняемых преобразований:

- интегрируемые компоненты анализа – обеспечивают преобразование концептуальных моделей в онтологическую модель и продукций (M2M-преобразование);
- интегрируемые компоненты кодогенерации – обеспечивают преобразование элементов онтологической модели в код БЗ на целевом ЯПБЗ (M2C-преобразование);
- автономные компоненты кодогенерации – обеспечивают преобразование концептуальных моделей в код БЗ на целевом ЯПБЗ (M2C-преобразование).

По характеру использования:

- внешние – фактически размещенные на других серверах, в программной системе непосредственно размещается только описание программных интерфейсов, для реализации используя структурный шаблон «адаптер»;

- внутренние – размещенные непосредственно в составе сервис-ориентированной системы.

Таким образом, программный компонент представляет собой простейший транслятор, который переводит входные концептуальные модели, представленные на одном языке, в выходные БЗ на другом языке.

Одним из важнейших элементов данной технологии является – модель типового программного компонента, на основе которой создается соответствующий программный компонент.

1.2. Модель типового программного компонента

Для повышения эффективности разработки программных компонентов предлагается использовать специальную модель типового программного компонента:

$$M_{TPC} = \langle M_T, A_{IN}, CG_{OUT} \rangle, \quad (5)$$

где M_T – модель трансляции; A_{IN} – анализатор (парсер) входных моделей (исходных концептуальных моделей или расширенной онтологии); CG_{OUT} – генератор кода выходных моделей (расширенной онтологии или БЗ на целевом ЯПБЗ).

Архитектура типового программного компонента может быть представлена на рисунке 2.



Рисунок 2 – Архитектура типового программного компонента

В процессе создания модели типового программного компонента M_{TPC} пользователю необходимо сформировать модель трансляции M_T , на основе которой осуществляется преобразование исходных концептуальных моделей в целевые БЗ.

1.3. Трансформация моделей

Используя (5), определим модель трансляции M_T :

$$M_T = \langle MM_{CM}, MM_{KB}, T \rangle, \quad (6)$$

где MM_{CM} – метамодель исходной (входной) концептуальной модели; MM_{KB} – метамодель целевой (выходной) модели представления знаний (БЗ); T – оператор преобразования моделей.

При этом MM_{ONT} – метамодель онтологии, может выступать в качестве как исходной, так и целевой метамодели в процессе построения M_T .

Используя (6), подробнее опишем элементы модели трансляции M_T :

$$MM_{CM} = \langle E_{CM}, R_{CM} \rangle, \quad (7)$$

где E_{CM} – множество элементов (сущностей) метамодели исходной концептуальной модели; R_{CM} – множество отношений между элементами метамодели исходной концептуальной модели.

$$E_{CM} = \{e_1^{cm}, \dots, e_n^{cm}\}, e_i^{cm} = \langle id_i, name_i \rangle, i \in \overline{1, n}, \quad (8)$$

где id_i – идентификатор i элемента; $name_i$ – наименование i элемента или ссылка на другой элемент $e_{link}^{cm} = \langle id_{link}, name_{link} \rangle$, где id_{link} – идентификатор элемента ссылки; $name_{link}$ – наименование элемента ссылки.

$$R_{CM} = \{r_1^{cm}, \dots, r_m^{cm}\}, r_j^{cm} = \langle r_{LHS_j}^{cm}, r_{RHS_j}^{cm} \rangle, j \in \overline{1, m}, \quad (9)$$

где $r_{LHS_j}^{cm}$ – левая часть связи; $r_{RHS_j}^{cm}$ – правая часть связи. В свою очередь $r_{LHS_j}^{cm} = e^{cm}$ и $r_{RHS_j}^{cm} = e^{cm}$, где e^{cm} – ссылка на элемент.

$$MM_{KB} = \langle E_{KB}, R_{KB} \rangle, \quad (10)$$

где E_{KB} – множество элементов (сущностей) метамодели целевой БЗ; R_{KB} – множество отношений между элементами метамодели целевой БЗ.

Описание элементов MM_{KB} идентично описанию элементов MM_{CM} .

Основным элементом модели трансляции M_T является оператор преобразования T . Таким образом, формализовать постановку третьей основной задачи можно следующим образом. Необходимо определить оператор трансформации концептуальной модели T :

$$T : CM \rightarrow KB, \quad (11)$$

где CM – исходная концептуальная модель; KB – целевая БЗ.

При этом $KB \sim M_{KRL}$, где M_{KRL} – целевая модель на ЯПЗ (KRL).

В соответствии с видами синтаксиса (абстрактного и конкретного) концептуальных языков моделирования и ЯПЗ согласно [Greenfield et al., 2004] уточним (11):

$$T = \langle T_{CM-KB}, T_{CM-ONT}, T_{ONT-KB} \rangle,$$

$$T_{CM-KB} : M_{XML} \Leftrightarrow MM_{CM} \rightarrow MM_{KB} \Leftrightarrow Code_{KRL},$$

$$T_{CM-ONT} : M_{XML} \Leftrightarrow MM_{CM} \rightarrow MM_{ONT} \Leftrightarrow ONT,$$

$$T_{ONT-KB} : ONT \Leftrightarrow MM_{ONT} \rightarrow MM_{KB} \Leftrightarrow Code_{KRL},$$

где T_{CM-KB} – оператор преобразования исходной концептуальной модели в код БЗ на целевом ЯПЗ; T_{CM-ONT} – оператор преобразования исходной концептуальной модели в расширенную онтологию; T_{ONT-KB} – оператор преобразования расширенной онтологии в код БЗ на целевом ЯПЗ; M_{XML} – исходная концептуальная модель, представленная в формате XML; $Code_{KRL}$ – код БЗ, представленный на целевом ЯПЗ; ONT – онтологическая модель;

Оператор преобразования моделей T состоит из трех подоператоров в соответствии с приведенной выше классификацией программных компонентов (по типу выполняемых операций).

Уточним подоператор преобразования T_{CM-KB} : процесс трансформации M_{XML} в $Code_{KRL}$ на целевом ЯПЗ осуществляется путем установления соответствий (определение правил трансформации) между абстрактными элементами исходной метамодели MM_{CM} и целевой метамодели MM_{KB} .

Введем оператор R_T – набор установленных правил соответствия элементов метамodelей (правила трансформации):

$$R_T = (r_1, r_2, \dots, r_n) : MM_{CM} \rightarrow MM_{KB}, \quad (12)$$

где r_i – правило трансформации (продукция). При этом:

$$r_i = \langle e_i^{in}, e_i^{out}, p_i \rangle, i \in \overline{1, n}, \quad (13)$$

где e_i^{in} – исходный элемент метамодели MM_{CM} (левая часть правила); e_i^{out} – целевой элемент метамодели MM_{KB} (правая часть правила); p_i – приоритет выполнения правила, определяющий последовательность выполнения правил, $p_i \in \overline{1, k}$.

Для представления и хранения модели трансляции M_T разработан специальный предметно-ориентированный (декларативный) язык (DSL) – Язык Представления Модели Трансляции (ЯПМТ).

Грамматика разработанного ЯПМТ принадлежит к классу контекстно-свободных грамматик (LL(1)). Конструкции ЯПМТ позволяют в декларативном виде описывать элементы модели трансляции (в основном, установление правил соответствия элементов метамodelей).

2. Метод трансформации концептуальных моделей в БЗ

Метод трансформации представляет собой систематизированную совокупность действий, которые нацелены на решение задачи автоматизированного формирования кода БЗ на целевом ЯПЗ, путем трансформации исходных концептуальных моделей.

В процессе синтеза БЗ может быть использована модель онтологии, которая представляется либо как исходная модель (концептуальная модель), на основе которой синтезируется код БЗ, либо как целевая модель (БЗ), полученная путем автоматизированного анализа исходных концептуальных моделей.

Метод определяет горизонтальную, экзогенную, одностороннюю трансформацию [Mens et al., 2006], т.е. преобразование исходной и целевой моделей одного уровня иерархии, но описанных на различных языках моделирования (разные концептуальные языки моделирования и ЯПЗ).

Основные принципы разрабатываемого метода (подхода) трансформации:

- возможность использования двухступенчатой трансформации (на первом этапе М2М-преобразование, затем М2С-преобразование), что повышает технологичность процесса разработки БЗ, за счет использования онтологии;
- использование продукционной модели, представленной в нотации RVML и обеспечивающей унифицированное описание правил, для дополнения (уточнения) полученных знаний.

В общем виде метод (алгоритм) трансформации исходных концептуальных моделей в код БЗ на целевом ЯПЗ может быть представлен в виде последовательности действий:

На этапе 1 средствами внешних программ пользователь строит концептуальную модель предметной области, которая представляется в формате XML. Этот формат является универсальным и наиболее распространенным способом интеграции программных систем и обеспечения обмена информацией между приложениями.

На этапе 2 в процессе автоматизированного

анализа XML-структуры концептуальной модели выделяются понятия предметной области и их отношения. Анализ происходит на основе разработанной модели трансляции M_T (правил трансформации).

Далее (этап 3) на основе извлеченных понятий и их связей формируется онтологическая модель, как универсальное представление знаний, независящее от исходной концептуальной модели или ЯПБЗ.

На этапе 4 при помощи специальной графической нотации RVML предоставляется возможность визуального отображения и модификации (проверки) полученных зависимостей (продукций).

На этапе 5 происходит автоматическая генерация кода БЗ в формате CLIPS или OWL на основе онтологической модели.

Следует отметить, что этапы 3 и 4 могут отсутствовать, т.к. преобразование исходной концептуальной модели может происходить напрямую в код БЗ, без использования онтологической модели.

Заключение

Решение задач связанных с получением, структурированием и формализацией знаний позволяет повысить эффективность процесса разработки БЗ ИС. В настоящей работе предлагается автоматизировать программную реализацию (синтез кода) БЗ на основе анализа концептуальных моделей.

При решении задач исследования был проведен комплексный анализ области автоматизированного получения (приобретения) знаний, методов и средств трансформации моделей, подходов к разработке программных систем, а также различных стандартов (языков) концептуального, когнитивного и онтологического моделирования. В качестве источников концептуальных моделей предлагается использовать модели представленные на XML. В качестве целевого ЯПБЗ использованы CLIPS и OWL.

На основе разработанных моделей и методов был реализован исследовательский прототип сервис-ориентированной системы для разработки программных компонентов, синтеза БЗ. Одной из возможностей данной системы является обеспечение совместной, распределенной, коллективной работы специалистов-предметников (инженеров по знаниям, программистов и др.) в процессе формирования БЗ для ИС.

Безусловно, данный подход (технология) не позволяет исключить ошибки обусловленные неточностью или неполнотой анализируемых концептуальных моделей, однако, автоматическая генерация декларативного кода на основе этих моделей позволяет использовать принцип быстрого прототипирования при создании БЗ с последующей их проверкой и тестированием (дополнением) в

сторонних средствах. По результатам проверки возможно внесение необходимых изменений в модели (расширенная онтология или производственная модель RVML) и повторная генерация кода БЗ.

Результаты, представленные в статье, получены при частичной финансовой поддержке РФФИ, проекты 15-37-20655, 15-07-03088

Библиографический список

- [Гаврилова и др., 2000] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000. – 384 с.
- [Грищенко и др., 2013] Грищенко М.А., Юрин А.Ю., Павлов А.И. Разработка экспертных систем на основе трансформации информационных моделей предметной области // Программные продукты и системы. – 2013. – №3. – С. 143-147.
- [Дородных и др., 2015] Дородных Н.О., Юрин А.Ю. Использование диаграмм классов UML для формирования продукционных баз знаний // Программная инженерия. – 2015. – №4. – С. 3-9.
- [Частиков и др., 2003] Частиков А.П., Гаврилова Т.А., Белов Д.Л. Разработка экспертных систем. Среда CLIPS. СПб.: БХВ-Петербург, 2003. – 608 с.
- [Booch et al., 2005] Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide, 2nd Edition. Addison-Wesley, New York, 2005, 496 p.
- [Grau et al., 2008] Grau B.C., Horrocks I., Motik B., Parsia B., Patel-Schneider P., Sattler U. OWL 2: The next step for OWL // Web Semantics: Science, Services and Agents on the World Wide Web. – 2008, 6(4), 309-322.
- [Greenfield et al., 2004] Greenfield J., Short K., Cook S., Kent S., Crupi J. Software factories: assembling applications with patterns, models, frameworks, and tools. Wiley Pub., 2004, 696 p.
- [Mens et al., 2006] Mens T., Gorp Van P. A Taxonomy of Model Transformations. Electronic Notes in Theoretical Computer Science. - 2006, №152, 125-142.
- [XMI] Документация спецификации XML Metadata Interchange (XMI). 2015. URL: <http://www.omg.org/spec/XMI> (дата обращения: 10.11.2015).
- [XTM] Документация спецификации XML Topic Maps (XTM). 2015. URL: <http://www.topicmaps.org/xtm> (дата обращения: 10.11.2015).

AN APPROACH FOR DESIGN OF KNOWLEDGE BASES ON THE BASIS OF COMPUTER-AIDED TRANSFORMATION OF CONCEPTUAL MODELS

Dorodnykh N.O., Yurin A.Yu.

Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of the Russian Academy of Sciences (IDSTU SB RAS), Irkutsk, Russia

tualatin32@mail.ru

iskander@icc.ru

The paper describes the concept of the approach for creation of software components designed for development of knowledge bases (OWL and CLIPS) on the basis of the transformation of conceptual models represented in XML. The architecture of the service-oriented software implementing this approach is proposed. The model of a typical software component (including a translation model) and the algorithm for transformation of conceptual models to the program codes with the opportunity to clarify of logical rules in the RVML notation are described.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

СИСТЕМА ПРОГРАММИРОВАНИЯ ПРОДУКЦИОННЫХ БАЗ ЗНАНИЙ: PERSONAL KNOWLEDGE BASE DESIGNER

Грищенко М.А., Дородных Н.О., Юрин А.Ю.

*Институт динамики систем и теории управления имени В.М. Матросова СО РАН,
г. Иркутск, Россия*

makcmg@icc.ru

tualatin32@mail.ru

iskander@icc.ru

Описана система для программирования продукционных баз знаний: «Personal Knowledge Base Designer», ориентированная на непрограммирующего специалиста. Применение системы позволяет автоматизировать этапы формализации знаний и генерации программного кода. Для представления продукций (логических правил) используется обобщенная модель, которая позволяет абстрагироваться от особенностей определенных языков программирования баз знаний. Приведено описание архитектуры и основных функций. Программная система позволяет интегрироваться с CASE-средством IBM Rational Rose, в части импорта концептуальных моделей в форме диаграмм классов UML, описывающих основные предметные понятия и отношения между ними. Программная система обладает расширяемой архитектурой, т.е. реализована возможность подключения динамических библиотек (модулей) поддерживающих различные языки программирования баз знаний. В настоящий момент реализован модуль поддержки языка представления знаний CLIPS (C Language Integrated Production System).

Ключевые слова: анализ концептуальных моделей; программирование; концептуальная модель; базы знаний; генерация программного кода.

Введение

Сложность и трудоемкость процесса разработки экспертных систем (ЭС) обусловлена, главным образом, сложностью и трудоемкостью этапа разработки баз знаний (БЗ), который включает задачи по формализации предметных знаний и их описанию на определенном языке программирования баз знаний (ЯПБЗ) [Гаврилова и др., 2000]. Повышение эффективности решения данных задач, путем их автоматизации, обуславливает необходимость разработки специализированных программных средств. Подобные программные средства в виде редакторов БЗ и специализированных систем программирования (например, Visual Expert System Designer, Expert System Designer, ES-Builder, ДИЭКС и др.), позволяют повысить эффективность процесса разработки за счет использования визуального моделирования, шаблонов представления знаний, автоматизации процесса верификации БЗ и генерации их программного кода.

Одним из перспективных направлений в данной

области является создание редакторов, обладающих способностью интегрироваться с CASE-средствами (например, IBM Rational Rose и др.) и системами когнитивного и онтологического моделирования в части импорта и анализа концептуальных моделей, что и обуславливает актуальность исследований и разработку специализированной программной системы – Personal Knowledge Base Designer [PKBD].

1. Personal Knowledge Base Designer

Разработанное программное обеспечение представляет собой систему программирования продукционных БЗ [Юрин и др., 2012], рассмотрим подробнее его функции и архитектуру.

1.1. Функции

Согласно сформулированным требованиям, Personal Knowledge Base Designer обеспечивает:

- возможность создания элементов продукционных БЗ (шаблонов фактов и правил, а также фактов и правил) непрограммирующим пользователем, благодаря использованию набора

подпрограмм-мастеров, предварительно подготовленных шаблонов фактов и правил, а также обобщенной модели продукций, которая позволяет абстрагироваться от особенностей их описания в разных ЯПБЗ;

- использование авторской нотации RVML (Rule Visual Modeling Language) [RVML] для визуального представления логических правил (продукций);

- интеграцию с CASE-средством IBM Rational Rose, в части импорта концептуальных моделей (диаграмм классов UML), которые могут быть использованы на этапе концептуализации;

- интеграцию с CLIPS (C Language Integrated Production System) [Частиков и др., 2003], в части синтеза отжуждаемого программного кода БЗ, а также его тестирования, путем включения в состав модулей программной системы машины вывода CLIPS;

- возможность функционирования в режиме «проблемно-ориентированный редактор», используя предварительно разработанные описания шаблонов фактов и правил [Берман и др., 2015] и ограничивая возможность их изменения;

- формирование специализированных отчетов.

1.2. Архитектура

С целью реализации требований и функций, разработана архитектура (рисунок 1) [Юрин и др., 2012], включающая следующие основные модули:

- *управления базами знаний* - обеспечивает загрузку и сохранение БЗ в формате ЕКВ – XML-подобный формат программной системы для хранения знаний;

- *управления метауровнем представления знаний* - обеспечивает внутреннее представление продукционной модели знаний, которое не зависит от определенного ЯПБЗ, а также манипулирование (создание, удаление, редактирование) элементами этой модели;

- *управления модулями поддержки ЯПЗ* - обеспечивает подключение и отключение модулей ЯПБЗ, а также доступ к их функциям;

- *интеграции с графическими моделями* - обеспечивает загрузку элементов из UML-моделей (диаграмм классов), построенных в CASE-средстве IBM Rational Rose;

- *управления машинами вывода* - обеспечивает использование машин вывода (в виде динамических библиотек) для тестирования БЗ, включая объяснение полученных результатов;

- *управления семантическим уровнем* - обеспечивает загрузку подготовленных ранее шаблонов фактов и правил (без возможности их изменения), реализуя возможность функционирования системы в режиме «проблемно-ориентированный редактор»;

- *графический пользовательский интерфейс* - обеспечивает доступ к перечисленным функциям.

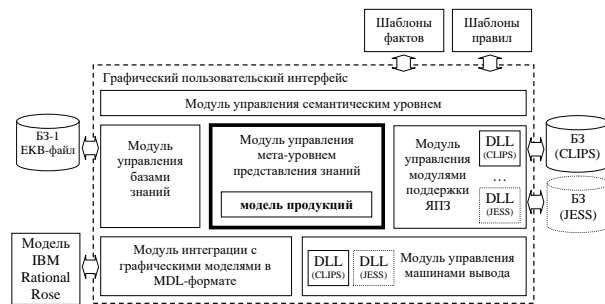


Рисунок 1 – Архитектура системы программирования продукционных БЗ

1.3. Обобщенная модель продукций

Важным элементом программной системы является использование обобщенной модели продукций (рисунок 2) для хранения и представления знаний, включающей понятия: база знаний, шаблон, факт, слот, правило, условие, предусловие, действие, функция, переменная, аргумент.

Разработанная модель позволяет абстрагироваться от особенностей описания продукций в разных языках представления знаний и хранения знания в собственном независимом формате. Программная реализация структуры на уровне хранения знаний выполнена при помощи XML.

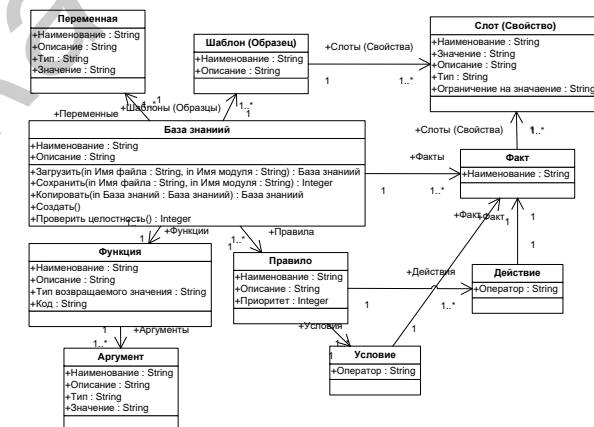


Рисунок 2 – Обобщенная модель продукций

1.4. Интерфейс пользователя

Графический интерфейс пользователя представлен основным рабочим пространством и набором программ-мастеров.

Основное рабочее пространство пользователя разделено на три рабочих области (рисунок 3):

- «Проводник» - отображает элементы БЗ (шаблоны, факты, правила), открытой для редактирования, и содержит кнопочную панель для вызова мастеров, реализующих основные функций манипуляции с элементами БЗ;

- «Информация об объекте» - обеспечивает возможность просмотра описания выбранного элемента БЗ в одной из трех форм: на естественном языке, на ЯПБЗ (например, CLIPS), в виде RVML-схемы [RVML];

- «Справочная информация» - содержит окна справочных сведений и истории действий пользователя (с возможностью их отмены).

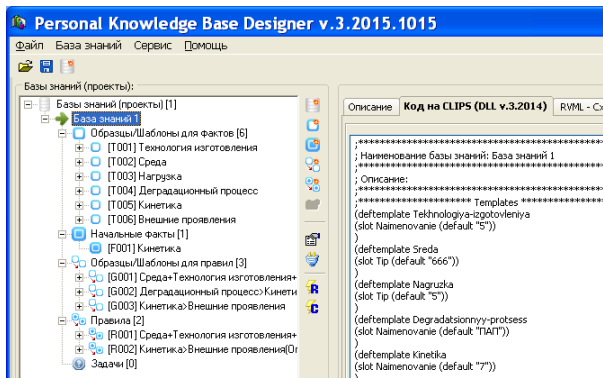


Рисунок 3 – Основное рабочее пространство пользователя

Мастера подставляют собой последовательность экранных форм, сегментирующих и упорядочивающих процессы ввода и редактирования элементов БЗ.

В частности, при вводе шаблона факта пользователю последовательно предлагается задать: имя шаблона (используется для отображения в редакторе), описание и свойства (слоты) (рисунок 4).

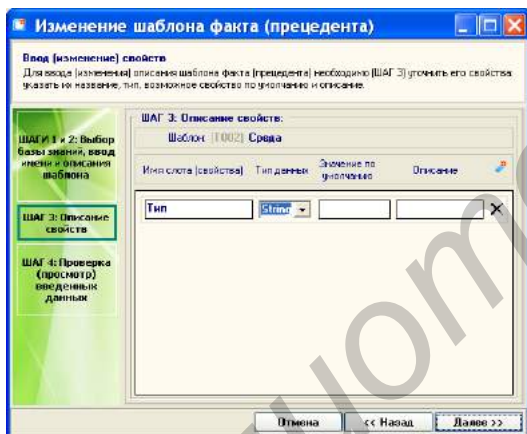


Рисунок 4 – Изменение/Ввод шаблона факта

Подобные мастера применяются при вводе и редактировании фактов и правил (рисунок 5) и при тестировании баз знаний (рисунок 6).

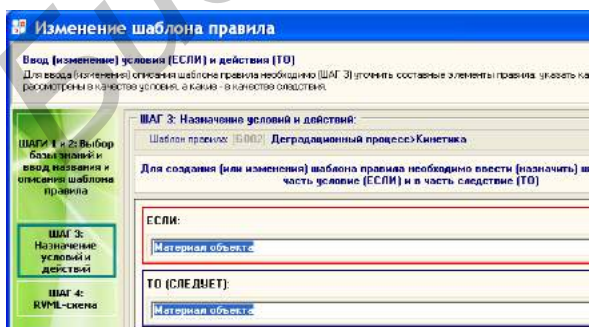


Рисунок 5 – Изменение/Ввод шаблона правила

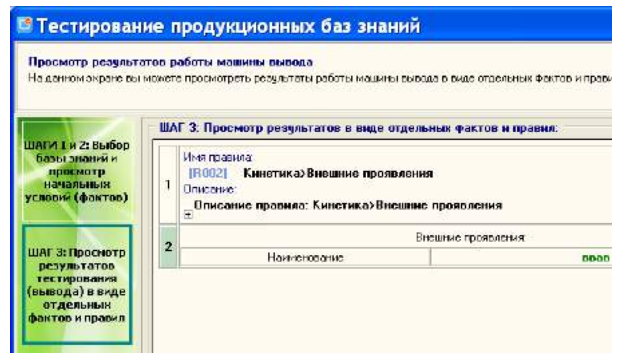


Рисунок 6 – Результаты проверки работоспособности (тестирования)

1.5. Импорт концептуальных моделей

Одним из способов автоматизации формирования БЗ с помощью программной системы является анализ концептуальных моделей.

В частности, анализ модели (рисунок 7) позволяет сформировать описание шаблонов фактов и правил (рисунок 8), на основании которых будут добавлены конкретные элементы фактов и правил. При этом каждый элемент БЗ имеет графическое представление в виде RVML-схемы (рисунок 9).

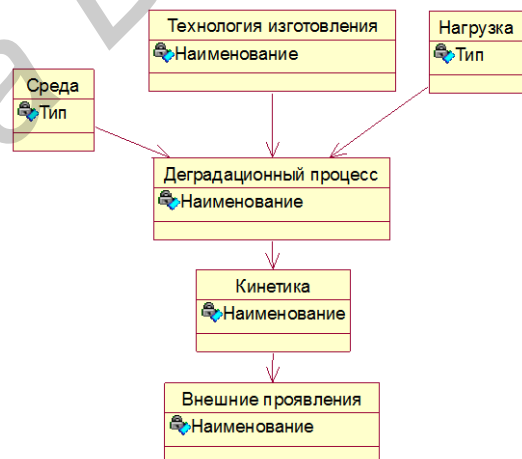


Рисунок 7 – Диаграмма классов UML, описывающая основные понятия и отношения из области диагностики технического состояния

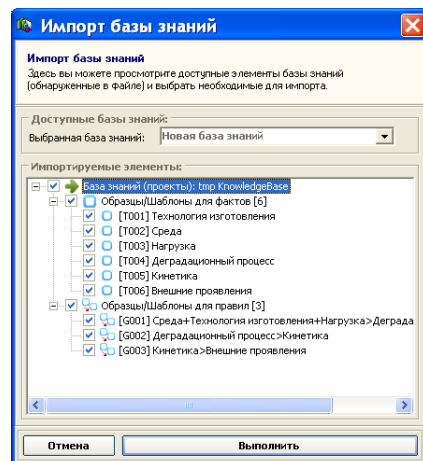


Рисунок 8 – Просмотр импортируемых из mdl-файла (IBM Rational Rose) элементов



Рисунок 9 – Пример RVML-схемы

Заключение

Эффективное создание ЭС и БЗ для решения задач в различных предметных областях требует разработки и использования специализированного инструментария. Одним из видов подобного инструментария являются системы программирования БЗ. Примером подобных систем является Personal Knowledge Base Designer [PKBD]. В работе приведено описание ее функций, архитектуры и интерфейса.

Основными отличительными свойствами программной системы являются:

- *использование обобщенного представления продукции*, обеспечивающего поддержку наиболее распространенных ЯПБЗ (например, CLIPS и др.), которое реализовано в виде модели для хранения знаний, включающей классы: база знаний, шаблон, факт, слот, правило, условие, предусловие, действие;
- *ориентация на непрограммирующего специалиста*. Свойство реализовано с помощью набора мастеров, обеспечивающих описание знаний в виде продукции. Программная реализация данного свойства расширяет область пользователей редактора, за счет экспертов и системных аналитиков, не обладающих навыками программирования и знаниями специализированных ЯПБЗ;
- *модульность*. Возможность расширять поддержку системой различных ЯПБЗ. В настоящий момент реализован модуль поддержки ЯПБЗ CLIPS.

Программная система применялась при выполнении работ по договору № 052013НИР от 19 сентября 2013 г. с ОАО «ИркутскНИИХиммаш» [Берман и др., 2015]. В настоящий момент используется в учебном процессе в Иркутском национальном исследовательском техническом университете (ИрНТУ) при выполнении лабораторных работ по курсам «CASE-средства» и «Инструментальные средства информационных систем».

Представленные результаты получены при частичной финансовой поддержке РФФИ, проекты 15-07-03088, 15-07-05641, 16-37-00041.

Библиографический список

[Берман и др., 2015] Берман А.Ф., Николайчук О.А., Грищенко М.А., Юрин А.Ю. Проблемно-ориентированный

редактор продукционных баз знаний // Программные продукты и системы. – 2015, №2. – С.13-19.

[Гаврилова и др., 2000] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000. – 384 с.

[Частиков и др., 2003] Частиков А.П., Гаврилова Т.А., Белов Д.Л. Разработка экспертных систем. Среда CLIPS. СПб.: БХВ-Петербург, 2003. – 608 с.

[Юрин и др., 2012] Юрин А.Ю., Грищенко М.А. Редактор баз знаний в формате CLIPS // Программные продукты и системы. – 2012, №4. – С.83-87.

[PKBD] Personal Knowledge Base Designer. URL: <http://www.knowledge-core.ru/index.php?p=pkbd> (дата обращения: 12.11.2015).

[RVML] Язык визуального моделирования правил - Rule Visual Modeling Language. URL: <http://www.knowledge-core.ru/index.php?p=rvml> (дата обращения: 12.11.2015)

SOFTWARE FOR RULE KNOWLEDGE BASES DESIGN: PERSONAL KNOWLEDGE BASE DESIGNER

Grishenko M.A., Dorodnykh N.O., Yurin A.Yu.

Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of the Russian Academy of Sciences (IDSTU SB RAS), Irkutsk, Russia

makcmg@icc.ru

tualatin32@mail.ru

iskander@icc.ru

The paper describes software for designing the rule knowledge bases: «Personal Knowledge Base Designer». The software intended for non-programming specialists. Application of the software allows to automate the steps of formalization of knowledge and generation of program codes. The generalized model that abstracts the specific features of knowledge bases programming languages is used for the presentation of the logical rules. The description of the architecture and main functions are presented. The software allows to use files of IBM Rational Rose and import the elements of conceptual models (UML class diagrams) as the basic concepts and relationships between them. The software has an extensible architecture that implemented the ability to connect the dynamic-link libraries (modules) supporting a variety of programming languages for knowledge base design. At this moment the support module for CLIPS (C Language Integrated Production System) is implemented.



УДК 004.822

ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ СТРУКТУРНОГО СИНТЕЗА

Бикмуллина И.И.

*Казанский национальный исследовательский технический университет
им. А.Н. Туполева, г.Казань, Россия*

elsiyar-b@yandex.ru

В статье развивается идея применения и разработки автоматизированного структурного синтеза UML моделей на основе семантических отношений предметной области. Для предлагаемого подхода автоматизированного структурного синтеза информационных систем более подробно рассматриваются объекты UML с семантической точки зрения и решения принятые с целью адекватного, профессионального синтеза диаграммы классов UML.

Ключевые слова: свойства; UML; диаграммы классов; автоматизированный синтез моделей; программирование.

Введение

В современной технологии программного обеспечения (ПО) роль эксперта, рассматриваемой информационной технологией предметной области, находится на уровне консультанта. Так как совершенная технология разработки программ в идеале состоит из трех этапов, где автоматизированным этапом является только третий этап:

1 этап. Разработка онтологии предметной области экспертом этой области с помощью разработчика (программиста). Создаваемая система может опираться на несколько предметных областей, что влечет привлечение к разработке нескольких экспертов. В этом случае от разработчика требуется разобраться в этих разных предметных областях, найти общий язык с экспертами и формализовать данные. Это требует от разработчика владения несколькими специальностями.

2 этап. Моделирование. То есть на основе глоссария, онтологии предметной области разрабатываются диаграммы классов UML [Рамбо и др., 2002]. Моделирование диаграмм UML осуществляется разработчиком вручную или составляется им в приложении, какого - либо ПО. Однако стремление разработчиков как можно быстрее получить программный код приводит к тому, что разработчики используют моделирование диаграмм UML лишь, в крайнем случае, лишь тогда, когда проектирование системы зашло в тупик, что бывает достаточно запоздалым использованием

данного этапа в технологии. Отсюда, моделирование UML моделей не используется в полной мере.

3 этап. Автоматизированная генерация программного исходного кода на основе разработанных моделей UML с помощью Rational Rose или Visual Studio.

Анализ существующей технологии разработки ПО доказывает необходимость разработки инструментов автоматизированного проектирования структурных моделей информационных систем.

В статье [Бикмуллина, 2015] представлено усовершенствование современной технологии разработки, автоматизации ИС (РИС) за счет автоматизированного проектирования диаграмм классов UML на основе семантических моделей.

Автоматизация синтеза диаграммы классов позволяет уменьшить трудоемкость проектирования структурных моделей ИС, повысить качество программного проекта за счет возможности многовариантного анализа структурных моделей, стимулировать разработчика на предварительное продумывание структуры прорабатываемой системы, изменить характер труда программиста в направлении от ручной проработки диаграмм UML к выбору оптимального варианта из множеств автоматически полученных вариантов.

Парадигмы синтеза РИС подразумевают правила синтеза моделей, синтеза объектов диаграммы классов UML. Синтез объектов диаграммы классов UML включают в себя синтез классов, синтез

отношений. Синтез классов помимо основной парадигмы образования класса с помощью свойств РИС, включает в себя определение частных случаев, таких как абстрактный класс, интерфейс. Синтез отношений подразумевает парадигмы для разработки иерархических и неиерархических отношений. При этом если на начальном этапе важность выделения парадигм не столь видна, то на этапе формирования многовариантного моделирования важность данной системы становится очевидной.

Основой для автоматического синтеза структурной модели являются правила принятия решения построения элементов структурных моделей. Каждое правило принятия решений определяет возможные структурные сочетания компонентов и атрибутов. Семантические отношения между составляющими этой модели весьма разнообразны, но они могут быть описаны и представлены с помощью семантических структур. При этом задача РИС сводится к тому, чтобы получить изделие требуемой спецификации. В объектно-ориентированной технологии создания программных изделий спецификация РИС задается системой диаграмм. Система диаграмм – это свойства и основа изготовления РИС. Таким образом, необходимым условием существования РИС является спецификация РИС в виде системы диаграмм. Природа специфики в понимании свойств РИС основывается на разнообразии вариантов абстрагирования, использующихся в разных науках. Отличия типов исследуемых свойств, способов моделирования данных типов во многом определяется дифференциацией наук.

1. Свойства разрабатываемой информационной системы

Процесс РИС - это придание объектам моделирования желаемых свойств, адекватно отражающих необходимые черты моделируемой предметной области.

Свойства РИС описывается в зависимости от особенностей ИС, требований к ней, рассматриваемых как техническое средство, инструмент решения поставленных задач. Особенность – это черта, характеризующая объект с точки зрения его связи с другими объектами: возможности превращения в другой объект под давлением другого объекта, возможности влияния на другой объект. Также программное изделие должно иметь комплекс особенностей. Одной из таких особенностей является та операционная среда, благодаря которой программное изделие превращается в инструмент решения задач автоматизации. Особенность РИС как технического средства - это отношение данного РИС к другим объектам (естественным и искусственным), с которыми программное изделие взаимодействует. Таким образом, особенность – это характерная черта чего-то по отношению к другим предметам. В этом смысле прикладная программа обладает

набором свойств, которые позволяют решать возложенную на нее прикладную задачу. Вычислительная среда, наоборот представляет собой лишь особенность, так как черта, не зависит от прикладной задачи.

В автоматизированных системах современная обработка данных в основном работает с терминами. Для создания интеллектуальных автоматизированных систем, в том числе и систем автоматизации проектирования РИС, нужно от терминов перейти к понятиям. Для этого нужно включить в обработку явным образом выписанные смыслы(все), которые обозначены конкретными терминами. Понятием является языковая модель, четко определяющая смысл. Понятие в отличие от термина (зафиксированного понятия, где смысл зафиксирован) не статично во времени, его смысл меняется достаточно динамично, но при этом сохраняя значительную стабильность. Любой научный термин нагружен базовым списком свойств (многие из которых лишь подразумеваются). И когда в математике говорится «множество», то неявно предполагается, что неважна последовательность его элементов и его элементы не повторяются и т.п. В моделировании деталей машин также имеется много примеров: «консоль», «вал», «втулка» и т.п. Смысл этих терминов известен.

Было принято решение, что понятия (*ConceptsUM*) в диаграмме классов [Бикмуллина, 2015] будут формализоваться по принципу:

$$Concepts_{UML} = \langle Concepts, Relations, Initial_Concept \rangle, \quad (1)$$

где понятия диаграммы классов состоят из множества понятия (*Concepts*), множества отношений (*Relations*) различного вида (иерархические, неиерархические) между данными понятиями и начального понятия (*Initial_Concept*) в описании класса понятий. Рассмотрим вышеизложенные компоненты понятия диаграммы класса более подробно:

$Concepts = \{Concept_{i=1}\}^{conceptscount}$ – конечное непустое множество понятий (включая *Initial_Concept*), при этом $Concept_i$ состоит из:

$$Concept_i = \langle Concept_{Value}, Concept_{Kind}, Concept_{TermType} \rangle, \quad (2)$$

где $Concept_{Value}$ - значение понятия;
 $Concept_{Kind}$ - тип понятия;
 $Concept_{TermType}$ - тип терминального понятия.

Итак, понятие включает в себя:

1. Значение понятия включает в себя:

$$Concept_{Value} = \langle Concept_{ValueType}, Concept_{ValueValue} \rangle, \quad (3)$$

где значение понятия описывается своим типом $Concept_{ValueType}$ и значением $Concept_{ValueValue}$.

$Concept_{ValueType} \in \{ \langle \text{«Строковое»}, \langle \text{«Целое»}, \langle \text{«Вещественное»}, \langle \text{«Логическое»}, \langle \text{«Бинарные данные»} \rangle \rangle \rangle \}$.

Значение понятия есть непустая последовательность символов, которая идентифицирует определенный класс объектов описываемой предметной области, множество значений (сорт) или представляет в ней конкретное (константное) значение:

$$Concept_{Value_Value} \in String \cup Integer \cup Real \cup Boolean. \quad (4)$$

String – множество строк. *Integer* – множество целых чисел. *Real* – множество вещественных чисел. *Boolean* – множество {«true», «false»}.

Значением понятия может быть последовательность символов, интерпретируемых как строковая константа ($Concept_{ValueType} = \text{«Строковое»}$), целое число из множества целых чисел ($Concept_{ValueType} = \text{«Целое»}$), вещественное число из множества вещественных чисел ($Concept_{ValueType} = \text{«Вещественное»}$), элемент множества {«true», «false»} ($Concept_{ValueType} = \text{«Логическое»}$), ($Concept_{ValueType} = \text{«Бинарные данные»}$).

2. Тип понятия включает в себя:

$Concept_{Kind} \in \{\text{«Терминальное»}, \text{«Нетерминальное»}\}$. То есть понятие может быть терминальным ($Concept_{Kind} = \text{«Терминальное»}$) или нетерминальным ($Concept_{Kind} = \text{«Нетерминальное»}$).

3. Тип терминального понятия состоит из:

$Concept_{TermType} \in \{\text{«Идентификатор»}, \text{«Константа»}, \text{«Сорт»}, \text{«Не определено»}\}$. Терминальное понятие может быть идентификатором ($Concept_{TermType} = \text{«Идентификатор»}$), константой ($Concept_{TermType} = \text{«Константа»}$), или обозначать некоторый сорт ($Concept_{TermType} = \text{«Сорт»}$).

Если понятие $Concept_i$ является нетерминальным, то $Concept_{TermType} = \text{«Неопределено»}$.

Также на понятия накладываются ограничения, которые могут отсутствовать, если текстовое представление информации не требуется.

Описание синтаксических ограничений:

$$Syntax_Restrictions = \langle Lexica, Syntax \rangle, \quad (5)$$

где *Lexica* - лексические ограничения; *Syntax* - синтаксические ограничения.

Описание лексических ограничений:

$$Lexica = \{ \langle Lexem_Type_i, Definition_i \rangle \}_{i=1}, \quad (6)$$

где $Lexem_Type_i \in \{\text{«Идентификатор»}, \text{«Целое число»}, \text{«Вещественное число»}, \text{«Строковая константа»}, \text{«Ограничитель строковой константы»}\}$; $Definition_i$ – строка символов, включающая метасимволы и представляющая конкретное лексическое ограничение для вида лексем.

Синтаксические ограничения задаются для нетерминальных понятий:

$$Syntax = \{ \langle Concept_i, Definition_i \rangle \}, \quad (7)$$

где $Concept_i \in Concepts \setminus Terminal_Concepts$; $Definition_i$ – строка символов, включающая метасимволы и представляющая конкретное синтаксическое ограничение для конкретного понятия.

Далее возникает необходимость в предоставлении связей, используемых для формулировки сочетаемости понятий:

1) «И»,

$$P(x) = P_1(x) \wedge P_2(x) \wedge \dots \wedge P_n(x). \quad (8)$$

Таким образом, например, фраза «БПЛА (беспилотный летательный аппарат) может быстро доставить небольшую посылку в любую точку города, минуя пробки. Кроме того, с его помощью можно доставлять посылки в труднодоступные места» будет записано как «БПЛА(x) = беспилотный летательный аппарат(x) \wedge быстро доставляющий небольшую посылку в любую точку города(x) \wedge минуя пробки(x) \wedge доставляющий посылки в труднодоступные места(x)».

2) «НЕ»,

$$P(x) = \neg P_n(x). \quad (9)$$

Отсюда, например, фраза «Беспилотный летательный аппарат — это летательный аппарат без экипажа на борту» будет записано как «беспилотный летательный аппарат(x) = летательный(x) \wedge аппарат(x) \wedge \neg экипажа на борту(x)».

3) «ИЛИ»,

$$P(x) = P_1(x) \vee P_2(x) \vee \dots \vee P_n(x). \quad (10)$$

Например, фраза «Беспилотный летательный аппарат — это любое удаленно управляемое или вовсе самостоятельное (интеллектуальное) средство» будет записано как «беспилотный летательный аппарат(x) = (самостоятельное(x) \wedge интеллектуальное(x) \vee удаленно управляемое(x)) \wedge средство(x)».

Также из выше приведенного примера видно, что приоритет у «И» выше чем у «ИЛИ», то есть между ними взаимоотношения такие же, как между умножением и сложением или логическим & и \vee . Правила структурной сочетаемости понятий:

1) «горизонтальное объединение» – объединение подряд идущих понятий в словосочетание.

2) «целое-часть», $y : x_1 \dots x_n$. Где слева в формуле находится «целое» в виде термина - определяемое понятие, а справа находятся понятия – «части» данного «целого». При этом возможен вариант как строгого включения частей (как неотъемлемые компоненты одного целого), то есть обязательного присутствия и в последствие включения всех частей в целое $y : x_1 \wedge x_2 \wedge \dots \wedge x_n$, либо не строгого включения (компоненты достаточно самостоятельны, чтобы существовать

отдельно от целого) $y : x_1 \vee x_2 \wedge x_3 \vee \dots \vee x_n$. Так, например, высказывание «Хвостовое оперение состоит из двух частей: киль и стабилизатор» будет записано как «*хвостовое оперение*» : «*киль*» \wedge «*стабилизатор*».

Характеристика понятия «целое / часть» отражает способность понятия структурно присоединять другие понятия или, наоборот, присоединяться к господствующему компоненту сочетания [Барков, 2003]. В работе части в обоих случаях будут относиться к атрибутам, а целое к названию класса. И применяется данное правило для фиксации структурных свойств объекта, а также его характеристик. Однако на практике правила образования структур «целое-часть» недостаточно, так как правила структурной сочетаемости имеют более богатый набор средств интересующие нас сочетаемости. Также возможно пересечение соседних фраз, когда окончание одной является уже началом следующей.

3) Установление тождественных понятий. «Поиск в модели экстенционально тождественных или интенционально тождественных элементов (например, с целью поиска тех элементов, являющихся в сечении кругом, являющимися телами вращения)» [Барков, 2003].

4) «Зависимость» - характеристика зависимости понятия, обозначает обязательность совместного использования структурных элементов изделия [Барков, 2003]. Данное правило набирает силу при определении наиболее устойчивых словосочетаниях, например, «*искусственный интеллект*» или когда к понятию привязывается его характеристика, доопределяющая рассматриваемое понятие, например, «*воздушный шар*», «*кусок металла*». Также когда объекты конкретной предметной области друг без друга являются бесполезными элементами, например словосочетание, «*болт, гайка*».

5) «Компонентность», обозначает отнесение параметров рассматриваемого понятия, его характеристик (например, «*размер моста*», так «*размер*» становится атрибутом понятия «*мост*») к его атрибутам, то есть к атрибутивным свойствам изделия, например, «*габаритный размер*» [Барков, 2003].

6) «Наследовательность», означает, что у понятий есть что-то общее («*воздухоплавательный аппарат*» и «*аэростат*») и между ними есть смысловая ассоциация «целое-часть», например, «*машина*» и «*легковая машина*», «*грузовая машина*».

Для построения интеллектуальных систем необходимо содержание понятия определить как описание его свойств. Для выделения объема понятия необходимо процедурным путем вычислить свойства рассматриваемых объектов. Итак, свойство является описанием, характеризующим данную предметную область и позволяющим точно определить программное изделие при необходимой степени декомпозиции. При отсутствии этой черты

программное изделие превращается в другое изделие.

2. Роль отношений в моделируемой диаграмме классов

Задачей в исследовании является построение системы бинарных отношений *Relation*.

$Relations = \{Relation_{i=0}\}^{count}$ – конечное, возможно пустое, множество отношений. Каждое отношение $Relation_i$ является направленным бинарным отношением, имеющим, возможно, спецификатор множественности и связывающим два понятия. Отношение описывается следующим образом:

$$Relation_i = \langle Relation_{EndSp}, Begin_Concept, End_Concept \rangle, \quad (11)$$

где $Relation_{EndSp}$ – спецификатор множественности;
Begin_Concept - понятие-начало отношения;
End_Concept - понятие-конец отношения.

Рассмотрим более подробно:

1. Спецификатор множественности (кардинальность) является характеристикой отношения, определяющий способ порождения понятия-экземпляра вместе с отношением к нему по понятию-концу данного отношения:

$Relation_{EndSp} \in \{\langle \text{«Конкретность»}, \langle \text{«Единственность»}, \langle \text{«Множество»} \rangle\}$, где

1) $Relation_{EndSp} = \langle \text{«Конкретность»} \rangle$ (пустой спецификатор) означает, что может быть порождено только одно понятие-экземпляр и его значение (имя) должно совпадать со значением (именем) его понятия-прототипа (*End_Concept*).

2) $Relation_{EndSp} = \langle \text{«Единственность»} \rangle$ означает, что по *End_Concept* может быть порождено только одно понятие-экземпляр, но его значение (имя) обязательно должно быть задано при порождении. Порожденное понятие при этом является сущностью из класса объектов или элементом из множества значений, идентифицируемого понятием-прототипом (*End_Concept*) (строка, целое значение, вещественное значение, логическое значение).

3) $Relation_{EndSp} = \langle \text{«Множество»} \rangle$ означает, что по *End_Concept* может быть порождено любое количество (но, по крайней мере, одно) понятий-экземпляров и их значения (имена) обязательно должны быть заданы при порождении. Каждое порожденное понятие при этом является сущностью из класса объектов или элементом из множества значений, идентифицируемого понятием-прототипом (*End_Concept*).

2. *Begin_Concept* – понятие, из которого исходит – понятие-начало отношения.

$$Begin_Concept \in Concepts \setminus Terminal_Concepts, \quad (12)$$

где *Terminal_Concepts* – множество терминальных понятий, $Terminal_Concepts \subset Concepts$. Понятием-началом отношения может быть любое нетерминальное понятие.

3. *End_Concept* – понятие, в которое входит – понятие-конец отношения:

$$End_Concept \in Concepts \setminus \{Initial_Concept\}, \quad (13)$$

где *Initial_Concept* – начальное понятие в описании класса понятий, оно единственно, и через него не может быть выражено ни одно другое понятие из описываемого класса понятий:

$$Initial_Concept \in Concepts. \quad (14)$$

Понятием-концом отношения может быть любое понятие за исключением *начального понятия*.

Бинарные отношения синтезируются учитывая следующую классификацию отношений определенную на основе анализа:

1) виды отношений взаимодействия классов:

1.1) иерархические

а) ассоциация:

$$R_{AS}(O_4) \{C_i(O_4) \times C_j(O_4)\} \quad (15)$$

«используется для», «находится в процессе», «быть результатом», «быть полученным при», «служить параметрами», «быть полученным с помощью», «использоваться для» [Глоба и др., 2013].

i) один к одному,

ii) один ко многим,

iii) многие к одному.

б) наследования (обобщения),

$$R_n(O_4) = a_i, r_i | A_{Cm}(O_4) \longrightarrow a_i, r_i | A_{Ck}(O_4) \quad [\text{Глоба и др., 2013}]. \quad (16)$$

Класс и наследование атрибутов и отношений его подклассами:

$$A(C_1), R(C_1) \longrightarrow A(C_{11}), R(C_{11}), A(C_1), R(C_1) \longrightarrow \longrightarrow A(C_{12}), R(C_{12}), A(C_1), R(C_1) \longrightarrow A(C_{13}), R(C_{13}) \quad [\text{Глоба и др., 2013}]. \quad (17)$$

1.2) структурные отношения («часть-целое»)

$$C_1 \subset C_{11} \wedge C_{12} \wedge C_{13} \quad [\text{Глоба и др., 2013}]. \quad (18)$$

а) агрегации,

б) композиции.

в) «класс-данные» (класс – атрибуты, класс - функции)- «входит в состав», «является характеристикой».

$$R_{CD}(O_4) = C_j(O_4) \subseteq D_i(O_4). \quad (19)$$

Отношение данного вида (формула 19) реализовано для всех классов данной онтологии $C_1(O_4) \subseteq D$, $A_{C1} \subseteq A_D$ [Глоба и др., 2013], то есть $A \subseteq A_D$, $C_1 \subseteq A$

1.3) неиерархические

а) реализация,

б) зависимость.

2) виды отношения доступности (разновидности методов доступа):

2.1) без ограничений (public),

2.2) закрытое (private),

2.3) защищенное (protected).

3) для языков с ООП

3.1) конструктор (C++, C#),

3.2) деструктор (для языков C++)

4) виды классов

4.1) абстрактное

4.2) интерфейс

4.3) шаблон

4.4) утилита

Где O_4 – онтология; C_1 – класс, A -атрибуты, R -отношения, D – вид атрибута, $O_4 = \{C, A, R, D\}$.

Перечисленные отношения составляют основу диаграммы классов.

Иерархическая модель – это абстрактных данных ранжированная модель. Из выше представленной классификации отношение наследования означает наличие сущностей предка и потомка, а также передача предком потомку свои методы, структурные части.

Отношение наследования помимо выше сказанного может создавать иерархию, в которой существует общая сущность, может модифицироваться, конкретизоваться на более частные, конкретные сущности. Отношение агрегации и композиции являются одними из представителей иерархии целое – часть. В агрегации при этом части существуют самостоятельно, а в композиции части опекаются разбиваемой сущностью.

Отличие иерархических и неиерархических отношений заключается в существовании в неиерархическом отношении понятия «братья». То есть для описания неиерархического понятия возникает необходимость расширения описания иерархического понятия. Особенности неиерархических отношений:

1) в ряде случаев рассматриваемые отношения обладают иерархическими свойствами и вовсе не несут неиерархическую нагрузку. Например, одно из отношений зависимости может использоваться с такой же целью как иерархическое структурное отношение агрегации. Таким образом, возможен вариант, когда неиерархическое отношение воспринимается, как иерархическое и обладает лишь иерархическими свойствами;

2) чаще всего неиерархические отношения строятся на базе иерархических отношений, то есть отношение, связывающее два класса, обладает как иерархическими свойствами, так и неиерархическими.

Из выше представленных данных и на основе анализа вытекает, что описание неиерархического понятия базируется на иерархическом понятии. В рассмотренном в статье [Бикмуллина, 2015] примере (о нахождении среднего значения и дисперсии) одним из возможных вариантов выборки видов отношений является:

– наследования (от среднего значения к дисперсии),

– агрегации {(файл, запись), (МЭД, файл), (массив, запись)} или

– композиции {(МЭД, массив), (МЭД, файл), (массив, запись)} или

– сервер-клиент {(файл, МЭД), (запись, массив)}).

Этот этап в данной статье рассмотрен не достаточно подробно, чтобы представить всю систему парадигм, однако цель данной статьи заключалась в представлении общей картины данной системы.

Заключение

В работе более подробно рассматриваются объекты UML с семантической точки зрения и решения принятые с целью адекватного, профессионального синтеза моделей диаграммы классов UML и в дальнейшем включаемые в усовершенствование современной технологии РИС за счет автоматизированного проектирования диаграмм классов UML на основе семантических моделей.

Библиографический список

[Рамбо и др., 2002] Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник – СПб.: Питер, 2002. - 656 с.

[Бикмуллина, 2015] Бикмуллина И.И. Технология автоматизированного синтеза информационных систем с помощью семантических моделей предметной области/ И.И. Бикмуллина// OSIS, 2015, С. 445-450

[Барков, 2003] Барков И.А. Теория конструкторской семантики/ И.А. Барков// ИжГТУ, Ижевск, 2003, С. 30

[Глоба и др., 2013] Глоба Л.С., Терновой М.Ю., Новогрудская Р.Л. Метод организации слабосвязанных информационных ресурсов на порталах знаний / Л.С. Глоба, М.Ю. Терновой, Р.Л. Новогрудская// Минск, OSIS, 2013, С. 49-54

PROGRAMMING PARADIGMS INFORMATION TECHNOLOGY STRUCTURAL SYNTHESIS

Bikmullina I.I.

*Kazan National Research Technical University
named after A. N. Tupolev (KAI), Kazan, Russia*

elsiyar-b@yandex.ru

The paper develops the idea of the application and development of computer-aided structural synthesis of UML models based on semantic relations of the subject area. The proposed approach for automated structural synthesis of information systems described in more detail the UML objects from the semantic point of view and decisions taken for adequate, professional synthesis UML class diagrams.

Introduction

In modern software technology the role of the expert under consideration of the information technology subject area, is at the level of a consultant. As modern technology development programs ideally consists of three steps, where the automated stage is the third stage:

Stage 1. Development of domain ontology by expert this field by using a developer (programmer). The system can rely on a few subject areas, which implies the involvement of several experts. In this case, the designer has to understand these different subject areas, to find a common language with experts and to formalize the data. This requires the developer to own several specialties.

Stage 2. Modeling. That is based on the Glossary of ontology developed the UML class diagram [Rumbaugh et al., 2002]. Modeling language UML diagrams is provided by the developer manually or composed them in the app any software. However, the desire of developers as soon as possible to get the code leads to the fact that developers use modeling diagrams UML only in the extreme case where system design is at an impasse, which is quite late by this stage in the technology. Hence, modelling with UML models is not fully used.

Stage 3. Automated generation of source code based on the developed models using UML rational rose or visual Studio.

Analysis of the existing technologies of software development proves the need to develop tools for automated design of structural models of information systems (IS).

In the article [Bikmullina, 2015] presents the improvement of modern technology development, IS automation through computer-aided design of class diagrams of UML based semantic models.

Automation of synthesis of a class diagram makes it possible to reduce the complexity of designing structural IS models to improve the quality of a software project by allowing multivariate analysis of structural models, to stimulate developer's preliminary thinking patterns are working on the system, change the nature of the work of the programmer from manual study of the UML diagrams to the choice of the optimal variant of the sets automatically received options.

The paradigm of synthesis of the development, automation of IS rules imply synthesis models, synthesis of objects of the UML class diagram. The synthesis of objects class diagrams UML class diagrams include the synthesis, the synthesis of relations. The fusion classes in addition to the basic paradigm of education class using properties of the design, automation of IS involves identifying particular cases, such as abstract class, interface. The synthesis of the relationship implies the paradigm for the development of hierarchical and non-hierarchical relations. If at the initial stage the importance of distinguishing paradigms are not so visible, at the stage of formation of the multivariate modeling, the importance of this system becomes obvious.

The basis for automatic synthesis of structural models are the rules the decision of constructing the elements of structural models. Each rule of a decision specifies the possible structural combinations of components and attributes. Semantic relations between the components of this model are very diverse, but they can be described and represented using semantic structures. The development task of automation IS is to get the product to the required specification.

Conclusion

The paper presents more detail the UML objects from the semantic point of view and decisions taken adequate, professional synthesis models UML class diagrams in the future be included in the improvement of modern technology development, IS automation through computer-aided design of UML class diagrams on the basis of semantic models.



УДК 004.822:514

О ДВУХ КЛАССАХ СЕМАНТИЧЕСКОЙ КОРРЕЛЯЦИИ, НАБЛЮДАЕМЫХ В ФУНКЦИОНАЛЬНЫХ СЕМАНТИЧЕСКИХ СЕТЯХ

Савкин Л.В.

Публичное акционерное общество «Радиофизика»,
г. Москва, Россия

android4.1@mail.ru

В работе рассмотрены два класса семантической корреляции, которые предложено использовать в процедурах пофрагментной верификации функциональных семантических сетей. В качестве первого класса выступает структурно-функциональная корреляция, основанная на анализе степени сходства орграфов, описывающих выделенные фрагменты сети. В качестве второго класса выступает референциальная корреляция, основанная на сопоставлении верхних (итоговых) семантических значений верифицируемых фрагментов.

Ключевые слова: функциональная семантическая сеть; верификация; класс; корреляция; структурно-функциональный; референциальный.

Введение

Одной из фундаментальных и конструктивно сложных задач, решаемых в процессе построения эффективных функциональных семантических сетей (ФСС), является создание качественного формального аппарата, позволяющего избавиться от противоречивости описания объектов и предикатных соотношений между ними [Болотова, 2012; Кузнецов О.П., 2014]. С ростом общей сложности (разнородности) ФСС вполне естественным образом возникает проблема локальных функциональных коллизий, регистрируемых в процессе взаимодействия объектов как в пределах одного уровня иерархии ФСС, так и в рамках примыкающих друг к другу соседних иерархических уровней.

В последнем случае речь идет о рассмотрении ФСС в виде структурированной совокупности функциональных сценариев [Беляев, 2013], каждый из которых представляет собой строго ограниченный набор базовых объектов типа «вход-выход» и образует тем самым независимый функциональный фрагмент ФСС. В связи с этим ставится вопрос о верификации функциональных фрагментов ФСС.

Цель работы – рассмотрение двух классов семантической корреляции фрагментов ФСС, связанных со способами выбора корреляционных признаков сопоставляемых фрагментов.

1. Общая задача семантической корреляции фрагментов ФСС

1.1. Представление фрагментов ФСС в виде орграфов NP-полных задач

Ключевой идеей данной работы является тот факт, что задачу семантической корреляции двух и более фрагментов ФСС всегда можно рассматривать с точки зрения двух аспектов.

Во-первых, независимо от сложности того или иного фрагмента ФСС, его внутреннюю структуру всегда можно представить в виде орграфа $G(F, X)$ (рис. 1), для которого F будет являться множеством функциональных объектов типа «вход-выход», а X будет представлять собой множество предикатных дуг (направленных ребер) [Касьянов, 2003].

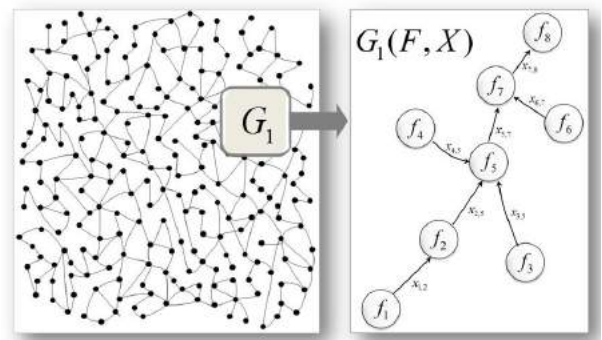


Рисунок 1 – Орграф фрагмента ФСС

В качестве предикатных дуг далее мы будем рассматривать дуги, описываемые весовыми коэффициентами $\beta = 1$, что будет означать полное

отсутствие функциональных поправок при передаче промежуточных функциональных значений от одного объекта ФСС к другому [Венда, 1990]. Кроме того, необходимым условием оптимальности топологии фрагмента ФСС будет являться возможность реализации на орграфе $G(F, X)$ NP-полных задач, что, в свою очередь, позволит обеспечить верификацию фрагментов ФСС не только со «смысловой» (приводящей к единственному значению, выводу, атрибуту или слову) точки зрения, но и с топологической. Данное условие позволит снизить неоднозначность итоговых функциональных значений, формируемых на основе графоаналитических характеристик верифицируемых фрагментов ФСС.

Во-вторых, при анализе выходных данных функциональных сценариев, состоящих из независимых фрагментов ФСС, не всегда требуется знать структурно-функциональные особенности (т. е. орграф) всех задействованных в реализации сценария фрагментов. Достаточно будет знать лишь выходные (или «верхние») значения функциональных объектов орграфов, которые в совокупности реализуют требуемые сценарии [Хорошевский, 2013; Вагин и др., 2008]. Это обстоятельство иллюстрирует пример ФСС, содержащей как минимум два уровня иерархии.

1.2. О понятии «семантической корреляции» применительно к ФСС

Рассмотрим для начала общее определение семантической корреляции, которое предлагается использовать в рамках большинства моделей ФСС.

Определение 1. Под *общей семантической корреляцией* фрагментов ФСС будем понимать степень сходства их внутренних (структурно-функциональных) или итоговых (референциальных) семантических значений, формируемых посредством конечного набора предикатно связанных между собой функциональных объектов, образующих сценарные фрагменты ФСС.

В виду вышеотмеченной в п.п. 1 вариативности подходов к верификации ФСС с позиций семантической корреляции, которая также отражена в определении общей корреляции ФСС, перейдем к рассмотрению двух соответствующих классов семантической корреляции.

2. Классы семантической корреляции фрагментов ФСС

В зависимости от того, как мы будем уточнять определение общей семантической корреляции ФСС, можно выделить два самостоятельных класса: структурно-функциональная корреляция и референциальная корреляция.

Рассмотрим эти классы более подробно.

2.1. Класс №1. Структурно-функциональная корреляция

В качестве определения первого класса семантической корреляции фрагментов ФСС предлагается использовать следующее:

Определение 2. Под *структурно-функциональной семантической корреляцией* фрагментов ФСС будем понимать степень сходства орграфов, полностью описывающих структурные (предикатные, атрибутивные и т.п.) и функциональные (объектные) особенности построения фрагментов ФСС.

Поскольку в данном классе семантической корреляции ФСС речь идет об анализе степени сходства орграфов выделенных фрагментов ФСС, то все подклассы класса структурно-функциональной корреляции будут определяться лишь выбором конкретного способа сопоставления орграфов верифицируемых фрагментов.

2.1.1. Прямое сопоставление орграфов и их анализ на наличие морфизмов

Рассмотрим рис. 2, на котором представлены два орграфа $G_1(F, X)$ и $G_2(F, X)$, каждый из которых полностью описывает структурно-функциональные особенности соответствующих им выделенных фрагментов ФСС.

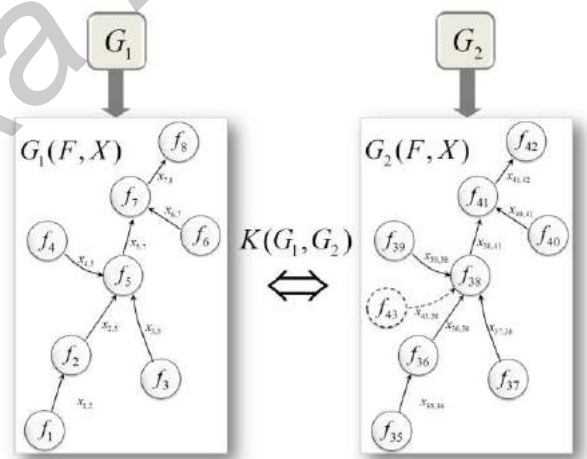


Рисунок 2 – К пояснению случая структурно-функциональной корреляции двух фрагментов ФСС

Исходя из определения 2, для двух орграфов, описываемых соответствующими выражениями

$$G_1(F, X) = G_1(f_j \in F, x_k \in X), \quad (1)$$

$$G_2(F, X) = G_2(f_L \in F, x_M \in X), \quad (2)$$

где f_j и f_L – наборы функциональных объектов, а x_k и x_M – наборы предикатных дуг, выражение для структурно-функциональной корреляции орграфов (1) и (2) можно представить в следующем виде

$$K(G_1, G_2) = \begin{cases} f_j \Leftrightarrow f_L, \\ x_k \Leftrightarrow x_M. \end{cases} \quad (3)$$

Вполне очевидно, что выражение (3) учитывает лишь проверку на соответствие между наборами функциональных объектов типа «вход-выход» и

наборами предикатных дуг орграфов выделенных фрагментов ФСС $G_1(F, X)$ и $G_2(F, X)$. Поэтому, более детальным анализом степени сходства рассматриваемых орграфов будет являться их анализ на предмет наличия морфизмов. В зависимости от выявленных типов морфизмов орграфов $G_1(F, X)$ и $G_2(F, X)$ можно будет выделить самостоятельные подклассы структурно-функциональной корреляции между фрагментами ФСС: изоморфный подкласс, гомоморфный подкласс и т. п.

Возвращаясь к выражению (3), необходимо также отметить, что саму процедуру сопоставления наборов элементов множеств F и X во многих случаях будет удобно реализовать, используя матрицы смежности и матрицы инцидентности. В таких случаях необходимо будет исходить из сложности (в первую очередь топологической) самих орграфов.

2.1.2. «Слепая» количественная оценка орграфов

Критерием структурно-функциональной корреляции орграфов $G_1(F, X)$ и $G_2(F, X)$, безусловно, может служить и «слепая» количественная оценка числа функциональных объектов типа «вход-выход» и предикатных (или атрибутивных) дуг.

В этом случае выражение (3) можно будет представить в следующем виде

$$K^N(G_1, G_2) = \begin{cases} \sum_J f^{(G_1)} \Leftrightarrow \sum_L f^{(G_2)}, \\ \sum_K x^{(G_1)} \Leftrightarrow \sum_M x^{(G_2)}, \end{cases} \quad (4)$$

где $\sum_J f^{(G_1)}$ и $\sum_L f^{(G_2)}$ – общее (суммарное) число функциональных объектов типа «вход-выход», относящихся к соответствующим орграфам $G_1(F, X)$ и $G_2(F, X)$, а $\sum_K x^{(G_1)}$ и $\sum_M x^{(G_2)}$ – общее число их предикатных дуг.

2.2. Класс №2. Референциальная корреляция

В качестве определения второго класса семантической корреляции фрагментов ФСС предлагается использовать следующее:

Определение 3. Под референциальной (т. е. соотносящейся по значению) корреляцией фрагментов ФСС будем понимать степень семантического сходства фрагментов ФСС в целом, окончательные (итоговые) значения которых формируются на выходах верхних функциональных объектов в орграфах фрагментов ФСС.

Класс референциальной корреляции фрагментов ФСС не принимает во внимание структурно-функциональные особенности орграфов верифицируемых фрагментов.

Рассмотрим рис. 3, на котором представлены два фрагмента ФСС с одинаковыми итоговыми

семантическими значениями, формируемыми в верхних объекта f_8^{up} и f_{43}^{up} .

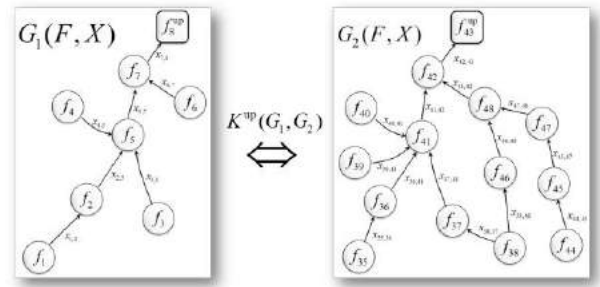


Рисунок 3 – К пояснению случая референциальной корреляции двух фрагментов ФСС

Из данного рисунка видно, что орграфы $G_1(F, X)$ и $G_2(F, X)$, описывающие структурно-функциональные особенности фрагментов ФСС совершенно различны.

Выражение для референциальной корреляции двух фрагментов ФСС можно представить в виде

$$K^{\text{up}}(G_1, G_2) = \begin{cases} f_J^{\text{up}} \Leftrightarrow f_L^{\text{up}}, \\ x_K \neq x_M, \end{cases} \quad (5)$$

где f_J^{up} и f_L^{up} – верхние функциональные объекты орграфов типа «вход-выход», формирующие итоговые семантические значения выделенных фрагментов ФСС. Неравенство наборов x_K и x_M в упрощенной форме отражает факт различных топологических характеристик орграфов $G_1(F, X)$ и $G_2(F, X)$.

Таким образом, основной идеей использования референциальной корреляции при верификации фрагментов ФСС является возможность наблюдения за верхними семантическими значениями функциональных объектов с целью их сравнения и последующего выявления семантического сходства.

Заключение

В работе были рассмотрены два класса семантической корреляции, которые можно использовать при пофрагментной верификации ФСС различных типов. Первый класс семантической корреляции фрагментов ФСС учитывает степень сходства структурно-функциональных особенностей выделенных фрагментов, которые отражаются на их орграфах. Во втором классе семантической корреляции фрагментов ФСС сопоставляются лишь итоговые семантические значения фрагментов, формируемые в «верхних» функциональных объектах орграфов. Каждый из рассмотренных классов семантической корреляции может применяться в процедурах верификации большинства известных типов ФСС.

Библиографический список

[Беляев, 2013] Беляев, М. Г. Аппроксимация многомерных зависимостей по структурированным выборкам/ М.Г. Беляев// Искусственный интеллект и принятие решений. - 2013. - №3. - С.24-39.

[Болотова, 2012] Болотова, Л.С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях: учебник/Л. С. Болотова// ФГБОУ ВПО РГУИТ; ФГАУ ГНИИ ИТТ «Информика». – М.: Финансы и статистика, 2012.

[Вагин и др., 2008] Достоверный и правдоподобный вывод в интеллектуальных системах / Вагин В.Н. [и др.]; – М. : ФИЗМАТЛИТ, 2008.

[Венда, 1990] Венда, В.Ф. Системы гибридного интеллекта: Эволюция, психология, информатика / Венда В.Ф. [и др.]; – М. : МАШИНОСТРОЕНИЕ, 1990.

[Касьянов, 2003] Касьянов, В.Н. Графы в программировании: обработка, визуализация и применение/ В. Н. Касьянов, В. А. Евстигнеев // СПб.: BHV, 2003. – 1104 с.

[Кузнецов О.П., 2014] Кузнецов, О. П. О концептуальной семантике/ О.П. Кузнецов // Искусственный интеллект и принятие решений, 2014, № 3, С. 32-39.

[Хорошевский, 2013] Хорошевский, В.Ф. Семантическая интерпретация паттернов данных на основе структурного подхода / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2013. - № 2. - С.3-13.

ABOUT TWO CLASSES OF SEMANTIC CORRELATION WATCHED ON THE FUNCTIONAL SEMANTIC NETWORKS

Savkin L.V.

Public joint-stock company «Radiofizika»,

Moscow, Russia

android4.1@mail.ru

In operation two semantic correlation classes for fragmentary verification procedures using in the functional semantic networks are considered. As the first class structurally functional correlation based on the analysis of a level likeness digraphs describing in selected network fragments appears. As the second class referential correlation based on comparison of the upper (total) semantic values of verifiable fragments appears.

Keywords: functional semantic network; verification; class; correlation; the structurally functional; referential.

Introduction

One of fundamental and complex problems in the course of creation effective functional semantic networks (FSN) consist in creation of the high-quality formal device FSN description. It is necessary to get rid of inconsistency of the description objects and predicate ratios in between. With growth of general complexity FSN there is a local functional collisions problem registered in the course of interaction objects. They can arise both at one level of network hierarchy and on several.

Main objective of operation is reviewing to two classes semantic correlation in fragments of the FSN connected to methods of a choice correlative signs compared fragments: structurally functional correlation class and referential correlation class.

Main Part

The general semantic correlation of fragments FSN is understood as a level of likeness their total or internal semantic values created by means of a set function objects forming the finished FSN fragments.

We will understand a level of likeness digraphs which are completely describing structural and functional features of fragments FSN as a structurally functional semantic correlation of fragments.

All subclasses of structurally functional correlation are defined by a choice of a specific method comparing digraphs.

By direct comparison of digraphs it is important to consider their inner pattern. It is reflected in a level of likeness verifiable fragments both from the structural point of view and from the function point of sight. In deep analysis of a level likeness it is possible to carry out data digraphs analysis on existence of morphisms. In most cases will be cost research digraphs on connectivity matrixes and incidence matrixes.

In certain cases it is possible to manage a simple quantitative assessment of the elements forming digraphs fragments: summary number of the function objects and summary number of predicate arcs.

We will understand a level of semantic likeness of fragments of FSN as referential correlation of fragments in general which final values are created on outputs in upper function objects digraphs fragments.

The class FSN fragments referential correlation doesn't take structurally functional features of verifiable fragments into account.

The mains to features of referential correlation are:

- The analysis statuses upper semantic function objects creating aggregate values FSN fragments;
- Absence of need about knowledge to an inner verifiable functional fragments pattern;
- Implementation simplicity in comparison with a structurally functional correlation class and possibility distribution of this method on multi-level FSN.

Thus, referential correlation procedure consists in scanning of the upper values objects of the digraph. For the majority of multi-level semantic networks types it can find broad application

Conclusion

In operation two classes of semantic correlation which can be used in case fragmentary verification of the different types functional semantic networks considered. Each of the considered classes of semantic correlation can be applied in procedures of verification majority of known FSN types. Other implicit classes of FSN semantic correlation are now researched



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РОССИЯ В КОНТЕКСТЕ МИРОВЫХ ЦЕНТРОВ КОМПЕТЕНЦИЙ И ПРЕВОСХОДСТВА

Хорошевский В.Ф.^{*}, Ефименко И.В.^{**}

^{}Вычислительный центр им. А.А. Дородницына ФИЦ ИУ РАН, г. Москва, Россия*

khor@ccas.ru

*^{**}Факультет гуманитарных наук НИУ ВШЭ, г. Москва, Россия*

*^{**}SemanticHub, г. Москва, Россия*

iefimenko@hse.ru

В работе представлен наукометрический анализ ландшафта исследований и разработок, выполняемых российскими специалистами в соавторстве со специалистами из других стран, с использованием семантических технологий. Целью исследования является идентификация российских и мировых центров компетенций и превосходства в различных предметных областях на основе специализированных инструментов, разработанных и реализованных авторами.

Ключевые слова: наукометрия; центры превосходства и компетенций; семантические технологии; извлечение информации из текстов

Введение

Задача выявления центров компетенций превосходства в прорывных научно-технологических направлениях на основе мониторинга разнородных информационных ресурсов представляет интерес как с научно-исследовательской, так и с прикладной точки зрения [Акоев и др., 2014]. При этом классический подход к решению этой задачи, основанный на использовании статистических методов библиометрического анализа наиболее значимых публикаций, как показывает целый ряд последних исследований, целесообразно развить за счет использования семантических технологий [Boyack et al., 2013; Efimenko et al., 2014; Хорошевский и др., 2015].

Актуальность и практическая значимость соответствующих исследований определяется, с одной стороны, потребностью любого государства в достоверной информации для определения своей научно-технической политики, а с другой – желанием бизнеса знать, инвестиции в какие направления и коллективы наиболее перспективны.

Особую роль результаты выявления центров компетенций и превосходства в прорывных научно-технологических направлениях играют для позиционирования любой страны в современном

постиндустриальном обществе.

С учетом вышесказанного, в настоящей работе представлен наукометрический анализ ландшафта исследований и разработок, выполняемых российскими специалистами в соавторстве со специалистами из других стран, с использованием семантических технологий, что, по нашему мнению, может обеспечить идентификацию российских и мировых центров компетенций и превосходства в различных предметных областях.

Дополнительное исследование было проведено для выявления научно-технических направлений, где Российская Федерация занимает лидирующие позиции, а также для идентификации стран-партнеров России по исследованиям и разработкам на основе сведений о соавторстве.

1. Методы, средства и данные для наукометрии ландшафтов исследований и разработок

1.1. Предварительные замечания

Как известно, современные библиометрические базы данных (Web of Science, Scopus, Google Scholar и др.) предоставляют пользователям не только поисковые функции, но и целый ряд полезных аналитических инструментов, позволяющих изучать развитие научных направлений, приоритеты стран в

области исследований и разработок, интересы отдельных научно-исследовательских организаций, авторов и многое другое. При этом важную роль в данной области играют не только общие, но и специализированные инструменты, впечатляющий обзор которых дан в работе [Coboetal., 2011].

Вместе с тем, практически все инструменты характеризуются рядом ограничений, связанных как с особенностями самих библиометрических баз, так и с объективной организацией научно-технической информации.

Так, например, в широко распространенных инструментах библиометрии до сих пор до конца не решена проблема объединения синонимов даже для таких простых случаев, как наименования государств (Russia и RussianFederation; China и Peoples Republic China; USA и US и т.п.) и, что существенно сложнее, наименований организаций (Moscow State University и Lomonosov Moscow State University, а также MSU и т.п.).

Другой известной проблемой является сложность интерпретации содержания научных работ на основе ключевых слов автора, на использовании которых, прежде всего, основан содержательный анализ исследований и разработок в рамках библиометрии. Это связано, в частности, с тем, что во многих случаях авторы используют слишком общие ключевые слова. Возможным решением является семантический анализ аннотаций и, при наличии доступа, полных текстов научных публикаций. Необходимость использования полнотекстового анализа научно-технической информации отмечается в настоящий момент многими ведущими исследователями в данной области [Uphametal., 2010; Wangetal., 2010; Li, etal., 2011; Borneretal., 2012; Boyacketal., 2013; Efimenkoetal., 2014; Кулинич, 2011; Хорошевский и др., 2015], а развитие семантических технологий для нужд библиометрии и наукометрии, по-видимому, уже можно считать важным зарождающимся трендом в мире.

С учетом вышесказанного, целью настоящей работы является разработка методологии и интеллектуального инструментария для выявления центров компетенций и превосходства с применением полнотекстового анализа, а также проверка предлагаемых решений на статистически значимом объеме данных из библиометрической БД Web of Science.

1.2. Методология исследования

В основу решения задачи выявления центров компетенций и превосходства были положены следующие гипотезы:

- для позиционирования специалистов и/или организаций в научно-технических сообществах важнейшим информационным каналом являются публикации полученных результатов;

- тематическое картирование исследований и разработок по публикациям может быть базисом для выявления прорывных направлений;
- цитирование опубликованных работ может использоваться в качестве одного из критериев оценки значимости результатов;
- соавторство в публикуемых работах может быть свидетельством признания компетенций членов авторских коллективов;
- аффилиации авторов с организациями могут использоваться для позиционирования последних в качестве центров компетенций и превосходства;
- геоландшафты авторских коллективов дают представление не только о кооперации в науке и технике, но и о научно-технической политике государств;
- факты финансирования исследований и разработок разными фондами (государственными и/или коммерческими) могут использоваться для выявления центров компетенций и превосходства.

Следует сразу отметить, что перечисленными выше гипотезами не исчерпывается перечень индикаторов, которые могут быть положены в основу разработки полномасштабных моделей центров компетенций и превосходства. Вместе с тем, по нашему мнению, для проверки именно этих гипотез библиографических баз данных имеется достаточно информации. Поэтому в настоящем исследовании именно они и положены в основу методологии выявления центров компетенций и превосходства.

1.3. Платформа наукометрии Semantic Hub

1.3.1. Архитектура платформы

Для интеллектуального наукометрического анализа корпусов текстов была разработана платформа Semantic Hub, общая архитектура которой представлена на Рис. 1.

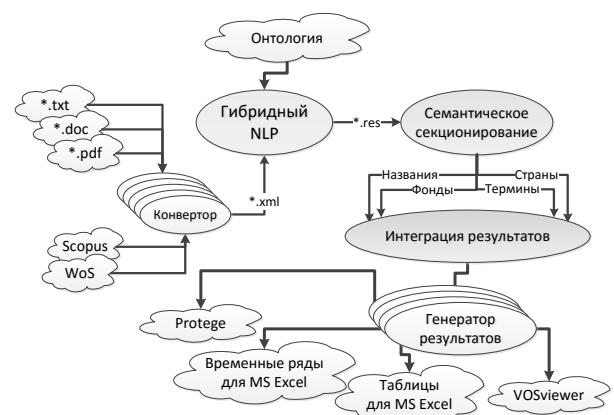


Рисунок 1 – Общая архитектура платформы Semantic Hub

Компоненты основных стадий этой платформы кратко обсуждаются ниже.

1.3.2. Стадия препроцессинга

Для эффективной обработки корпусов текстов в рамках платформы Semantic Hub используется

несколько конверторов, реализованных на языке Java. Результатом работы этих конверторов является XML-представление исходных данных в формате, описанном в работе [Хорошевский и др., 2015]. Для целей настоящего исследования использовался конвертор WoS->XML.

1.3.3. Стадия извлечения информации из текстов

Ядром платформы Semantic Hub является гибридный лингвистический процессор [Efimenkoetal., 2016], реализованный авторами в среде GATE, которая была разработана в Шеффилдском университете Великобритании¹. Спецификой этого процессора является возможность обработки различных семантически значимых фрагментов XML-представления с использованием лингвистических и статистических методов, а также интеграция результатов обработки отдельных текстов в результат обработки коллекции документов и множества коллекций документов.

Ключевую роль в гибридном лингвистическом процессоре играют модули извлечения из текстов терминов (в т.ч., многословных), поскольку именно они определяют качество дальнейшего наукометрического анализа и саму возможность семантически значимой интерпретации результатов. При этом следует отметить, что методы на основе извлечения биграмм и/или триграмм плохо работают в данном случае, поскольку научно-технические термины часто характеризуются существенно более сложными языковыми конструкциями, чем те, которые выделяются на уровне биграмм и/или триграмм. Использование обычных NP-чанкеров также не может рассматриваться как решение проблемы, поскольку, как правило, они не используют знаний о предметной области и уже в силу этого выделяют значительное число нерелевантных именных групп (например, таких как «thispaper» или «asanexample», «эта статья», «проведенный эксперимент» и др.).

Алгоритмы, реализованные в рамках гибридного лингвистического процессора платформы Semantic Hub, базируются на синтаксическом и семантическом анализе ЕЯ-текста, используют 100+ правил-шаблонов и обеспечивают извлечение из текстов сложных именных групп, например, таких как «high-angleannulardark-fieldscanningtransmissionelectronmicroscopy (haadf-stem)», «methanol-tolerantoxxygenreductionreactionofpolymerelectrolytefuelcell», а также именных групп с формулами и аббревиатурами. Дополнительно к лингвистическим правилам в данном процессоре задействованы статистические алгоритмы TF-IDF ранжирования выделенных терминов и отсекающие статистически незначимых терминов по заданному порогу. Кроме того, в гибридном лингвистическом процессоре платформы Semantic Hub используются специальные методы формирования словарей стоп-

слов и выражений на базе концепции «черного ящика» [Efimenkoetal., 2014].

Реализованные алгоритмы обеспечивают формирование значительно более «чистых» списков терминов, извлекаемых из научно-технических текстов. В рамках данной работы извлекались термины 3-х типов: многословные термины (с включением аббревиатур), собственно аббревиатуры формулы.

1.3.4. Стадия генерации результатов

На выходе стадии извлечения информации из текстов формируются множества специальных аннотаций, которые поступают на вход модулей генерации выходных представлений. Для целей настоящего исследования использовались специализированные генераторы статистических данных, полученных в процессе обработки ЕЯ-текстов, таблицы временных рядов, матрицы колокаций, а также специальные матрицы для визуализации результатов в инструментарии VOSviewer[Ecketal., 2010]. Все генераторы реализованы на языке Java.

Таким образом, в рамках платформы Semantic Hub реализован полный цикл получения результатов для дальнейшего наукометрического анализа с использованием внешних аналитических инструментов.

1.4. Инструменты визуализации результатов

Для интерпретации результатов, полученных автоматически, безусловно, должны привлекаться экспертные сообщества. При этом, как правило, используются достаточно сложные аналитические инструменты. Поэтому до начала обсуждения полученных результатов, по нашему мнению, целесообразно хотя бы кратко обсудить те методы и средства визуализации, которые лежат в основе использованных в настоящем исследовании аналитических инструментов.

1.4.1. Гистограммы

Для визуализации распределений терминов и стран в настоящем исследовании использовались гистограммы, которые строились с использованием соответствующих функционалов MSExcel. Возможности этих функционалов хорошо известны и потому в данной работе не обсуждаются.

1.4.2. Временные ряды

Построение временных рядов терминов и стран в рамках настоящего исследования осуществлялась на результатах генерации нормированных распределений терминов и/или стран по отдельным временным периодам. При этом для собственно визуализации использовался MSExcel, соответствующие функционалы которого также хорошо известны и потому в данной работе не обсуждаются.

¹ GATE Official site: <https://gate.ac.uk/>

1.4.3. Ландшафты

В настоящем исследовании для визуализации ландшафтов использовались соответствующие задаче функционалы MSExcel, а также средства системы VOSviewer².

Как отмечалось выше, функционалы MSExcel хорошо известны и в дополнительных пояснениях, по-видимому, не нуждаются. Иная ситуация с использованием системы VOS viewer, в которой реализованы достаточно сложные алгоритмы кластеризации и визуализации. Поэтому ниже остановимся кратко на функционалах этой системы.

Как известно, одним из основных инструментов визуализации результатов в системе VOSviewer являются библиометрические карты [Noyons, 2004; Borneretal., 2012]. Для построения таких карт в качестве основных выступают этапы

- Построения матриц колокаций концептов,
- вычисления коэффициентов «силы связи» концептов,
- позиционирования концептов на основе их «силы связи» в пространстве меньшей размерности, а также
- собственно

визуализации в пространстве меньшей размерности в виде тепловых карт/или карт распределения плотности концептов.

При этом элементами матриц колокаций являются частоты совместной встречаемости концептов в разных документах. Так, например, для матрицы колокаций геоимен: если в авторских коллективах 13 статей одновременно присутствуют специалисты из России (R) и Германии (G), в 21 статье – одновременно специалисты из Франции (F) и Голландии (N), а специалистов из Бельгии (B) и Индии (I), одновременно присутствующих в авторских коллективах, нет $N_{RG}=13; N_{FN}=21; N_{BI}=0$.

Вычисление коэффициентов «силы связи» между концептами базируется на нормализации данных из матрицы колокаций в соответствии со следующей формулой:

$$c_{ij} = \frac{m c_{ij}}{c_{ii} c_{jj}} \quad \text{для } i \neq j. \quad (1)$$

Где c_{ij} соответствует количеству документов, в которых концепты i и j встречаются вместе, c_{ii} соответствует количеству документов, в которых встречается концепт i , m фиксирует общее число документов. Такой подход, по мнению авторов VOSviewer, дает лучшие результаты по сравнению, например, с косинусной мерой или мерой Джакарда. Для примера, при условиях, указанных выше для коэффициентов матрицы колокаций геоимен и общего числа документов в корпусе $m=100; N_{RR}=100; N_{GG}=50; N_{FF}=30; N_{NN}=20; N_{BB}=10; N_{II}=3$ значения $C_{RG}=2.6; C_{FN}=3.5; C_{BI}=0$.

Позиционирование концептов в тепловых картах/или в картах распределения плотности концептов осуществляется на основе специального метода учета «силы связи» между концептами [Ecketal., 2009; Ecketal., 2014].

Суть VOS-метода заключается в следующем. Необходимо расположить n концептов в 2-мерном пространстве так, чтобы расстояние между любой парой концептов i, j отражало их «силу связи» c_{ij} максимально аккуратно. При этом концепты с большими значениями «силы связи» должны располагаться близко, а концепты с малыми значениями «силы связи» – далеко друг от друга. Идея VOS-метода состоит в минимизации взвешенной суммы квадратов евклидовых расстояний между всеми парами концептов. Для исключения случаев, в которых все концепты «стянутся» в одну точку, вводится ограничение на сумму всех расстояний, которая должна быть равна заданной константе. В математическом смысле производится минимизация функции

$$E(x_1, \dots, x_n) = \sum_{i < j} c_{ij} \|x_i - x_j\|^2 \quad (2)$$

Где вектор $x_i = (x_{i1}, x_{i2})$ фиксирует расположение концепта i в 2-х мерном пространстве, $\| \cdot \|$ задает евклидову норму, а минимизация функции осуществляется с учетом следующего ограничения

$$\frac{1}{n(n-1)} \sum_{i < j} \|x_i - x_j\| = 1. \quad (3)$$

При этом численное решение задачи оптимизации функции с учетом ограничений осуществляется за 2 шага. Сначала задача оптимизации с ограничениями трансформируется в задачу оптимизации без ограничений, которая решается с помощью мажоритарного алгоритма [Vorgetal., 2005]. Для исключения эффекта получения локального минимума мажоритарный алгоритм запускается 10 раз со случайными начальными данными.

Собственно визуализация результатов выполнения предыдущих этапов происходит в виде тепловых карт и/или карт распределения плотности концептов.

Визуализация тепловых карт осуществляется с помощью Java-апплета, где место концепта фиксируется его наименованием-меткой. При этом важность концепта специфицируется размером шрифта метки, который связан с числом документов где концепт присутствует. Распределение «интереса» между концептами специфицируется цветом кластера, в которых эти концепты находятся.

Предполагается, что кластеризация концептов уже проведена на базе использования рассмотренных выше метрик. Пусть, для

²<http://www.vosviewer.com/relatedsoftware/>

определенности, в кластер 1 попали документы с авторами из России и Германии (всего в кластере 150 документов), в кластер 2 – из Франции и Голландии (всего в кластере 50 документов), а в кластер 3 – из Бельгии и Индии (всего в кластере 30 документов). Тогда область концепта i «окрашивается» в палитре RGB с помощью следующих формул:

$$R_i = \frac{p_i^1}{p_i^1 + p_i^2 + p_i^3} 180 + 75, \quad (4)$$

$$G_i = \frac{p_i^2}{p_i^1 + p_i^2 + p_i^3} 180 + 75, \quad (5)$$

$$B_i = \frac{p_i^3}{p_i^1 + p_i^2 + p_i^3} 180 + 75, \quad (6)$$

где p_i^1 задает отношение числа документов из кластера 1, в которых концепт i присутствует, к числу документов из кластера 1, p_i^2 – отношение числа документов из кластера 2, в которых концепт i присутствует, к числу документов из кластера 2, а p_i^3 – отношение числа документов из кластера 3, в которых концепт i присутствует к числу документов из кластера 3.

Из наших предположений и формул (4-6) нетрудно получить цвета концептов. Так, например, $RGB_R=(255, 75, 75)$ и $RGB_G=(255, 75, 75)$, что понятно, поскольку авторы из этих стран находятся в одном кластере. Но $RGB_F=(75, 255, 75)$ и $RGB_N=(75, 255, 75)$, поскольку авторы из этих стран находятся в другом кластере.

Известным недостатком тепловых карт является частое перекрытие наименований концептов, что затрудняет понимание общей структуры карты концептов. Поэтому в систему VOSviewer включена опция визуализации результатов в виде карт распределения плотности концептов, где представлены не все концепты, а только те из них, которые встречаются часто, причем для индикации концептов с одинаковой плотностью в разных областях карты используются одинаковые цвета. При этом плотность концепта зависит от числа его «соседей» и от их важности следующим образом: увеличение числа «соседей» концепта и уменьшение расстояния между ними приводит к увеличению плотности, причем увеличение числа документов, в которых присутствует концепт, также увеличивает его плотность. Для вычисления значений плотности концептов авторы системы VOSviewer используют функциональные возможности пакета MATLAB. Для цветового ранжирования концептов значения их плотностей упорядочиваются на шкале «голубой-красный» от меньших значений (голубой цвет) к большим (красный цвет).

1.5. Формирование корпуса для проведения исследования

Для решения поставленных выше задач

авторами был проведен библиометрический анализ 255 000+ библиографических описаний научных публикаций с российским авторством (соавторством), представленных в Web of Science за 2009-2015 гг., на базе доступных инструментов из этой БД, а также углубленный семантический анализ 7 000 аннотаций наиболее цитируемых публикаций из этого корпуса с использованием собственного инструментария семантического наукометрического анализа [Efimenko et al., 2014; [Хорошевский и др., 2015] и специализированного инструментария визуализации результатов VOSviewer, разработанного в Лейденском университете. При этом российскими считались те авторы, которые явно указали свою аффилиацию с организациями, расположенными в России.

Для формирования корпуса в качестве первого шага был сформирован запрос к БД Web of Science, позволяющий отобрать все публикации, где среди авторов присутствуют российские соавторы (или единственный автор аффилирован с российской организацией), за период 2009-2015 гг. Число публикаций по годам в результатах запроса представлено в Таблице 1.

Таблица 1 – Публикации с российскими авторами по данным Web of Science, 2009-2015 гг. (по состоянию на 25 января 2016 г.)

№	Год	Число публикаций
1.	2009	34 370
2.	2010	33 415
3.	2011	34 640
4.	2012	34 689
5.	2013	37 268
6.	2014	41 032
7.	2015	40 379

Необходимыми элементами моделей центров компетенций и превосходства являются индикаторы о результатах, влияющих на развитие последующих исследований и разработок, а также признание научным сообществом, в т.ч. международным. Таким образом, было принято решение выполнять детализированный анализ для публикаций с самыми высокими показателями цитируемости. Поскольку значения числа публикаций по годам близки для всех лет из рассматриваемого периода, для каждого года было выбрано по 1 000 самых цитируемых публикаций. Показатели цитируемости по годам представлены в Таблице 2.

Таблица 2 – Статистика цитирований для Топ-1000 публикаций с российскими авторами по данным Web of Science, 2009-2015 гг. (по состоянию на 25 января 2016 г.)

№	Год	Число цитирований для наиболее цитируемой публикации (1 место в рейтинге)	Число цитирований для последней публикации из Топ-1000 (1 000 место в рейтинге)
1.	2009	1 887	36
2.	2010	3 979	32
3.	2011	1 650	28

4.	2012	4 368	23
5.	2013	508	17
6.	2014	2 579	10
7.	2015	161	3

Таким образом, был сформирован корпус текстов для семантического анализа объемом 7 000 аннотаций.

2. Результаты и обсуждение

Как отмечалось выше, в рамках настоящего исследования был проведен библиометрический анализ достаточно полного корпуса публикаций с российским соавторством с помощью инструментов Web of Science и семантический наукометрический анализ Топ-1000 публикаций из этого корпуса по каждому из 2009-2015 гг. с использованием платформы SemanticHub и внешних аналитических инструментов.

Ниже представлены полученные результаты и приводится их обсуждение.

2.1. Результаты библиометрии

Подготовка статистических данных для анализа рассмотренного выше корпуса осуществлялась с использованием соответствующих инструментов Web of Science, а для их интерпретации использовались подходящие функционалы MSEXcel.

На Рис. 2-4 выборочно представлены сведения о топ-10 странах-партнерах, полученные на основе данных о соавторстве на всем множестве публикаций за каждый год (анализ был выполнен для всех стран мира, по доступным сведениям, не менее 100 стран для каждого года). Результаты анализа показывают, что ключевые партнеры (топ-5) не меняются с течением времени, при этом в целом по рейтингу наблюдается рост значимости партнерства с отдельными странами. В частности, за рассматриваемый период изменилось место Китая в рейтинге стран-партнеров, причем речь идет не о случайных выбросах, а о тренде: 14 место в 2009 году, 9 место в 2010 и 2011 годах, 7 место в 2012- 2014 годах, 6 место в 2015 году.

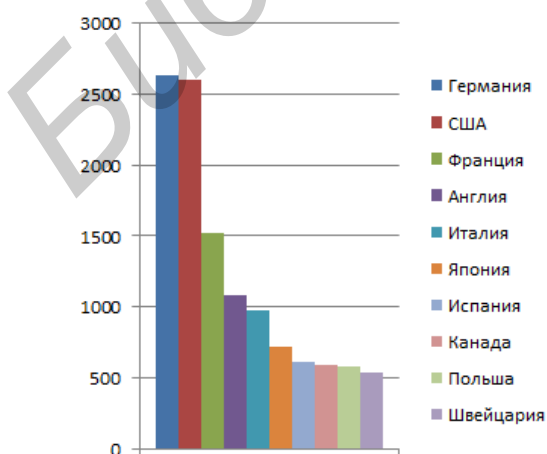


Рисунок2 – Топ-10 стран-партнеров РФ на основе сведений о соавторстве, по числу публикаций, 2009 год

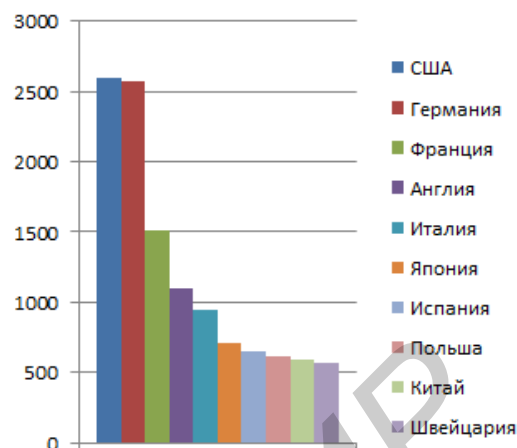


Рисунок3 – Топ-10 стран-партнеров РФ на основе сведений о соавторстве, по числу публикаций, 2010 год

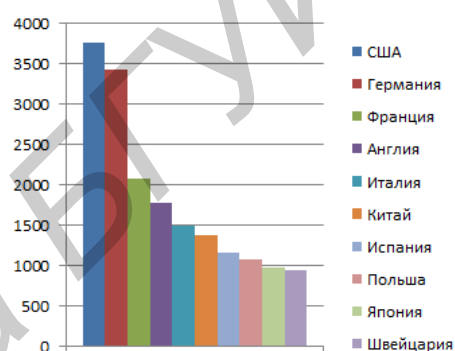


Рисунок4 – Топ-10 стран-партнеров РФ на основе сведений о соавторстве, по числу публикаций, 2015 год

Отдельно был выполнен анализ наиболее результативных коллабораций – стран-партнеров по наиболее цитируемым (топ-1000) публикациям. Пример результатов для 2009 года представлен на Рис. 5. Результаты анализа показывают, что наиболее результативные с точки зрения показателей цитируемости партнерства, в большинстве случаев, совпадают с партнерствами, наиболее продуктивными по числу публикаций.

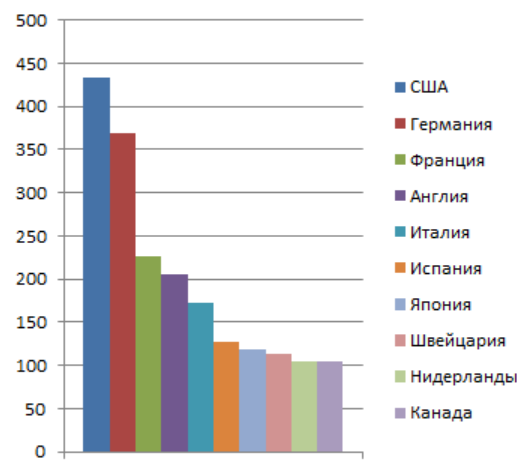


Рисунок5 – Топ-10 стран-партнеров РФ на основе сведений о соавторстве, по топ-1000 наиболее цитируемых публикаций, 2009 год

Дополнительно был выполнен анализ научных направлений, которым соответствуют публикации российских авторов (в т.ч. с зарубежными соавторами). Для этого были использованы сведения о категориях WebofScience (направления научных исследований по классификатору, используемому данной платформой), где представлены российские публикации, как на всем объеме публикаций, так и для наиболее цитируемых работ, для каждого года отдельно и за период 2009-2015 гг. в целом.

Пример результата для периода в целом (топ-10 категорий WebofScience по числу публикаций с российскими авторами) представлен на Рис. 6.

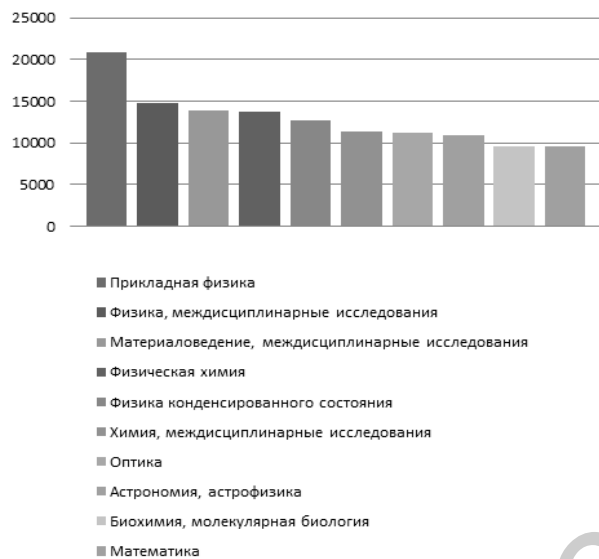


Рисунок 6 – Топ-10 категорий Web of Science по числу публикаций с российскими авторами, 2009-2015 гг.

Результаты анализа показывают, что классические инструменты библиометрии дают общее представление о направлениях исследований и разработок, которые могут рассматриваться как научно-технологические области, где позиционируются центры компетенций и превосходства. Однако речь в данном случае идет только о верхнеуровневом анализе. Детальные сведения о ландшафте исследований и разработок не могут быть получены с помощью инструментов библиометрических баз данных без использования интеллектуального семантического анализа.

2.2. Семантический наукометрический анализ корпуса самых цитируемых публикаций

В результате работы модулей и компонент платформы SemanticHub из текстов научных публикаций и/или их аннотаций в рамках настоящего исследования извлекались значимые научно-технологические концепты в форме именных групп (однословных и, чаще, многословных терминов), позволяющие сделать вывод о содержании публикации и сути выполненной научной работы. Научно-технологические концепты извлекались также из названий публикаций.

Дополнительно к научно-технологическим концептам из аффилиаций авторов извлекались наименования государств, а из благодарностей – наименования фондов, финансирующих исследования и разработки.

2.2.1. Гистограммы

Как представляется, для дальнейшего анализа результатов наибольший интерес представляют распределения частот выделенных с помощью гибридного лингвистического процессора научно-технологических концептов, аффилированных с авторами стран и фондов, финансирующих исследования и разработки. Общие гистограммы распределения этих индикаторов за 2009-2015 гг. представлены на Рис. 7-9.

Научно-технологических терминов выделено 101397. При этом на уровне отсечения 1 их 19173, а на уровне отсечения 10 – всего 1414. Ниже, на Рис. 7, представлено распределение частот встречаемости Топ-100 терминов.

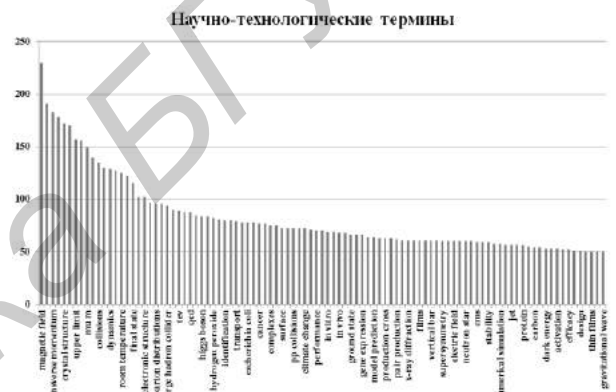


Рисунок 7 – Распределение Топ-100 терминов (2009-2015 гг.)

Всего выделено 166 стран, представленных в авторских коллективах. 98 стран представлены не менее чем в 10 случаях, 61 страна – не менее чем в 100 случаях и 40 стран – не менее чем в 400 случаях. На Рис. 8 приведенное распределение Топ-40 стран из авторских коллективов.

Страны в авторских коллективах

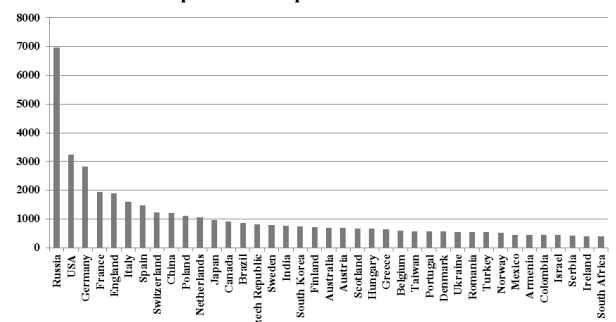


Рисунок 8 – Распределение Топ-40 стран (2009-2015 гг.)

Как показал анализ результатов, гипотеза о том, что для наукометрического анализа источников финансирования исследований и разработок можно использовать значения соответствующего тэга записей WebofScience, оказалась неверной. Такая ситуация, по нашему мнению, связана с тем, что в словарях фондов в БД WebofScience представлены

стандартизованные наименования, а авторы специфицируют одни и те же фонды в разделах благодарностях самыми разными способами. И более того, выходной формат тэга фондов в этой БД неоднозначен. Поэтому результаты обработки тэга фондов в данном исследовании были вручную отфильтрованы, а на Рис. 9 представлены результаты для Топ-100 финансирующих организаций.

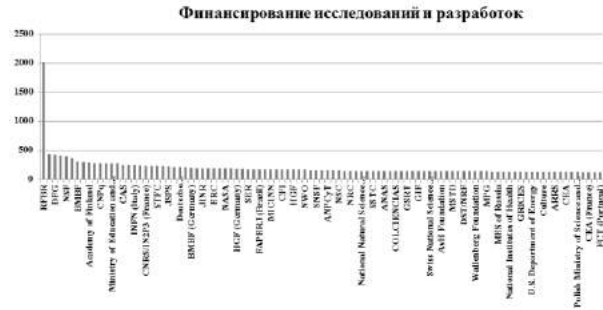


Рисунок 9 – Распределение Топ-100 фондов (2009-2015 гг.)

Анализ представленных выше гистограмм частот показывает достаточно четкое соответствие закону Ципфа.

2.2.2. Временные ряды

В настоящей работе для анализа динамики исследований и разработок строились нормализованные временные ряды терминов и других индикаторов. Для примера на Рис. 10-11 представлены полученные результаты.

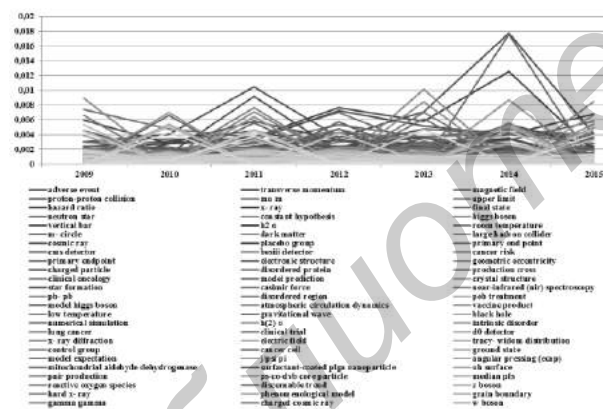


Рисунок 10 – Временные ряды Топ-100 терминов (2009-2015 гг.)

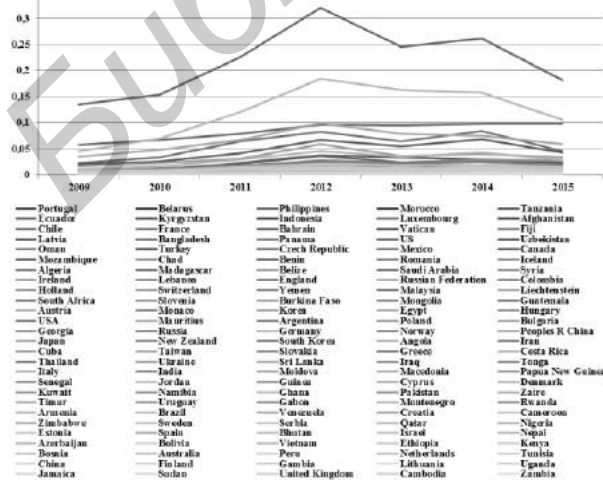


Рисунок 11 – Временные ряды Топ-100 стран (2009-2015 гг.)

2.2.3. Ландшафты

Как известно, для аналитиков наибольший интерес представляют карты научных направлений. Поэтому в данной работе строились тепловые карты и карты распределения плотности научно-технических терминов, стран авторских коллективов и фондов, финансирующих исследования и разработки. Для примера, на Рис. 12 представленная тепловая карта стран из авторских коллективов, а на Рис. 13 – спектр сетей кластеров научно-технических терминов. Детальное обсуждение полученных результатов представлено ниже.

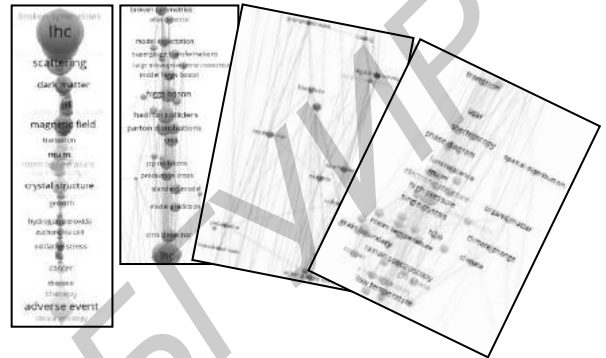


Рисунок 12 – Тепловая карта стран из авторских коллективов по странам (2009-2015 гг.)



Рисунок 13 – Сети научно-технических терминов (2009-2015 гг.)

Анализ результатов показывает, что Топ-5 стран (по данным Web of Science) это Россия-США-Германия-Франция-Англия. Следующую по рангу «весов» системы VOSviewer группу составляют Италия и Испания, а замыкают «чертову дюжину» самых активных стран Швейцария, Китай, Польша, Голландия, Япония и Канада.

Понятно, что Россия входит в Топ-5, поскольку в исходном корпусе изначально присутствовали только те статьи, в которых были (co)авторы из России. Что же касается остальных стран, их присутствие в лидирующих группах, по оценке экспертов, вполне предсказуемо.

Для выявления центров компетенции и превосходства интересе не просто анализ научно-технической активности по странам, но и выявление тех тематик, где страны-лидеры концентрируют свои исследования и разработки. С учетом этого, на Рис. 14 фрагмент карты наиболее частотных терминов из статей Топ-5 стран, а на Рис. 15 –

тепловая карта Топ-10 научно-технических терминов из статей Топ-5 стран.

Последняя наиболее интересна для понимания того, чем занимаются страны-лидеры. При этом важно, что теми же вопросами занимается и Россия, причем ее уровень компетентности в этих вопросах высок, поскольку в противном случае российских исследователей не приглашали бы в авторские коллективы самых цитируемых статей, опубликованных в высоко рейтинговых научных изданиях.

Анализ данных, представленных на Рис. 15, показывает, что страны-лидеры концентрируются (по данным Web of Science), в основном, в области наиболее значимых областей физики.

По нашему мнению, результаты анализа демонстрируют не только научно-техническую важность новых направлений физики, но и давнюю ориентацию Thomson Reuters на индексирование работ в области ядерной физики, космоса и некоторых других направлениях для мониторинга

получаемых в других странах научно-технических решений.

3. Направления дальнейших исследований и разработок

С учетом полученных результатов в качестве направлений дальнейших исследований и разработок целесообразными представляются авторам следующие:

- Расширение корпуса для обработки за счет других библиометрических баз (Scopus, Pub Med, Google Scholar и т.п.), а также за счет тематических архивов научных сообществ.
- Специализация запросов к базам данных библиометрии с целью формирования тематических корпусов для дальнейшей обработки.
- Подключение уже разработанных авторами модулей детализации аффилиаций авторов к гибриднему лингвистическому процессору платформы SemanticHub.

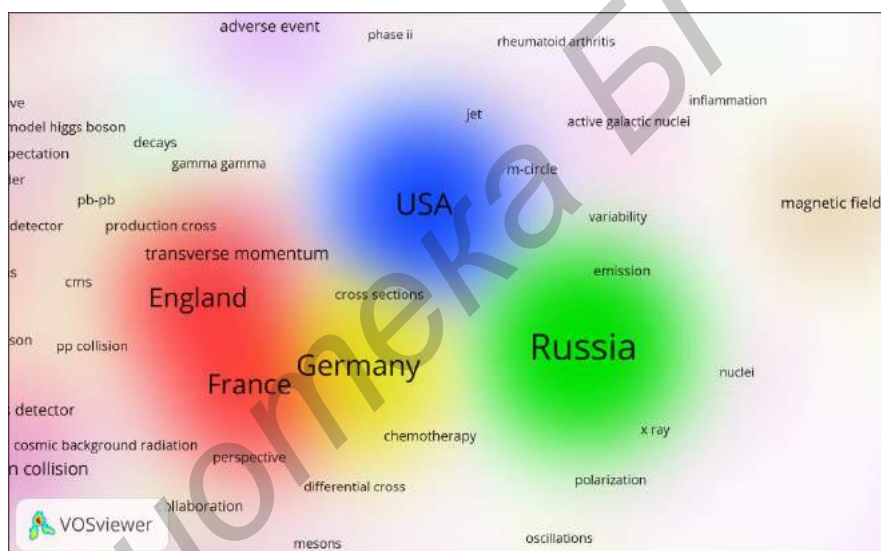


Рисунок 14 – Фрагмент карты наиболее частотных научно-технических терминов из статей Топ-5 стран (2009-2015 гг.)

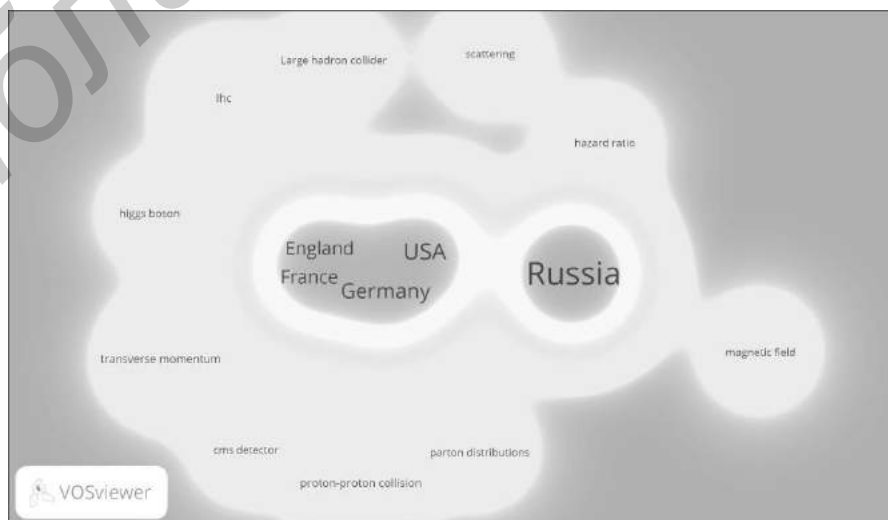


Рисунок 15 – Тепловая карта Топ-10 научно-технических терминов из статей Топ-5 стран (2009-2015 гг.)

- Доработка гибридного лингвистического процессора платформы SemanticHub в части наименований финансирующих исследования и разработки фондов.

- Использование в рамках анализа не только наиболее частотных терминов, но терминов, которые могут представлять «слабые сигналы».

Заключение

В работе представлен наукометрический анализ ландшафта исследований и разработок, выполняемых российскими специалистами в соавторстве со специалистами из других стран, с использованием семантических технологий с целью выявления российских и мировых центров компетенций и превосходства в различных предметных областях на основе специализированных инструментов, разработанных и реализованных авторами.

Как показал проведенный анализ, платформа интеллектуальной наукометрии Semantic Hub может быть использована для решения поставленных задач и дает семантически интерпретируемые результаты.

Работа выполнена при частичной поддержке гранта РФФИ № 15-01-06819 «Исследование и разработка онтологических моделей центров компетенции/превосходства в прорывных научно-технологических направлениях на основе мониторинга разнородных информационных ресурсов», а также Фонда «Центр стратегических разработок» (в рамках разработки Стратегии научно-технологического развития РФ на долгосрочный период).

Библиографический список

[Акоев и др., 2014] Акоев М.А. и др. Руководство по наукометрии: индикаторы развития науки и технологии. – Екатеринбург, Изд-во Урал. ун-та, 2014. – 250с.

[Кулинич, 2011] Кулинич А.А. Компьютерные системы анализа ситуаций и поддержки принятия решений на основе когнитивных карт: подходы и методы. // Проблемы управления, 2011, № 4 С.31-45.

[Хорошевский и др., 2015] Хорошевский В.Ф., Ефименко И.В. Семантическая технология картирования семантических технологий (Наукометрический анализ конференций OSTIS). // Труды V международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». Минск 19-21 февраля 2015 г., - Минск, БГУИР, 2015, с. 43-57.

[Borg et al., 2005] Borg I., Groenen P. J. F. Modern Multidimensional Scaling. Springer, second edition, 2005.

[Borner et al., 2012] Borner, K., Boyack, K. W., Milojevic, S., & Morris, S. (2012). An introduction to modeling science: Basic model types, key definitions, and a general framework for the comparison of process models. / Modeling Science Dynamics (Understanding Complex Systems), Springer-Verlag, p 3-22.

[Boyack et al., 2013] Boyack, K. W., Small, H., & Klavans, R.. (2013). Improving the accuracy of co-citation clustering using full text. Journal of the American Society for Information Science and Technology, 64(9), 1759-1767.

[Cobo et al., 2011] Cobo M.J., López-Herrera A.G., Herrera-Viedma E., and Herrera F. Science Mapping Software Tools: Review, Analysis, and Cooperative Study Among Tools. Journal of the American Society for Information Science and Technology, 62(7):1382–1402, 2011

[Eck et al., 2009] Van Eck, N.J., & Waltman, L. (2009). How to normalize cooccurrence data? An analysis of some well-known similarity measures. Journal of the American Society for Information Science and Technology, 60(8), 1635–1651.

[Eck et al., 2010] Van Eck N.J., Waltman L. Software survey: VOSviewer, a computer program for bibliometric mapping. Scientometrics, 84(2), 523–538.

[Eck et al., 2014] Van Eck N.J., Waltman L. Visualizing bibliometric networks. /In Y. Ding, R. Rousseau, & D. Wolfram (Eds.), Measuring scholarly impact: Methods and practice (pp. 285–320). Springer.

[Efimenko et al., 2014] Efimenko I., Khoroshevsky V. New Technology Trends Watch: an Approach and Case Study. // Proc. of AIMS-2014. Springer-Verlag, 2014. P. 170-177.

[Efimenko et al., 2015] Efimenko I., Khoroshevsky V. Peaks, Slopes, Canyons, Plateaus: Identifying Technology Trends throughout the Life Cycle. // International Journal of Innovation and Technology Management (Special Issue “Bibliometrics and Social Network Analysis Methods for Technology and Innovation Management”, 2015) (In press)

[Efimenko et al., 2016] Efimenko I., Khoroshevsky V., Noyons E. Anticipating Future Pathways of Science, Technologies & Innovations: (Map of Science)² Approach. Chapter in Book: Anticipating Future Innovation Pathways through Large Data Analytics (eds.: T. Daim, A. Porter, D. Chiavetta, O. Saritas), Springer Verlag, 2016 (In press)

[Li, et al., 2011] Li H., Xu F., Uszkoreit H.: TechWatchTool: Innovation and Trend Monitoring. In: Proc. of the International Conference on Recent Advances in Natural Language Processing 2011 RANLP 2011, Tassar, Bulgaria, pp. 660-665 (2011).

[Noyons, 2004] Noyons E. C. M. Science maps within a science policy context. In H. F. Moed, W. Glanzel, and U. Schmoch, editors, Handbook of Quantitative Science and Technology Research, pages 237–255. Kluwer Academic Publishers, 2004.

[Upham et al., 2010] Upham, S. P., & Small, H. (2010). Emerging research fronts in science and technology: Patterns of new knowledge development. Scientometrics, 83(1), 15-38.

[Wang et al., 2010] Wang et al. Identifying technology trends for RD planning using TRIZ and text mining, RD Management, vol. 40, N 5, 2010.

RUSSIA AMONG THE WORLD CENTERS OF EXCELLENCE

Khoroshevsky V.F.* , Efimenko I.V.**

**Dorodnicyn Computing Centre,
Federal Research Center «Computer Science and
Control» RAS, Moscow, Russia*

khorr@ccas.ru

***Faculty of Humanities, NRU HSE, Moscow,
Russia*

Semantic Hub, Moscow, Russia

iefimenko@hse.ru

The paper presents a scientometric analysis of the landscape of R&D carried out by Russian scientists in cooperation with foreign co-authors. A novel approach is based on semantic technologies in Scientometrics. It makes use of tools for intelligent big data analysis which were developed by the authors of the paper within the SemanticHub platform. The research is aimed at identification of Russian and world-level centers of excellence in various R&D domains.

Keywords: Scientometrics; Centers of excellence; Semantic Technologies; Information Extraction



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822: 004.912

ПОСТРОЕНИЕ СЕТЕЙ СОВАВТОРСТВА ПО ДАННЫМ СЕРВИСА GOOGLE SCHOLAR CITATIONS

Ландэ Д.В., Балагура И.В., Андрущенко В.Б.

*Институт проблем регистрации информации НАН Украины,
г. Киев, Украина*

dwlande@gmail.com

balaguraira@mail.ru

valentyna.andrushchenko@gmail.com

В работе приводится алгоритм построения сети соавторства ученых, регулируемой их научными интересами. Сеть соавторства формируется на основе зондирования сервиса GoogleScholarCitations. Показано, что дескрипторы, определяющие тематическую направленность, влияют на размер формируемой сети, а также на динамику ее роста. Показано, что кластеры в сетях соавторства могут рассматриваться как основа для выявления научных школ.

Ключевые слова: сеть соавторства; наукометрия; Google Scholar Citations; предметная область; граф связей; зондирование сетей.

Введение

Развитие Интернета, поисковых систем и социальных сетей повлияло на многие сферы деятельности человека, в том числе и на развитие науки. Трансформация реферативных баз данных, возникновение различных научных платформ для распространения, обмена научной информацией спровоцировало новые подходы ведения научных исследований, формирования научных групп и поиска возможного сотрудничества. Как следствие развития сервисов научной информации, появились новые возможности оценки научной информации и изучение закономерностей научного взаимодействия [Ortega, 2015]. Основным инструментом изучения закономерностей научного сотрудничества являются сети соавторов, с помощью которых можно получить не только наукометрические оценки, но и определять экспертов для решения сложных заданий. Одним из крупных сервисов научной информации является Google Scholar, который позволяет создавать ученым профили, содержащие соответствующую библиографическую информацию, а также осуществлять поиск публикаций со всего мира. Изучению сетей соавторов так же как и сервиса Google Scholar Citation (<http://scholar.google.com/citations>) посвящено большое количество работ, что подтверждает актуальность проводимых исследований [Liu, 2015].

Среди них методы построения сетей соавторов, определения значимых узлов, структуры сети, исследование цитирования в Google Scholar, а так же соответствующих корпусов др. [Brezina, 2012].

Предлагается методика построения сетей соавторства – моделей сотрудничества ученых на основе зондирования наукометрических сетей. Как такая сеть в работе рассматривается сеть понятий, соответствующих тегам сервиса GoogleScholarCitations.

Целью работы является описание теоретических принципов и методологии автоматизированного формирования сетей соавторства, в частности, областях ComplexNetworks и TextMining путем зондирования большой информационной сети. Для достижения этой цели применяется специальный алгоритм сканирования ресурсов сервиса Google Scholar Citations с целью получения репрезентативного набора соавторов как основы (узлов) будущей сети. Под зондированием сетей будем понимать выборку небольшого объема важнейшего содержания из больших сетей, которые по технологическим причинам не подлежат полному сканированию [Lande, 2015].

Очевидно, сеть соавторства может иметь достаточно большие размеры, если ее не ограничивать определенной тематикой, например, задаваемой тематикой первого автора, начиная с которого идет процесс формирования этой сети.

Данный эффект значительно усложняет восприятие сформированной сети и приводит к такому эффекту, как «дрейф тематики». Также имеет место одинаковое написание фамилий и инициалов различных ученых. Для преодоления этих эффектов применяется тематическая фильтрация, т.е. используются дескрипторы, приписываемые авторам наукометрической сети, определяющие их тематическую направленность. Соответствие этим дескрипторам в конечном итоге и определяют размер формируемых сетей соавторства, а так же динамику ее роста. Кроме того, распознавание кластеров в таких сетях может рассматриваться как основа для выявления научных школ, экспертных групп и т.п. [Landeets, 2013].

1. Описание модели

При построении сетей соавторства целесообразно применять модели, уже апробированные на пиринговых сетях (peer-to-peer, P2P – равный с равным), основанных на равноправии участников. Пиринговые сети состоят из узлов, каждый из которых взаимодействует лишь с некоторым подмножеством других узлов, что вполне соответствует сети соавторов.

Для поиска необходимых данных в таких сетях применяется несколько моделей. В модели "широкого первичного поиска" (Breadth First Search, BFS) запрос из некоторого стартового узла адресуется ко всем соседям (ближайшим по некоторым критериям) [Kalogeraki, 2002]. Когда некоторый другой узел получает запрос, выполняется поиск в его локальном индексе и в случае успеха возвращает результат. В противном случае запрос передается по сети далее. В нашем случае (зондирование сети соавторов) запрос передается далее во всех случаях, если не выполняются некоторые условия-ограничения.

Зондирование опорной модельной сети осуществляется по такому алгоритму [Lande, 2015]:

1. Выбирается определенное количество узлов опорной (зондируемой) сети, определяемых как базовые для новой сети, соответствующей результатам зондирования (в простейшем случае выбирается один узел).

2. Для каждого из рассматриваемых узлов опорной сети определяются смежные с ним узлы (соавторы), которые добавляются к создаваемой сети как результаты зондирования. Формируются ребра-связи к этим узлам из исходного узла.

3. От текущего узла опорной сети осуществляется переход к случайно выбранному соседнему узлу формируемой сети (соавтору).

4. Если имеет место "зацикливание" (выбирается узел, к которому уже был осуществлен переход по этому алгоритму) или несоответствие узла некоторому условию-ограничению, происходит переход к другому случайно выбранному узлу формируемой сети. Если таких узлов не осталось, считается, что сеть,

соответствующая результатам зондирования, построена

В качестве условий-ограничений на практике проверяется вхождение в теги соавтора некоторого множества допустимых. В рамках модели просто каждому из узлов с некоторой вероятностью приписывается возможность порождения от него последующих связей.

При моделировании приведенный алгоритм применялся для двух самых распространенных модельных сетей Erdős-Rényi (ER) и Barabási-Albert (BA). Известно, что модель ER [Erdős, 1960] – это случайная сеть, которая строится следующим образом: множество из N изначально не соединенных узлов попарно объединяют с вероятностью p . В результате создается сеть приблизительно с $pN(N-1)/2$ случайно выбранными связями.

Модель BA [Barabási, 1997] – сеть со степенным распределением степеней узлов (так называемых, безмасштабных сетей). Эта модель учитывает принцип преимущественного присоединения, который заключается в том, что чем больше связей имеет узел, тем более вероятно для него создание новых связей со вновь образуемыми узлами.

Следует отметить, что безмасштабными являются наиболее популярные реальные сети, такие как веб-пространство с гиперссылками, социальные сети, сети слов в литературных произведениях, сети протеинов, и т.п. [Newman, 2003] Авторами было показано, что сети соавторства тоже обладают свойством безмасштабности. Исходя из информации о том, что все известные большие сети цитирования, соавторства и т.п. обладают свойством безмасштабности, т.е. в чем-то близки по структуре сети Barabási-Albert, изучались модели, одна из которых базировалась на алгоритме BA. От этой модели принципиально отличаются случайные сети Erdős-Rényi, которые также изучались для сравнения. Сравнение показывает, что связанные области (ветки), соответствующие отдельным понятиям в случае модели ER достаточно длинные, а узлов, по которым следует маршрут зондирования больше, чем в более интересном случае модели BA (рис. 1). В данном случае нам важны именно качественные результаты, вид связанных цепочек, которыми моделируются ветки понятий. Следует отметить, что реальным сетям присущий еще и феномен "клуба богатых" (Rich Clube), который обуславливает более плотную связанность наибольших узлов-соавторов.

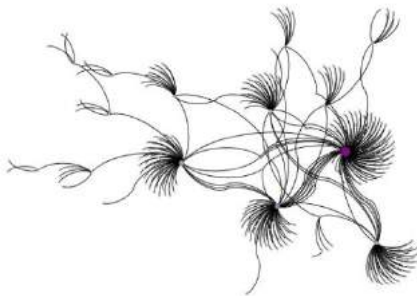


Рисунок 1 -- Пример сети, построенной зондированием сети типа Barabási-Albert

Именно на основании результатов качественного моделирования был сделан вывод о возможности формирования небольших связанных веток соавторов, соответствующих понятиям, интересующим пользователей сервиса GoogleScholarCitations.

2. Зондирование сети Google Scholar Citations

Приведенный алгоритм адаптировался к реальной сети соавторов сервиса Google Scholar Citations следующим образом (рис.2):

1. Выбирается первый автор, с которого начинается зондирование.
2. Экспертным путем определяется небольшой перечень базовых тегов-дескрипторов, соответствующих наиболее важным понятиям.
3. Открывается страница веб-сервиса, соответствующая выбранному автору.
5. К создаваемой сети добавляются все соавторы, содержащиеся на странице выбранного автора. Формируются ребра-связи к этим узлам (соавторам) из исходного узла (автора).
4. Из списка узлов формируемой сети случайным образом тот, на страницу которого планируется перейти для дальнейшего анализа. Этот узел также должен удовлетворять тематике выбранной предметной области (его теги входят в состав дескрипторов, определенных на шаге 2) и не входит в состав тех узлов, к страницам которых уже был осуществлен переход.
5. Если такой узел-автор выбран, то происходит переход к пункту 3.
6. Если такого автора не существует, то считается, что сеть зондирования построена.

В соответствии с приведенным алгоритмом процесс зондирования сети, начиная с определенного узла, прекращается при «зацикливании», т.е. когда в соответствии с

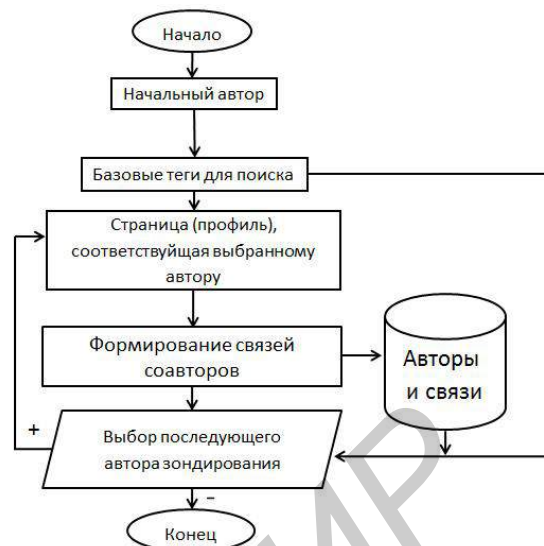


Рисунок 2 -- Алгоритм зондирования сети Google Scholar Citations

алгоритмом происходил переход к уже пройденному узлу, а также при отклонении оставшихся соседних узлов от основной тематики (определяется учетом лексического состава тегов). При этом само «зацикливание» является признаком перехода к следующему базовому автору или завершению процесса зондирования.

3. Примеры построения сетей соавторства

Построения в соответствии с приведенным алгоритмом построена сеть соавторов при достаточно широком списке дескрипторов-ограничений (computer, networks, language, information, complex, text) и ограничении на количество сканируемых узлов в 1000. С помощью программного средства получена визуализация данной сети соавторов (рис. 3).

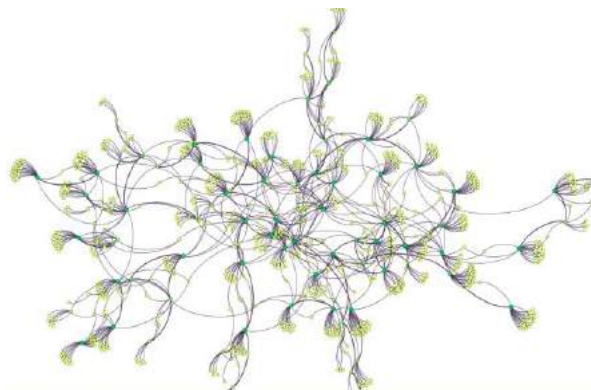


Рисунок 3 -- Фрагмент сети соавторов, построенная с учетом указанных широкой тематики дескрипторов

Как видно по рис. 4. динамика роста сети имеет четкий линейный тренд из-за чересчур широкой предметной области.

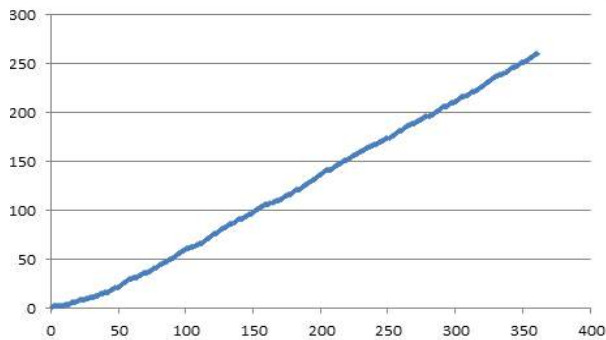


Рисунок 4 -- Динамика роста сети соавторов, построенная с учетом указанных дескрипторов (горизонтальная ось – шаги выбора узлов (авторов), вертикальная – количество узлов)

Для построения сетей соавторство по тематике TextMining экспертным путем были определены базовые теги на английском языке: language processing, text mining, information retrieval, complex networks. На рис. 5 приведен пример сети соавторов, построенной в соответствии с приведенным алгоритмом по указанным базовым дескрипторам-тегам (первый узел зондирования соответствовал одному из авторов данной статьи).

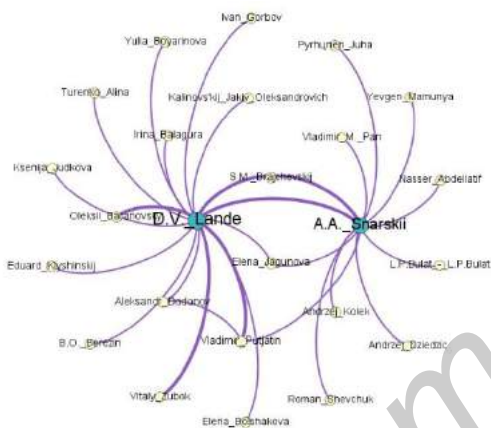


Рисунок 5 -- Небольшая сеть соавторов, построенная с учетом указанных дескрипторов

Динамика роста данной сети при зондировании сервиса GoogleScholarCitations имеет вид, представленный на рис. 6.

Применение методов кластерного анализа позволяют выявлять наиболее тесно связанные между собой группы ученых-соавторов, научных школ, экспертных групп. В данном случае под научной школой будем понимать неформальный творческий коллектив исследователей разных поколений, объединенных общей программой и стилем исследовательской работы, которые действуют под руководством признанного лидера. На рис. 7 показано визуальное представление процесса выявления кластеров путем пошагового удаления наименее весомых ребер из сети соавторства, построенной по тематике ComplexNetworks.

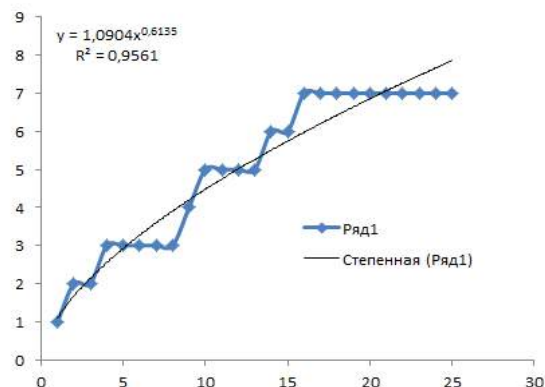


Рисунок 6 -- Динамика роста сети соавторов, построенная с учетом указанных дескрипторов

В данном случае приведенный выше алгоритм применялся к группе из 10-и ученых, имеющих в соответствии с данными GoogleScholarCitations наибольшую цитируемость.

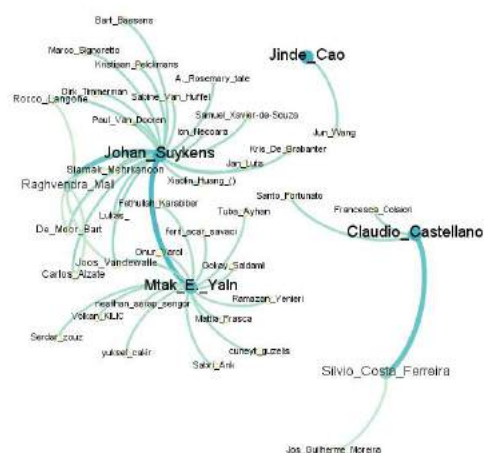
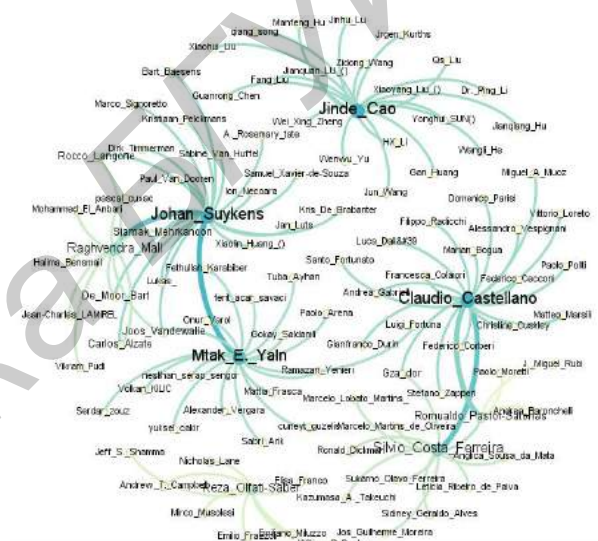


Рисунок 7 -- Кластеризация путем пошагового удаления ребер средствами программы Gephi

4. Методика построения модели предметной области

Дальнейший детальный наукометрический анализ предметной области позволяет выделить наиболее активных в научном сотрудничестве

ученых и научные группы для решения заданных вопросов.

Авторами была предложена и опробована методика наукометрического анализа, включающая методы анализа сложных сетей, методы фильтрации текста, визуализацию данных с помощью программных средств Gephi. На первом этапе определяется область и научные направления, по которым будет проведен анализ, загружается из сети (базы данных) и фильтруется файл с реферативной информацией. Результат первого этапа – отфильтрованные по определенным дескрипторам данные об авторах и связях между ними, т.е. матрица, соответствующая сети.

Второй этап – создание сети соавторов выбранной предметной области, а также основных характеристик сети с помощью программных средств Gephi, а также расчет дополнительных параметров с помощью собственных программных средств. В результате второго этапа определяются основные качества сотрудничества ученых, научные группы и самые коммуникативные ученые по определенному научному направлению.

Третий этап посвящен отбору полнотекстовых публикаций самых коммуникативных ученых и созданию текстового корпуса для выявления основных терминов (слов и словосочетаний) по научным направлениям.

На четвертом этапе выполняется визуализация сетей терминов ученых и области в целом, расчет основных параметров. Проводится обобщение результатов, описание основных характеристик, тенденций в области.

Результаты исследований дают возможность научно обосновать, автоматизировать и ускорить процедуру подбора компетентных экспертов для решения различных вопросов так и производства новой продукции.

Заключение

Предложен и реализован подход к формированию сетей соавторства в рамках предметной области, ограничительными элементами которого составляют некоторые маркеры знаний (теги), заранее заданные учеными – участниками проекта GoogleScholarCitations.

Следует отметить принципиальное отличие предложенной модели автоматического формирования сетей соавторства от существующих, базирующихся на непосредственном участии экспертов при выборе конкретных узлов и связей. В данном случае исследователь для построения сети использует лишь крупницы знаний, представленных в виде набора базовых тегов. В дальнейшем программа использует знания, заложенные соавторами, теги отмеченные как главные для них. Т.е. экспертная среда в этом случае существенно расширяется.

Модель применялась для отраслей науки ComplexNetworks и TextMining в рамках сервиса GoogleScholarCitations, но предложенный подход можно использовать и для других научных областей, или для других наукометрических массивов.

Результаты моделирования с помощью предложенной в разделе 3 методики также могут использоваться для создания модели предметной области, поиска групп экспертов. Выполнение подобных исследований могут способствовать усовершенствованию инструментария научных сервисов за счет внедрения технологий наукометрического анализа, улучшению эффективности аналитической деятельности.

Библиографический список

- [Barabási, 1997] Barabási A., Albert R. Emergence of scaling in random networks // Science. – 1997. – 286. – P. 509-512.
- [Brezina, 2012] Brezina V. Use of Google Scholar in corpus-driven EAP research // Journal of English for Academic Purposes. – 2012. – 11. – P. 319-331
- [Erdős, 1960] Erdős P., Rényi A. The Evolution of Random Graphs // Magyar Tud. Akad. Mat. Kutató Int. Közl., 1960. – 5. – P. 17-61.
- [Kalogeraki, 2002] Kalogeraki V., Gunopulos D., Zeinalipour-Yazti D. A Local Search Mechanism for Peer-to-Peer Networks // Proc. of CIKM'02, McLean VA, USA, 2002.
- [Lande, 2013] Lande Д.В., Горбов И.В., Балагура И.В. Характеристики сети соавторов медицинских наук // Клиническая информатика и телемедицина, 2013. - Т.9., Вып.10. - С. 141-144.
- [Lande, 2015] Lande D. A Domain Model Created on the Basis of Google Scholar Citations // CEUR Workshop Proceedings (ceur-ws.org). Vol-1536 urn:nbn:de:0074-1536-8. Selected Papers of the XVII International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2015) Obninsk, Russia, October 13-16, 2015.- pp. 57-61.
- [Liu, 2015] Liu J., Li Y., Ruan Z., Fu G., Chen X., Sadiq, Deng Y. A new method to construct co-author networks // Physica A. – 2015. – 419. – P. 29-39.
- [Newman, 2003] Newman M.E.J. The structure and function of complex networks // SIAM Rev. – 2003. – 45. – P. 167-256.
- [Ortega, 2015] Ortega J. How is an academic social site populated? A demographic study of Google Scholar Citations population // Scientometrics. – 2015. – 104. – P. 1-18

CREATION OF NETWORKS OF THE CO-AUTHORSHIP ACCORDING TO THE GOOGLE SCHOLAR CITATIONS SERVICE

Lande D.V., Balagura I.V., Andrushchenko V.B.

Institute for Information Recording NAS of Ukraine, Kiev, Ukraine

dwlande@gmail.com

balaguraira@mail.ru

valentyna.andrushchenko@gmail.com

The algorithm of creation of the network of a co-authorship of scientists regulated by their scientific interests is given in work. The network of a co-authorship is formed on the basis of sounding of the Google Scholar Citations service. It is shown that the descriptors defining subject influence the size of the formed network, and also dynamics of its growth. It is shown that clusters in networks of a co-authorship can

be considered as a basis for identification of schools of sciences.

Introduction

The objective of the work – is the description of the theoretical principles and methods of automatic formation of the co-authoring networks, in particular in the fields - Complex Networks and Text Mining sounding the great information network. To attain this aim the specific algorithm of Google Scholar Citations service scanning is used to receive the representative co-authors bank as the base units for the future network. Within the sounding notion we will perceive the small size fetch of the most important content from the large networks, which couldn't be sounded by the processing reason [Lande, 2015].

It's evident that the co-authoring network can be of a big size, if is not measured by the defined theme, targeted by the tags of the first author – the origin of the network formation.

Such an effect complicates considerably the perception of the formed network and reduces to the effect of “themes drift”. Also the identical last names and initials spelling can occur. To cope with these effects the thematic filtering is used i.e. the used descriptors are referred to authors of the scientometric network, and define their thematic direction.

Accordance of these descriptors in the final analysis defines the size of the formatting co-authoring networks and the dynamic of its growth. In addition the clusters identification in such networks can be perceived as a basis for the science schools, experts' groups etc. extraction [Lande, 2013].

Main Part

It is appropriate to use the approved on the peering networks (peer to peer, P2P) models, based on the equality of participants. Peering networks consist of units; each of it interacts only with the several subsets of other units, which corresponds to the co-authoring network.

The sounding of the reference model network is provided according to the next algorithm [Lande, 2015]:

1. The several number of reference (sounding) network units are defined as the basic ones for the new network, according to the sounding results (in the common case the one unit is chosen).
2. For every unit of the reference network the allied units are defined (co-authors), they are added to the network as the result of sounding. The arcs-connections to these units are formed from the root unit.
3. From the current unit of the reference network the pass to the randomly chosen neighboring unit (co-author) is implemented
4. If the circularity takes place or there is the mismatch of the unit to the several measuring condition, the pass to another randomly chosen unit is implemented. If there is no such unit, the network is considered to be built.

Exactly on the results of the quality modeling there was made a conclusion about the opportunity of forming the small branches of connected co-authors, according to the tags, users of Google Scholar Citations service are interested in.

The described algorithm was adapted to the real co-authoring network of Google Scholar Citations in such a way:

1. The first (root) author to begin the sounding is chosen.
2. The list of the basis tags according to the most important conception is defined appraisal.
3. The page of the web-service of the chosen author opens.
4. All the co-authors from the chosen author profile are added to the forming network.
5. The arcs-connections are tracing to these units (co-authors) from the root unit (author).
6. From the list of the forming network units the unit for the next page transition for the further analysis is chosen randomly. This unit must meet the themes of the chosen subject field (its tags are included into the descriptors, defined on the step 2) and is not the part of the units, which were traced.
7. If such a unit is chosen, so the pass to the step 3 is implemented.
8. If there is no such an author, the network is considered to be built.

According to the described algorithm the process of the network sounding from the several (root) unit is stopped after the circularity, when according to the algorithm the pass is implemented to the unit, been traced, and also if the left units are vary from the main themes (it defines by taking into account the lexical make-up of the tags). And the exact “circularity” is the feature of the pass to another root author or the end of the sounding process.

Conclusion

The suggested attempt is directed to form the networks of co-authorship in frames of the knowledge domain, limited elements of which are the several tags, targeted previously by the scientists – members of the Google Scholar Citations project.

It's necessary to notice that the basic difference of the suggested model of automatic way of the network formation from the existed ones, based on the direct participation of the experts in choosing straight units and connections. In this case the researcher uses only the tiny knowledge parts, inlayed by co-authors, tags, marked as the main for them. Thus the expert environment is widened considerably.

The model is used for the science fields Complex Networks and Text Mining in frames of the Google Scholar Citations, but the suggested attempt can be used for other knowledge domains and scientometric arrays.

The modeling results received using the procedure proposed in the chapter 3 also can be applied to create the subject field model.



УДК 681.3.

РАЗРАБОТКА ОБЪЕКТИВНЫХ МЕТОДОВ ОЦЕНКИ КОМПЕТЕНТНОСТИ УЧАСТНИКОВ НАУЧНЫХ ПРОЕКТОВ НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО АНАЛИЗА

Рогушина Ю.В. *, Гладун А.Я.**

**Институт программных систем НАН Украины, Киев, Украина*
ladamandraka2010@gmail.com

***Международный научно-учебный центр информационных технологий и систем НАН Украины,
Киев, Украина*
glanat@yahoo.com

Предложены объективные методы оценки компетентности участников научных и научно-образовательных проектов. Представленные методы базируются на семантическом сопоставлении описания проекта и документов, характеризующих компетентность его участником в выбранной сфере. Кроме того, предлагается использовать знания, извлекаемые из открытой среды Web – Wiki-справочники, наукометрические базы, официальные сайты организаций и сопровождающие их метаданные, онтологии предметных областей. Разработана специализированная онтология научной деятельности, позволяющая стандартизировать терминологическую базу описания квалификации участников проектов.

Ключевые слова: онтология, тезаурус, компетентность.

Введение

В настоящее время в различных сферах научной и технической деятельности предлагается большое количество разнообразных проектов – научных исследований, издания научных трудов, проведения междисциплинарных исследований, проведения научных исследований отдельными учеными и т.д.

При их сравнении и анализе таких проектов, как правило, оценивается научный уровень самого проекта, возможности его выполнения участниками и целесообразность реализации, прогнозируется научная и экономическая эффективность предполагаемых результатов [Васенкова и др., 2011]. Проекты могут отличаться как по сложности, так и по источникам финансирования – государственным, из международных фондов, от корпораций и частных фондов.

В общем случае целесообразность финансирования проекта сводится к двум факторам – к актуальности и полезности решаемой проблемы и к прогнозу того, что данный коллектив участников способен решить поставленную проблему за указанное время.

1. Компетенции и компетентность исследователей

Правилам подготовки различных научных проектов посвящено много методических разработок [Зінченко и др., 2015] а анализ их актуальности и прогнозирование эффективности является задачей экспертов соответствующих предметных областей (ПрО) [Петровский и др., 2009].

Эксперты, оценивающие проекты, должны обладать достаточно глубокими знаниями в той сфере, к которой относятся все проекты в целом, но могут не являться экспертами во всех анализируемых подобластях каждого отдельного проекта. В то же время требуется, чтобы авторы предлагаемых проектов обладали высокой квалификацией именно в той предметной области (ПрО), к которой относится конкретный проект, а не только в данном научном направлении. Поэтому возникает необходимость в разработке средств автоматизированной оценки компетентности потенциальных участников проекта на основе сопоставления материалов, которые описывают их знания, квалификацию и навыки, с описанием самого проекта. Необходимо, чтобы в таком

сопоставлении использовались знания о ПрО проекта.

Важным фактором, влияющим на эффективность выполнения научно-исследовательской работы, является компетентность исполнителей в соответствующей проекту ПрО. При этом следует различать общую квалификацию исследователей и их компетентность в тех вопросах, которые непосредственно связаны с выполнением проекта, включая как теоретические знания в этой сфере, так и практические навыки, и опыт, необходимый для успешной реализации проекта.

Следует различать компетенцию и компетентность. *Компетенция* – это понятие, в общем случае не связанное с конкретным лицом; *компетентность* – это отношения между лицом и компетенцией, которое означает, что определенный человек владеет данной компетенцией. В данном случае компетенция – это свойство проекта, а компетентность – свойство исследователя.

В общем смысле *компетенция* – способность успешно действовать на основе имеющихся знаний и практического опыта при решении задач. Элементы компетенции – знания и навыки, жизненный опыт, способности, черты характера, интеллект и т.д., объединенные в разных конфигурациях для решения человеком конкретных проблем.

В области управления персоналом под компетенцией обычно понимают формально описанные требования к личностным, профессиональным и другим качествам сотрудника. Однако в области научных исследований понятие «компетенция» является нечетким и сильно зависимым от специфики конкретной ПрО.

В то же время результаты работы научных сотрудников обычно довольно четко формализованы и открыты для анализа: это научные публикации, патенты, отчеты, описания прикладных разработок и прочие материалы, представленные в виде естественных языковых текстов с элементами структурированных данных (таблиц и графиков) и мультимедиа. В большинстве случаев все эти материалы представлены в электронной форме, которая предполагает создание средств их автоматизированного анализа. Значительная часть научных публикаций доступна через Web.

Компетентность – уровень достижений (опыта, знаний, привычек) лица в определенной ПрО. Компетентность может быть определена на основе анализа деятельности специалиста, его осведомленности в достижениях науки и техники, его понимания исследуемых проблем и возможных путей их решения. Для количественной оценки уровня компетентности используется *коэффициент компетентности*.

Оценки, получаемые на основе анализа формальных характеристик участников проекта (такие как дипломы, количество публикаций в

указанной ПрО, стаж работы и т.п.), являются слишком приблизительными и не учитывают семантику решаемой задачи.

Поэтому разработка объективных методов оценки компетентности претендентов на основе онтологического анализа с целью прогнозирования успешности реализации научных проектов представляется важной научной проблемой, решение которой должно базироваться на использовании и обработке знаний о той ПрО, для которой планируется выполнять исследования.

2. Постановка задачи

Для обеспечения объективных методов оценки компетентности участников проекта на поддержку и участие в разработке научных и научно-образовательных проектов необходимо разработать средства сопоставления описания научно-технического проекта и сведений о его участниках, характеризующих их знания и опыт в ПрО проекта, учитывающее знания об этой ПрО.

Такое сопоставление сводится к анализу двух множеств естественно-языковых документов и связанных с ними метаданных и знаний. Рассматриваемый подход позволит интегрировать персонифицированные онтологические знания об экспертах в интересующей нас ПрО и логические методы их обработки, используя онтологический анализ и методы семантического поиска.

3. Источники сведений о ПрО и специалистах

Предполагается, что основным источником сведений о проекте является его естественно-языковое описание (заявка, техническое задание), а также дополнительные внешние информационные ресурсы – онтологии, Wiki-страницы и т.д., содержащие структурированные и семантически значимые сведения о ПрО.

Более сложно проанализировать информация об участниках проекта. Часть сведений о них четко формализована и может быть однозначно оценена без учета семантики ПрО и дополнительных знаний о проекте. В частности, это уровень образования участников, их стаж работы в соответствующей области, наличие ранее выполненных проектов. К сожалению, этих сведений недостаточно для того, чтобы определить их компетентность для конкретного проекта, особенно для новых и динамично изменяющихся ПрО. Проблема заключается в том, что научные сотрудники, получившие одинаковое образование и даже защитившие диссертации по одной специальности, в процессе своих исследований специализируются в различных вопросах.

Одним из наиболее объективных критериев оценки сферы компетентности научных сотрудников является анализ их публикаций,

представленных в Web, – научных статей, докладов, отчетов и презентаций.

Пертигентность документа (публикации) проекту зависит от таких параметров, как количество упоминаний основных терминов проекта в документе и количество основных терминов, упомянутых в проекте. Существует много работ по автоматическому определению компетентностей на основе документов.

Однако различные информационные источники в Web имеют различную оценку качества представленной в них информации.

Кроме того, важно учитывать оценку деятельности исследователей научным сообществом – через наличие ссылок на их работы, а также упоминанием их работ в учебной и справочной литературе. Кроме того, источники сведений об исследователях могут быть импортированы из баз знаний интеллектуальных приложений, обеспечивающих персонализированное информационное обслуживание пользователей – например, из семантических поисковых и рекомендующих систем [Рогущина, 2015].

Таким образом, в качестве источников сведений о сфере компетентности авторов проекта предлагается рассматривать следующие ИР:

- Тексты статей (публикаций, тезисов, монографий, учебников, технических отчетов, авторских свидетельств и прочих объектов интеллектуальной собственности, представленных в виде естественно-языковых документов), опубликованные в Web, с учетом рейтингов наукометрических баз данных;
- Тексты статей (публикаций, тезисов, монографий, учебников, технических отчетов, авторских свидетельств и прочих объектов интеллектуальной собственности, представленных в виде естественно-языковых документов), представленные самими авторами (например, тексты статей из платных журналов, промежуточные отчеты и т.п.);
- Wiki-страницы авторов проекта, содержащиеся в различных Википедиях и Wiki-справочниках, которые обеспечивают структурированное представление информации;
- Wiki-страницы организаций, с которыми связаны авторы проекта;
- описание специальностей авторов проекта (паспорт специальности для научной степени, диплома);
- отчеты о ранее реализованных проектах и исследованиях разных типов, с учетом успешности их выполнения.

Одним из наиболее значимых факторов, определяющих квалификацию научного сотрудника, являются его диплом и – при наличии ученой степени – специальность, по которой защищена диссертация. Каждая специальность имеет свой паспорт – документ, закрепляющий область

исследований, за которые может быть присуждена учёная степень кандидата или доктора наук по соответствующей специальности. В паспорте специальности заложено также определение специальности, науки, в которые входит данная специальность.

В значительной мере компетентность сотрудника определяется местом его работы. Поэтому целесообразно учитывать сведения об организациях (коллективах), в которых работают авторы проекта, и сопоставлять их с ПрО проекта. Источником информации об организациях может служить Web – значительная часть предприятий и учреждений имеет собственные сайты и порталы. Если сведения представлены на официальном сайте организации, то они должны соответствовать действительности.

Иногда полезнее использовать уже структурированные знания об организации. *Организационная онтология* – это онтология, которая отображает знание об организационной и функциональной структуре определенного субъекта экономической деятельности, то есть его основные компоненты и связи между ними. Она содержит информацию о работниках предприятия, иерархию производственных отношений между ними; ресурсы, которые используются на предприятии в процессе производства; продукцию, создание которой есть следствием функционирования предприятия, и структурные единицы предприятия и связи между ними.

4. Описание наукометрических БД

Эффективность научной деятельности может оцениваться с использованием как качественных, так и количественных показателей. В основе качественных оценок лежат выводы экспертов. Однако субъективность подобных оценок значительно снижает достоверность получаемых результатов, а отсутствие количественного выражения затрудняет их использование.

Наукометрические показатели удобны для оценки фундаментальных исследований, востребованность которых оценивают через отзывы научного сообщества на публикации с результатами исследований. Возросший интерес к наукометрическим показателям вызван в первую очередь возможностью автоматизации процесса оценивания результатов научной деятельности [Штовба и др., 2013].

Сейчас при формировании рейтинга исследователя используют различные параметры – количество публикаций – суммарное или по отдельным типам (монографии, статьи, тезисы, публикации в изданиях, входящих в список ВАК, проиндексированные в Web of Science, Scopus или Google Scholar и т.п.) и ссылок на них. Иногда учитывают объем публикаций. На их основе формируются интегральные критерии, в которых баллы за публикацию определяются ее типом и

импакт-фактором издания, в котором она опубликована. Несколько сложнее рассчитывается рейтинг автора при наличии соавторов в публикациях. Обычно считается, что вклад всех соавторов одинаков, и оценку публикацию делят на число соавторов.

Сейчас в мире издается около 25 тыс. реферируемых журналов, в которых ежегодно публикуется около 2,5 млн. статей. Эффективность научной деятельности может оцениваться с использованием как качественных, так и количественных показателей. В [Бурков и др., 2013] анализируются наиболее эффективные и распространённые характеристики продуктивности научной деятельности, в частности, индекс Хирша и импакт-фактор.

Чтобы выявить ученых, пишущих много и качественно, в 2005 г. Х. Хирш предложил новый показатель – индекс Хирша (h-индекс) [Hirsch, 2005] который представляет собой максимальное целое число h , указывающее, что автор опубликовал h статей, каждая из которых процитирована хотя бы h раз. Простота расчетов и нечувствительность к типовым приемам искусственного улучшения вышерассмотренных показателей, мгновенно сделали индекс Хирша популярным наукометрическим индикатором.

Индекс Хирша может вычисляться с использованием как бесплатных общедоступных наукометрических баз данных в Интернете, (например, Google Scholar, Elibrary.ru, ADS NASA), так и баз данных с платной подпиской (например, Scopus или ISI Web of Science); однако платные базы данных часто тоже приводят h-индекс учёных в свободном доступе. Следует отметить, что индекс Хирша имеет разные значения одного и того же ученого в зависимости от проиндексированных ИР. Индекс Хирша дает более объективные результаты при отбрасывании ссылок авторов на собственные статьи. Например, в рейтинге ученых Украины согласно индексу Хирша выполняется подсчет по базе данных Scopus с отбрасыванием самоцитирования.

Следует отметить, что индекс Хирша хорошо работает лишь при сравнении ученых, работающих в одной области исследований из-за традиций, связанных с цитированием, в различных областях наук.

Импакт-фактор показывает, сколько раз в среднем цитируется каждая опубликованная в журнале статья в течение x последующих лет после выхода. Это численный показатель важности научного журнала, который ежегодно рассчитывается Институтом научной информации (Institute for Scientific Information, ISI) и публикуется в журнале Journal Citation Report. Импакт-фактор позволяет по формальным признакам сравнивать разные журналы и исследовательские группы.

Как правило, расчет импакт-фактора основан на трехлетнем периоде. Импакт-фактор журнала A за год A вычисляется по формуле

$$\text{Imp}(A, x) = \frac{\text{Cit}(A, x - 2, x) + \text{Cit}(A, x - 1, x)}{\text{Pub}(A, x - 2) + \text{Pub}(A, x - 1)}, \quad \text{где}$$

$\text{Cit}(A, y, z)$ – количество цитирований в течение года z статей, опубликованных в журнале A в году y в публикациях журналов, которые отслеживаются Институтом научной информации (США), а $\text{Pub}(A, y)$ – количество публикаций в журнале A за год y .

Индекс цитирования – это принятая в научном мире мера значимости научной работы какого-либо ученого или научного коллектива, которая представляет собой суммарное количество ссылок в проиндексированных работах на рассматриваемые публикации. Он отражает реакцию научного сообщества на приведенные в ней результатами исследований, т.е. уровень их востребованности другими учеными. Как правило, слабые и вторичные работы не цитируют. Цитируемость зависит не только от уровня научных результатов, но и от других факторов, например, своевременности. Для более точных оценок при определении индекса цитируемости не учитывают самоцитирование или цитирование соавторами, а также повторные цитирования одной работы одним и тем же ученым.

Для получения этих оценок используют наукометрические базы данных (НМБД) – библиографические и реферативные БД с инструментами для отслеживания цитируемости статей, опубликованных в научных изданиях.

Наиболее известна НМБД Scopus издательской корпорации Elsevier – библиографическая и реферативная база данных, индексирующая 18 000 научных изданий. Издания, которые индексируются этой НМБД, должны удовлетворять ряду условий – иметь англоязычное название и публиковать англоязычные версии аннотаций всех статей на своем Web-сайте; публиковать новые выпуски не реже одного раза в год; обладать авторитетностью (с учетом научного авторитета членов редколлегии), популярностью и доступностью и предусматривать контроль за качеством публикаций (например, научное рецензирование).

Наукометрический аппарат Scopus обеспечивает статистику цитируемости публикаций ученых и учреждений. В Scopus не используется понятие импакт-фактора, но широко применяется индекс Хирша. База данных доступна в условиях подписки через веб-интерфейс (<http://www.scopus.com>). Кроме того, авторы могут без регистрации просматривать свою страницу <http://www.scopus.com/search/form/authorFreeLookup.uri>

Не менее популярная НМБД *Web of Science* (WoS) компании Thomson Reuters объединяет реферативные базы данных публикаций в научных

журналах и патентов, в том числе базы, учитывающие взаимное цитирование публикаций.

Она содержит ссылки на полные тексты в первоисточниках и списки всех библиографических ссылок, встречающихся в каждой публикации, что позволяет в краткие сроки получить самую полную библиографию по интересующей теме. Одним из ключевых понятий ее наукометрического аппарата платформы является импакт-фактор научного издания. Доступна только для подписчиков.

Менее известна международная НМБД *Index Copernicus* (Польша) (<http://www.indexcopernicus.com>), которая включает индексирование, ранжирование и реферирование журналов, а также является платформой для научного сотрудничества и выполнения совместных научных проектов. База данных имеет несколько инструментов для оценки производительности, позволяющие отслеживать влияние научных работ и публикаций отдельных ученых или научных учреждений. В дополнение к оценке производительности, индекс Коперникус предлагает также традиционные реферирования и индексирования научных публикаций. Для использования требует регистрации.

Наиболее полно представленные в Web научные работы проиндексированы в *Google Scholar* (<http://scholar.google.com/>) – свободно доступной поисковой системе, которая индексирует полный текст научных публикаций всех форматов и дисциплин. Система обеспечивает поиск и формирование корректных ссылок в различных форматах. При поиске можно учитывать время создания и частоту цитирования документов. Основным наукометрическим показателем, который генерирует эта НМБД – индекс Хирша (как общий, так и за последние пять лет).

Существуют и национальные НМБД, ориентированные на индексацию и оценку публикаций на языках, отличных от английского. Например, Web-сайт «Украинский индекс научного цитирования» (<http://uincit.uran.ua>) предназначен для сбора, обработки и предоставления доступа к данным по показателям активности индивидуальных и коллективных субъектов научной деятельности Украины.

Предполагается, что сайт позволит просмотреть показатели публикационной активности как отдельных ученых, так и научных учреждений Украины, узнать показатели цитируемости их работ и ключевые наукометрические показатели.

Информация о публикации и показатели их цитируемости получают сайтом с внешних наукометрических источников, среди которых: международный реестр ученых ORCID, наукометрические платформы Web of Science и SciVerse Scopus, научно-издательская инфраструктура «Научная периодика Украины». На

данный момент сайт работает в тестовом режиме и не обеспечивает заявленные возможности.

При определении рейтинга ученых для определения возможного участия в Международных инвестиционных проектах, например проектах Рамочной Программы ЕС (Horizon-2020, TEMPUS, Erasmus, TESIS, DAAD, EUREKA, CRDF и др.) одним из важных аспектов выбора участников для международных проектов является, кроме его рейтинга научных трудов, его участие и опыт в предыдущих международных проектах, что является гарантией своевременного выполнения запланированных рабочих пакетов – WP [Бедоев, 2015]. Эту информацию также можно найти в Интернете на специализированных сайтах Еврокомиссии.

5. Использование семантических Wiki-ресурсов

В различных Wiki-проектах широко используются категории и семантические свойства. Чтобы связать страницу с определённой категорией, достаточно добавить в ее код имя категории в двойных квадратных скобках. Для добавления семантических свойств, задающих связи между страницами (и соответствующими понятиями ПрО) используется следующая конструкция: [[имя свойства::имя страницы]], которая обеспечивает формирование семантических троек “текущая страница”(свойство)”указанная страница”. Например, конструкция [[столица::Украина]] на странице “Киев” указывает, что Киев является страницей Украины.

В различных Wiki-справочниках представлены страницы организаций и – реже – их сотрудников. Если авторы проекта указывают на свою связь с такими страницами, то анализ их семантических свойств позволяет извлечь из них знания об этих объектах и – при наличии дополнительных навыков – пополнить соответствующую онтологию, классами которой являются категории, экземплярами – страницы, а отношениями – семантические свойства страниц. В частности, можно выявить связи с понятиями онтологии научной деятельности и онтологии ПрО проекта. Например, Wiki-страница позволяет установить, что организация, в которой работают авторы, относится к Национальной академии наук, а в разделе “Сфера исследований” есть понятия, описываемые в проекте. Наличие таких соответствий является положительным фактором.

6. Роль онтологии в оценке компетенций

Значительно повысить эффективность анализа компетентности авторов проекта позволит обработка информации на семантическом уровне. При этом полезно применять как специальные знания о предметной области, к которой относится проект, так и знания о типах и структуре документов, связанных с научной деятельностью.

Именно онтологии являются на сегодня широко распространенным средством представления знаний о ПрО, позволяющим анализировать и сопоставлять компетенции экспертов и разработчиков в новых исследовательских областях [Rogushina et al., 2012].

Чтобы использовать знания ПрО, необходимо использовать онтологию этой ПрО. Поиск (или разработка) такой онтологии является задачей авторов проекта, так как именно они должны определить, в какой сфере они планируют вести исследования и добиваться новых результатов на основе имеющихся у них знаний. Если же авторы не предоставляют такую онтологию, то используется более общая онтология той области знаний, к которой относится рассматриваемый проект. Например, авторы проекта «Разработка интеллектуальной системы информационного и когнитивного сопровождения функционирования Национальной рамки квалификаций» могут предоставить онтологию «Компьютерно-ориентированные учебные среды» (рис.1), но, если они это не сделали, то будет использована онтология соответствующего научного направления «Информационные и коммуникационные технологии».

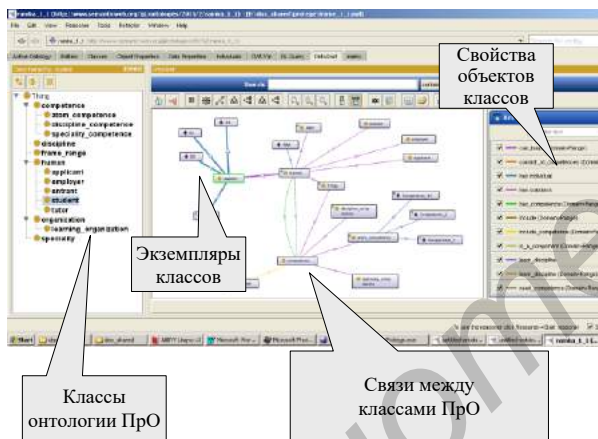


Рисунок 1 – Пример онтологической модели ПрО

Использование более общих онтологий приводит к двум проблемам: 1) необходимо обрабатывать значительно больший объем терминов (на этапе построения тезауруса проекта); 2) в проекте могут встречаться более общие, не существенные для сути проекта термины, обработка которых несколько понизит точность результирующей оценки.

Кроме того, сам факт наличия у авторов проекта онтологии, соответствующей ПрО, как правило, свидетельствует о более глубоких познаниях в выбранной области (особенно если речь идет об информационных технологиях).

Наряду с онтологиями ПрО, целесообразно применять общую онтологию научной деятельности, которая позволяет однозначно устанавливать терминологию, связанную с рейтингом публикаций, научными степенями и учеными званиями, типами организаций и т.д. Такая онтология специально разрабатывается для целей

определения компетентности авторов проекта на основе организационных онтологий научных организаций, классификатора УДК и паспортов специальностей ВАК. В такой онтологии отражаются, к примеру, такие отношения между классами, как «быть соавтором», «работать в организации», «быть автором публикации», «иметь ученую степень по специальности» и свойства «являться публикацией ВАК», «иметь индекс Хирша» (рис.2).

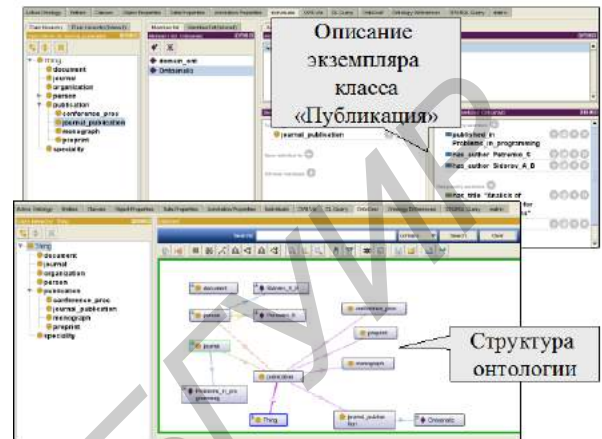


Рисунок 2. Онтология научной деятельности, используемая при анализе компетентности авторов проекта.

Эта онтология предлагается авторам проекта в качестве образца для описания и классификации предлагаемых документов, которые должны удостоверить уровень их компетентности в научной деятельности в целом и в предметной области проекта в частности.

7. Тезаурус и онтология ПрО как средство сопоставления естественных языковых документов

Для сопоставления проекта и компетенций участников проекта предлагается построить тезаурус проекта и тезаурусы всех ИР, описывающих участников проекта.

Тезаурус естественного языка ИР представляет собой проекцию онтологии, анализируемой ПрО на рассматриваемый документ. Более подробно алгоритм построения таких тезаурусов рассматривается в [Гладун и др., 2010].

Тезаурус участников определяется как объединение тезаурусов отдельных ИР. При этом следует учитывать вес отдельных ИР, в котором учитываются как значимость самого документа для описания компетенции, так и уровень доверия к самому ИР. Например, больший вес имеет автореферат диссертации, чем описание диплома, а при оценке публикаций полезно учитывать рейтинг различных журналов и конференций в наукометрических базах данных.

Можно говорить о том, что формируется корпус текстов, описывающий авторов проекта. Корпус —

не просто большое собрание текстов. Это собрание сбалансированное (тексты разного типа отобраны в определенной пропорции). Предполагается, что каждый из авторов проекта сам формирует набор документов, которые наиболее пертинентны предложенному проекту. Например, если у автора есть n научных публикаций, то он сам отбирает m из них, которые связаны с той проблемой, которой посвящен проект, причем не обязательно это должны быть статьи с наибольшим индексом цитирования или входящие в Scopus. Однако автор должен стремиться к тому, чтобы для всех понятий онтологии ПрО, для которых обнаружены лингвистические аналоги в тексте проекта, были найдены такие соответствия и в его работах (вес каждого отдельного сопоставления определяется весом ИР, определяемым как функция от статуса документа и его рейтинга).

Статус документа характеризует уровень его документальной подтвержденности, а рейтинг – его оценку наукометрическими БД.

Вначале строится тезаурус проекта Th_{proj} , который зависит от описания проекта и от выбранной онтологии ПрО. Он представляет собой множество пар (t_i, q_i) , где $t_i \in T$, T – множество терминов онтологии ПрО $O_{ПрО} = \langle T, R, A \rangle$, а q_i – количество найденных соответствий, определяющее вес этого термина (если определенный термин встретился в описании проекта 10 раз, то предполагается, что он более важен, чем тот, который встретился 2 раза). Для каждого термина онтологии ПрО предпринимается попытка найти фрагменты текста

Общая оценка компетентности коллектива авторов определяется как массив $\left(t_i, \sum_{j=1}^m R_{ИР_j} * v_{ИР_j} \right)$, где $t_i \in T$, T – множество терминов онтологии ПрО $O_{ПрО} = \langle T, R, A \rangle$, $R_{ИР_j}$ – количество найденных соответствий с этим термином в j -м ИР, а $v_{ИР_j}$ – вес j -го ИР.

Чтобы определить вес j -го ИР $v_{ИР_j}$, учитывается: 1) для научных публикаций – является ли статья публикацией ВАК (+5), материалами конференции (+1), опубликована в зарубежном издании (+3), проиндексирована Google Scholar (+2) и Scopus (+10); 2) для паспорта специальности, диплома – +5; 3) для описания профиля деятельности организации – +3; 4) для описаний ранее выполненных проектов – +3; 5) для автореферата диссертации – +3. Прочие документы учитываются с весом 1.

Важно, что в этом массиве присутствуют не все термины онтологии ПрО (которая в целом может быть значительно шире, чем непосредственно

рассматривается в проекте), а лишь те, для которых найдены соответствия в проекте.

Очевидно, что различные ИР имеют разный вес при оценке компетенций специалистов. Например, публикация в реферируемом журнале более показательна, чем выложенный в Web технический отчет. Поэтому при оценке публикаций целесообразно учитывать импакт-фактор опубликовавшего его издания.

В таком случае общая оценка компетентности авторов проекта оценивается следующим образом:

$$C = \sum_{i=1}^n q_i * \left(\sum_{j=1}^m R_{ИР_j} * v_{ИР_j} * Imp(ИР_j) \right) \quad (1),$$

где $Imp(ИР_j)$ – импакт-фактор издания, опубликовавшего ИР j .

В дальнейшем целесообразно ввести различные нормализующие коэффициенты, позволяющие уменьшить воздействие документов большого объема, слабо насыщенных терминами ПрО. Однако это требует более детального исследования состава документов, предоставляемых авторами на экспертизу, и в значительной области зависит от специфики проводимого конкурса.

Основным недостатком оценки (1) является то, что она не использует семантику ПрО, т.е. в ней не учитываются связи между терминами онтологиями (классами и экземплярами), представленные в онтологии ПрО. Для более корректного учёта этих связей предлагается использовать следующую оценку:

$$C = \sum_{i=1}^n q_i * \left(\sum_{j=1}^m R_{ИР_j} * v_{ИР_j} \right) * s_i \quad (2),$$

где параметр s_i определяет значимость i -го термина онтологии через количество его отношений с другими терминами онтологии, также вошедшими в тезаурус проекта, с учетом семантического расстояния между ними.

Чтобы, кроме специализации в выбранной ПрО, учитывать и общую квалификацию каждого из авторов проекта, целесообразно учитывать в оценке и их рейтинг исследователей, полученный из наукометрических баз данных. В частности, предлагается использовать сведения из Google Scholar и Scopus, так как эта информация является открытой и доступна всем пользователям Web. Кроме того, это позволяет дифференцировать квалификацию отдельных авторов, а не просто суммировать их результаты. При том, что оценка качества формирования коллектива авторов не является задачей экспертов, однако наличие в нем людей с резко различающейся квалификацией может вызывать вопросы. Предлагается

использовать следующий критерий оценки x -го автора проекта:

$$C_x = \sum_{i=1}^n q_i * \left(\sum_{j=1}^m P_{IP-x_j} * v_{IP-x_j} \right) * s_i * h_x \quad (3),$$

где учитываются не все ИР, а только непосредственно связанные с ним, а h_x – *cevvf* индекса Хирша ученого из Google Scholar и Scopus.

Общая оценка квалификации участников проекта может оцениваться как сумма оценок участников либо как их нормализованная сумма. Первый подход предпочтительнее, так как знания и опыт каждого могут использоваться независимо от количества участников. Поэтому нормализованная оценка может применяться только как дополнительная.

Заклучение

Предложен новый подход к проблеме объективного оценивания компетентности в контексте новых информационно-коммуникационных технологий, характеризующихся высокой динамикой развития научных исследований, использования и обработки информационных ресурсов об экспертах, интересующей нас ПрО. Целью разработки представленных методов является объективизация процесса оценки квалификации потенциальных участников на основе учет большого количества доступных знаний об их квалификации в сфере разработки.

Предложенный подход позволяет на основе описания проекта и естественной языковой информации об участниках проекта (их публикаций, дипломов, описаний ранее выполненных проектов, сведений об их организациях и т.д.) строить тезаурус проекта и тезаурусы участников, которые базируются на онтологии предметной области проекта, сопоставлять эти тезаурусы и определять на основе этого уровень компетентности участников.

Библиографический список

- [Hirsch, 2005] Hirsch, J. E. An index to quantify an individual's scientific research output / J.E.Hirsch // Proc. of the National academy of Sciences of the United States of America, 2005, 102(46), P.16569-16572.
- [Rogushina et al., 2012] Rogushina, J., Ontology-based competency analyses in new research domains / J.Rogushina., A. Gladun // Journal of Computing and Information Technology. V.20, N. 4, 2012, P.277-293].
- [Бедоев, 2015] Бедоев, Д.Г. Инвестиционное сотрудничество РФ и ЕС в сфере информационных технологий: проблемы и перспективы / Д. Г. Бедоев // Молодой ученый, №14, 2015, С. 226-230.
- [Бурков и др., 2013] Бурков, В.Н. Параметры цитируемости научных публикаций в наукометрических базах данных / В.Н.Бурков, А. А. Белошицкий, В. Д. Гогунский // Управління розвитком складних систем, 15, 2013, С.134-139.

[Васенкова и др., 2011] Васенкова, Е.И. Сравнительный анализ эффективности научных проектов / Е.И. Васенкова, Г.М. Казяк // XX международная научно-практическая конференция «Управление в социальных и экономических системах, 2011, С.101-103. – <http://www.bsu.by/Cache/pdf/239773.pdf>.

[Гладун и др., 2010] Гладун, А.Я. Тезаурус предметной области – инструмент представления знаний при проблемно-ориентированном поиске в Web / А.Я. Гладун, Рогушина Ю.В. // Праці Міжнародної науково-технічної конференції «Штучний інтелект. Інтелектуальні системи», 2010, Т. 2, С.78-90.

[Зінченко и др., 2015] Зінченко, О.С. Грантрайтинг: метод. рек. для органів публічної влади щодо написання проектних заявок / О.С. Зінченко, О.В. Кулініч, П.Ю. Куліш, за заг. ред. О.В. Кулініча. – Харків, 2015. — 80 с.

[Петровский и др., 2009] Петровский, А.Б. Построение интегральных показателей оценки результативности научных проектов / А.Б. Петровский, Г.В. Ройзензон, И.П. Тихонов // Intelligent Support of Decision Making / Ed. by K. Markov, A. Voloshyn, K. Ivanova, I. Mitov, –No. 10, Sofia: FOI ITHEA, 2009, P. 59-66.

[Рогушина, 2015] Рогушина, Ю.В. Разработка средств персонализации интеллектуальных Web-приложений / Ю.В. Рогушина // Материалы V международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» OSTIS-2015, Минск, БГУИР, 2015, С.265-270.

[Штовба и др., 2013] Штовба, С.Д. Обзор наукометрических показателей для оценки публи-кационной деятельности ученого / С. Д. Штовба, Е.В. Штовба // Управление большими системами: сборник трудов, (44). – <http://belmapo.by/assets/files/naukometricheskie-pokazateli-dlya-ocenki-deyatelnosti-uchenogo.pdf>.

DEVELOPMENT OF OBJECTIVE ONTOLOGY BASED METHODS FOR COMPETENCE EVALUATION OF SCIENTIFIC PROJECTS AUTHORS

Rogushina J., Gladun A.

Institute of software systems of National Academy of Sciences Ukraine, Kiev, Ukraine

ladamandraka2010@gmail.com

International Research and Training Center of Information Technology and Systems of National Academy of Sciences Ukraine, Kiev, Ukraine

glanat@yahoo.com

Objective methods of competence evaluation of scientific and educational projects researchers are proposed. These methods are based on the semantic mapping of project description and documents characterizing the competence of its researchers in selected subject domain. In addition, it is proposed to use the knowledge acquired from the Web open environment – Wiki-encyclopedias, scientometric database, the official sites of the organizations and their metadata, domain ontologies. Specialized ontology of scientific activity which allows to standardize the terminological base for researcher qualifications describing is designed. In addition, it is assumed ontology projects.

Keywords: ontology, thesaurus, competence.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.04: 004.822

АНАЛИЗ ПРОЦЕССОВ ИНФОРМАЦИОННОГО ОБМЕНА В НАУКОМЕТРИЧЕСКИХ БАЗАХ ДАННЫХ

Потебня А.В., Погорелый С.Д.

Киевский национальный университет им. Тараса Шевченко, г. Киев, Украина

admin@artem.bz.ua

sdp@univ.net.ua

В статье проведено исследование процессов распространения информации в наукометрических базах данных. На основе разработанной математической модели с применением методов комбинаторной оптимизации определены тенденции кластеризации баз данных и установлены основные способы поддержания их целостности. Выполнено экспериментальное исследование системы *ArXiv*.

Ключевые слова: наукометрическая база данных; задача разбиения графов; функция модулярности.

Введение

Модели сложных систем (*complex system*), которые описываются набором отдельных компонентов и способом взаимодействия между ними, являются чрезвычайно распространенными в современных научных исследованиях. К ним относятся социальные сети, наукометрические базы данных, всемирная паутина WWW, коммуникационные структуры и т.п. При этом, формирование сложных систем связано с появлением ряда новых свойств, в частности, синергизма (*synergy*) и эмерджентности (*emergence*). Явление синергии связано со стремительным ростом эффективности всей системы по сравнению с деятельностью отдельных компонентов. Эмерджентность является проявлением целостности системы и определяет появление качественно новых свойств, которые не являются присущими ни одному из отдельных участников [Newman, 2010].

В настоящее время общепринятой является глобальная интеграция научных исследований, которая осуществляется средствами наукометрических баз данных и социальных сетей. Однако, при этом эффективность научного развития оказывается зависимой от состояния соответствующих сложных систем, которые в связи с активизацией дисинергетических процессов могут войти в фазу рецессии или подвергнуться полному разрушению. Как следствие, важной задачей системного анализа является исследование процессов информационного обмена в таких системах и диагностика их общего состояния [Newman, 2006].

Например, популярная в данный момент база *Scopus* индексирует более 53 млн. научных работ. Кроме того, размерность этой системы растет ежегодно более чем на 2 млн. статей. Подобные тенденции развития стабильно демонстрируют другие наукометрические базы (*Web of Science, Google Scholar, IEEE Explore*) и академические социальные сети, предназначенные для обмена документами и их обсуждения (*ResearchGate, Academia.edu, Mendeley*). Вместе с тем, стремительное внедрение баз данных подвергается постоянной критике в связи с закрытостью ведущих систем, платным доступом к публикациям, непрозрачностью механизмов расчета импакт-факторов и ограничением доступа системы *Google Scholar* некоторыми издателями. Следствием стало формирование каталога открытых журналов *DOAJ* и ряда баз регионального назначения (например, *РИНЦ*), что привело к существенному разделению основных научных результатов. Кроме того, после короткой фазы роста, структура каждой из таких систем становится крайне неустойчивой и нуждается в дополнительной внешней поддержке для сохранения жизнеспособности [Fortunato, 2010].

Следует отметить, что целостность наукометрической базы определяется особенностями ее структурной организации и процессами циркулирования информации. В связи с этим, целью данной работы является разработка метода диагностики топологии и определения влияния узлов, необходимого современным системам для предотвращения катастрофического распада центрального ядра основного фрактального компонента на отдельные «острова», неспособные к обмену информацией. Кроме того, такой анализ

необходим для установления способности системы к проведению сетевой мобилизации, при которой большинство участников, получая оперативную информацию и работая в наиболее перспективном направлении, реализуют эффект синергии.

1. Формирование математической модели наукометрической базы

Чрезвычайно важной для исследования организации сложных сетей является задача обнаружения в них скрытых структур и определения режима их функционирования. Известно, что в процессе эволюции сложные системы склонны к формированию сообществ (модулей) с высокой плотностью сильных внутренних связей при незначительном количестве слабых внешних соединений. С этим связаны основные свойства таких систем – феномены «малых миров», «клуба богатых», «мало диаметра», «тяжелых хвостов» и др. Установлено, что для эффективной работы сети наиболее важны слабые межмодульные связи, по которым информация передается между отдельными сообществами. Наличие таких связей сокращает среднее расстояние между любыми научными статьями до 6 – 7 рёбер даже для крупнейших баз данных, а их разрушение неизбежно приводит к распаду системы на отдельные фрагменты. Таким образом, архитекторы наукометрических баз обязательно должны учитывать состояние слабых соединений при поддержке своих разработок.

Проблема выделения сообществ в сложных сетях является воплощением распространенной задачи разбиения графов (*graph partition problem*). Она требует поиска оптимального распределения всех статей по подмножествам, при котором заданная целевая функция (*objective function*) достигает экстремума. Важно, что задача разбиения графов относится к NP-полным задачам комбинаторной оптимизации (*combinatorial optimization problem*), и время ее решения экспоненциально зависит от размерности входных данных. Как следствие, при обработке графов реальных систем применяются эвристические алгоритмы, которые предусматривают более эффективное распределение ресурсов при поиске решений [Potebnia et al., 2015].

Применяя структурный подход, представим наукометрическую базу в виде ориентированного графа $G = (V, E)$, где V – набор публикаций, а E – множество цитирований между ними. Задача разбиения такого графа предусматривает распределение множества статей V на k непересекающихся подмножеств V_i , для которых $V = \bigcup_{i=1}^k V_i$ и $V_i \cap V_j = \emptyset$ при $i \neq j$. Обозначим через E_i множество внутренних ребер группы V_i , а через E_{ij} – набор внешних соединений между

сообществами V_i и V_j . Следует отметить, что в отличие от социальных сетей или мировой паутины, графы наукометрических баз данных являются ациклическими в связи с невозможностью цитирования еще не написанных статей. На рисунке 1 приведен пример упрощенной структуры основного фрактального компонента такого графа.

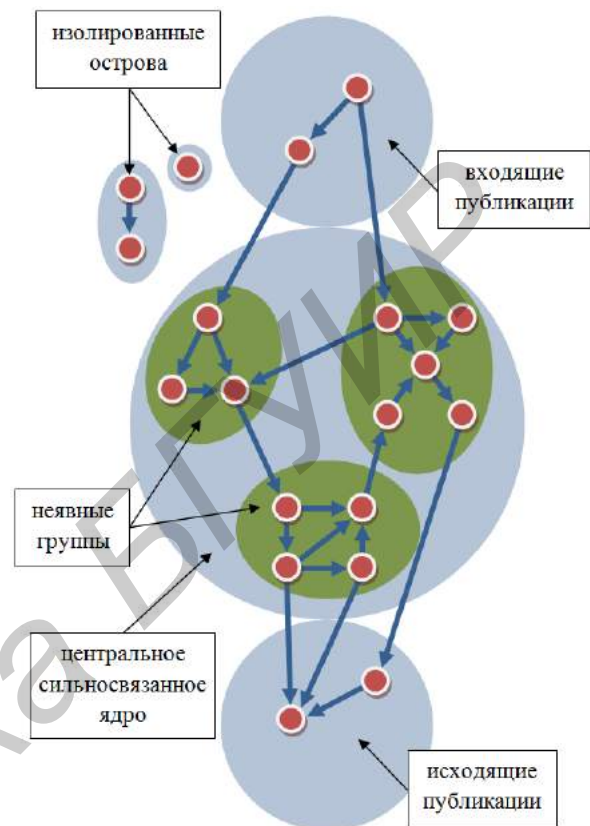


Рисунок 1 – Пример структуры основного фрактального компонента наукометрической базы

Для определения качества разбиения графа принято использовать функцию модулярности (*modularity*) Ньюмана-Гирвана Q , которая описывает соотношение плотности внутренних и межгрупповых связей для выбранного распределения узлов:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \sigma(c_i, c_j) \quad (1)$$

Здесь A – матрица смежности графа, k_i – степень вершины i , m – общее количество ребер $|E|$,

$$\sigma(c_i, c_j) = \begin{cases} 1, & c_i = c_j \\ 0, & c_i \neq c_j \end{cases} \quad (2)$$

где c_i – номер класса, к которому принадлежит вершина i . Экспериментально показано, что о реальном наличии сообществ в системе свидетельствуют значения модулярности, превышающие 0,3.

2. Методы оптимального разбиения наукометрических баз и определения влияния публикаций

Таким образом, задача оптимального разбиения графа может быть представлена в виде задачи максимизации функции модулярности Q . Однако, ее решение путем простого перебора всех вариантов является невозможным в связи с явлением «комбинаторного взрыва». Поэтому, в работе [Newman, 2006] предложен эффективный жадный алгоритм, основанный на последовательном объединении сообществ V_i и V_j , которые обеспечивают наибольший прирост функции модулярности:

$$\Delta Q(V_i, V_j) = |E_{ij}| - \frac{d(V_i)d(V_j)}{2|E|}, \quad (3)$$

где $|E_{ij}|$ – количество ребер между этими множествами, а $d(V_i)$ – степень множества V_i , которая является суммой степеней вершин $v \in V_i$.

При этом наиболее выгодным является объединение подмножеств с высокой плотностью межгрупповых связей при незначительном количестве внутренних соединений. Важно, что объединение изолированных наборов исключается, поскольку при $|E_{ij}| = 0$ величина прироста $\Delta Q(V_i, V_j)$ не может принимать положительные значения.

В этой работе для исследования реальных наукометрических баз данных применен ускоренный итерационный метод, который состоит из двух стадий [Lancichinetti et al., 2011]. Первый этап предусматривает формирование низкоразмерных сообществ путем оптимизации функции модулярности на локальном уровне. При этом определяется возможность объединения каждого множества V_i с его соседями V_j и рассчитываются соответствующие значения ΔQ . Вычисления на первой стадии заканчиваются получением локального максимума функции модулярности, а второй этап требует проведения агрегации подмножеств, образования кластеров большей мощности и определения веса межкластерных дуг. Итерации алгоритма продолжаются до получения устойчивых множеств, состав которых в дальнейшем остается неизменным.

Для определения относительной влияния научных публикаций предлагается метрика центральности узлов графа, которая может быть определена различными способами:

1. Центральность по степени (*degree centrality*) вычисляется как количество связей, инцидентных заданной вершине $C_D(v) = d(v)$.

Однако, некоторые статьи при высоком уровне цитирования могут быть связаны с другими кластерами в графе малым количеством ребер. Поэтому применение данной метрики не является исчерпывающим при определении влияния публикаций.

2. Центральность по близости (*closeness centrality*) показывает скорость распространения информации между узлами:

$$C_C(v) = \frac{|V| - 1}{\sum_{t \in V \setminus v} d_G(v, t)}, \quad (4)$$

где $d_G(v, t)$ – кратчайший путь от статьи v к вершине t . Она определяет близость отдельной публикации ко всем другим документам в базе. При этом учитывается не только наличие соседних узлов, но и состояние их связей с другими документами.

3. Центральность по посредничеству (*betweenness centrality*) вычисляется как количество кратчайших путей между всеми парами публикаций, которые проходят через заданный узел, то есть:

$$C_B(v) = \sum_{s \neq t \in V \setminus v} \frac{\sigma_{st}(v)}{\sigma_{st}}, \quad (5)$$

где σ_{st} – общее количество кратчайших маршрутов между вершинами s и t , а $\sigma_{st}(v)$ – их количество при условии прохождения через точку v . Однако, существенным недостатком этой метрики является ее вычислительная сложность.

4. Центральность по собственному вектору (*eigenvector centrality*) демонстрирует зависимость влияния узла $C_E(v)$ от значений влияния его соседей $C_E(t)$ при $t \in N(v)$. Вычисление этой метрики связано с решением системы уравнений вида:

$$C_E(v) = \frac{1}{\lambda} \sum_{t \in N(v)} C_E(t), \quad (5)$$

где λ – некоторые константы.

Кроме того, авторитетность публикаций может быть рассчитана с помощью известных методов ранжирования *PageRank* и *HITS (Hyperlink Induced Topic Search)*, которые широко применяются поисковыми системами при обработке веб-страниц. Алгоритм *PageRank* устанавливает значимость статьи путем подсчета влияния всех ссылок на нее, а метрика *HITS* предусматривает расчет для каждого узла оценок авторитетности и посредничества. Важно, что применение алгоритма *HITS* является чрезвычайно эффективным при анализе графов наукометрических баз, поскольку он позволяет выявить как цитируемые исследовательские работы, так и качественные аналитические обзоры (посредники), которые ссылаются на наиболее авторитетные узлы. Таким

образом, кроме индексов цитирования для определения влияния документов в наукометрических базах должны быть применены другие метрики, учитывающие представление статьи в различных неявных сообществах [Holme et al., 2002; Iyer et al., 2011].

3. Исследование процессов распространения информации в наукометрической базе ArXiv

ArXiv является одной из самых известных бесплатных наукометрических баз и содержит более 700 000 публикаций по вопросам физики, математики, компьютерных наук и биологии. В статье для исследования выбран фрагмент этой системы (секция физики высоких энергий), охватывающий 34 546 публикаций и 421 578 ссылок между ними. Связи со статьями, которые не входят в базу ArXiv, были изъяты из сформированного графа [Leskovec et al., 2014].

Установлено, что граф наукометрической базы содержит 61 компоненту слабой связанности (*weakly connected component*), которые имеют связь между всеми вершинами по крайней мере в одном направлении. Крупнейший из выделенных подграфов охватывает 99,58% всех статей, а все остальные компоненты являются островами, объединяющими лишь несколько публикаций. Вместе с тем, в графе выделено 21 608 компонент сильной связанности (*strongly connected component*), в пределах которых существует взаимосвязь между любыми узлами. При этом одна из них является гигантской и содержит 36,79% узлов, а размерность остальных островов не превышает десяти публикаций. То есть, большинство статей пытаются цитировать наиболее авторитетные публикации, однако обратную связь имеют лишь треть из них.

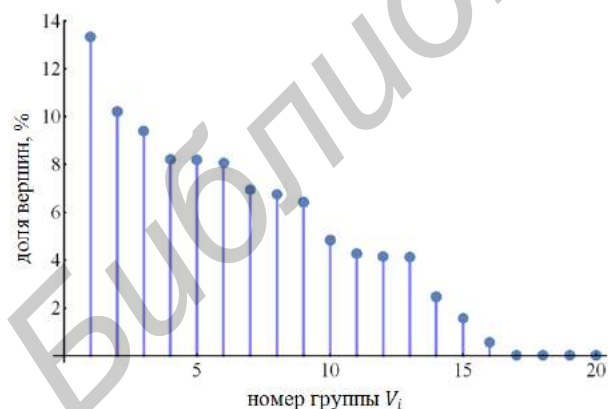


Рисунок 2 – Размерность самых крупных кластеров в наукометрической базе ArXiv

Путем оптимального разбиения графа было получено 80 неявных сообществ при значении модулярности $Q = 0,727$, что свидетельствует о существенной кластеризации. На рисунке 2 показана размерность 20 крупнейших групп. При этом лишь два из сложившихся множеств охватывают более 10% документов. На рисунке 3

приведена зависимость модулярности от значения порога цитирования публикаций, по которому выполнялась фильтрация узлов в графе. Видно, что статьи с низкими индексами цитирования, как правило, связаны только с узлами своего сообщества и не способствуют увеличению целостности системы.

Однако, изъятие из графа публикаций со средним уровнем цитирования (50 – 120 раз) приводит к стремительному увеличению модулярности, разрушению «мостов» между кластерами и отделению новых островов от гигантской компоненты. При этом наиболее цитируемые работы редко связаны между собой и при высоких уровнях порога отсечки располагаются на отдельных изолированных островах. Таким образом, основную связывающую роль в наукометрической базе выполняют статьи со средним уровнем цитирования.

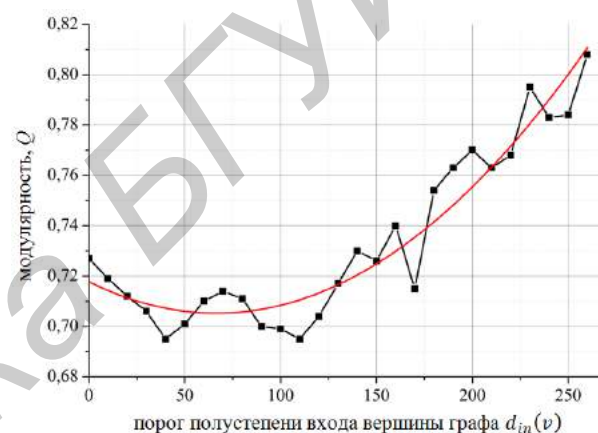


Рисунок 3 – Зависимость целостности базы ArXiv от значения порога цитирования публикаций

Рассмотрим самый большой компонент связанности графа системы ArXiv при значении порога отсечки, которое составляет 150 цитирований (рисунок 4). Видно, что высокие величины модулярности и значительная автономия сообществ приводят к недостаточности применения полустепеней входа для описания влияния публикаций (рисунок 4а). При этом мостовые узлы, изъятие которых приводит к катастрофическому распаду, имеют наименьшие показатели цитирования, а наибольшие значения метрики получают узлы, «погруженные» в глубину своих сообществ. Однако, при использовании в качестве метрики центральности по посредничеству (рисунок 4б) или по близости (рисунок 4в) влияние таких документов оказывается более высокой.

Применение алгоритма PageRank (рисунок 4г) приводит к получению наибольшего ранга статьями, связи которых, как правило, сосредоточены в пределах одного сообщества. Показательным является использование алгоритма HITS для нахождения лучших посредников, которые также содержатся внутри кластеров рядом с авторитетными исследовательскими работами

(рисунок 4д). Общая структура системы *ArXiv* для статей, цитируемых более 100 раз, показана на рисунке 5.

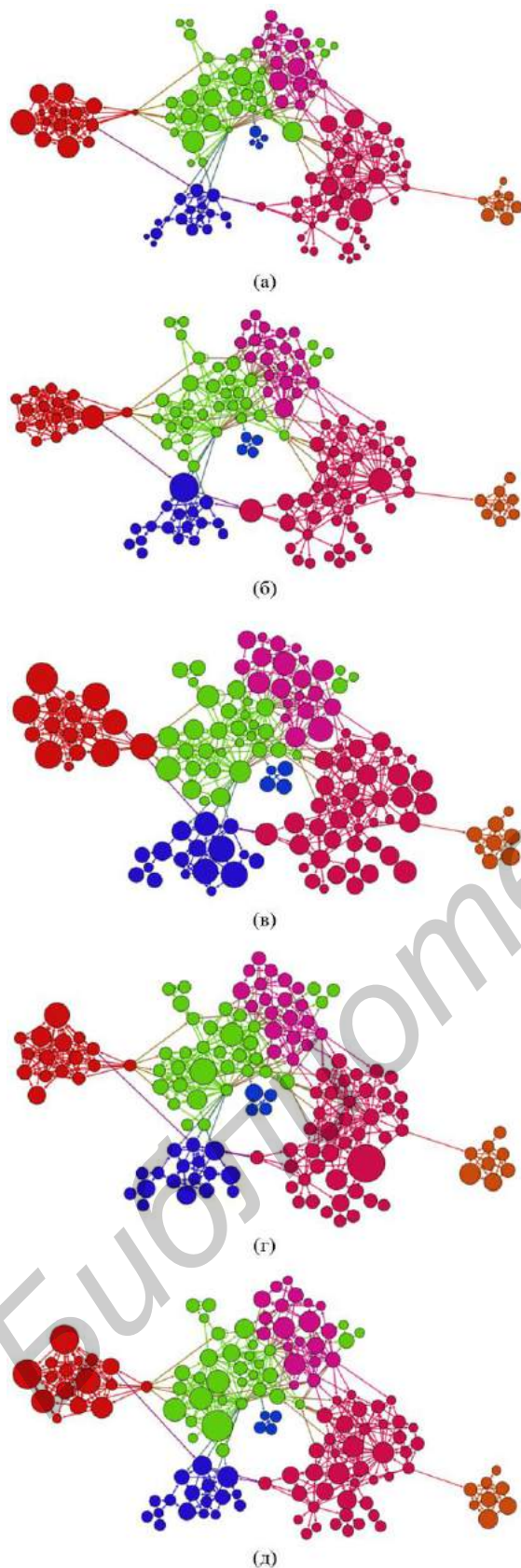


Рисунок 4 – Определение с помощью разных метрик влияния публикаций, цитированных в системе *ArXiv* более 150 раз. Цветами выделены разные сообщества в графе. Размер узлов указан пропорционально их авторитетности

В результате проведенных измерений было установлено, что значение модулярности составляет 0,937 для всемирной паутины; 0,763 для блогосферы *LiveJournal*; 0,881 для сети доверия пользователей системы шифрования PGP [Ovelgonne et al., 2010]. Кроме этого, рассчитанные значения Q для 10 случайных эго-графов социальных сетей *Facebook*, *Google+* и *Twitter* соответственно равны 0,494, 0,299 и 0,271. Таким образом, по целостности структуры база *ArXiv* уступает эго-графам, но существенно превосходит всемирную паутину. При этом система *ArXiv* и платформа *LiveJournal* имеют подобные тенденции к формированию неявных сообществ и проведению сетевой мобилизации.

Заключение

Таким образом, на примере системы *ArXiv* установлено, что структура ведущих наукометрических баз является распределенной на отдельные неявные сообщества, связи между которыми поддерживаются за счет публикаций со средним уровнем цитирования. Именно они, как правило, одновременно содержат ссылки на разные авторитетные работы и образуют между ними мостовые соединения, которые выполняют важную роль в процессе обмена информацией. Однако, использование индексов цитирования, распространенное в современных системах, не позволяет выявить такие критические узлы. Для этого более целесообразным является применение метрик центральности по посредничеству и по близости.

Библиографический список

- [Newman, 2010] Newman, M. Networks: An Introduction / M. Newman // Oxford University Press, 2010. – 720 p.
- [Newman, 2006] Newman, M. Modularity and community structure in networks / M. Newman // Proceedings of the National Academy of Sciences. – 2006. – № 103(23). – P. 8577 – 8582.
- [Fortunato, 2010] Fortunato, S. Community detection in graph / S. Fortunato // Physics Reports. Elsevier. – 2010. – № 486. – P. 75 – 174. DOI: 10.1016/j.physrep.2009.11.002.
- [Potebnia et al., 2015] Potebnia, A. Innovative GPU accelerated algorithm for fast minimum convex hulls computation / A. Potebnia, S. Pogorilyy // Proceedings of the 2015 Federated Conference on Computer Science and Information Systems. – 2015. – P. 555 – 561. DOI: 10.15439/2015F305.
- [Lancichinetti et al., 2011] Lancichinetti, A. Finding Statistically Significant Communities in Networks / A. Lancichinetti, F. Radicchi, J. Ramasco, S. Fortunato // PLoS ONE. – 2011. – № 6(4): e18961. DOI: 10.1371/journal.pone.0018961.
- [Holme et al., 2002] Holme, P. Attack vulnerability of complex networks / P. Holme, B. Kim, C. Yoon, S. Han // Physical Review E 65: 056109. – 2002. DOI: 10.1103/PhysRevE.65.056109.
- [Iyer et al., 2011] Iyer, S. Attack Robustness and Centrality of Complex Networks / S. Iyer, T. Killigback, B. Sundaram, Z. Wang // PLoS ONE. – 2013. – № 8(4): e59613. DOI: 10.1371/journal.pone.0059613.
- [Leskovec et al., 2014] Leskovec, J. SNAP Datasets: Stanford Large Network Dataset Collection. – June 2014. URL: <http://snap.stanford.edu/data>.
- [Ovelgonne et al., 2010] Ovelgonne, M. Randomized greedy modularity optimization for group detection in huge social networks / M. Ovelgonne, A. Geyer-Schulz, M. Stein // Proceedings of the 4th Workshop on Social Network Mining and Analysis. ACM, New York. – 2010.

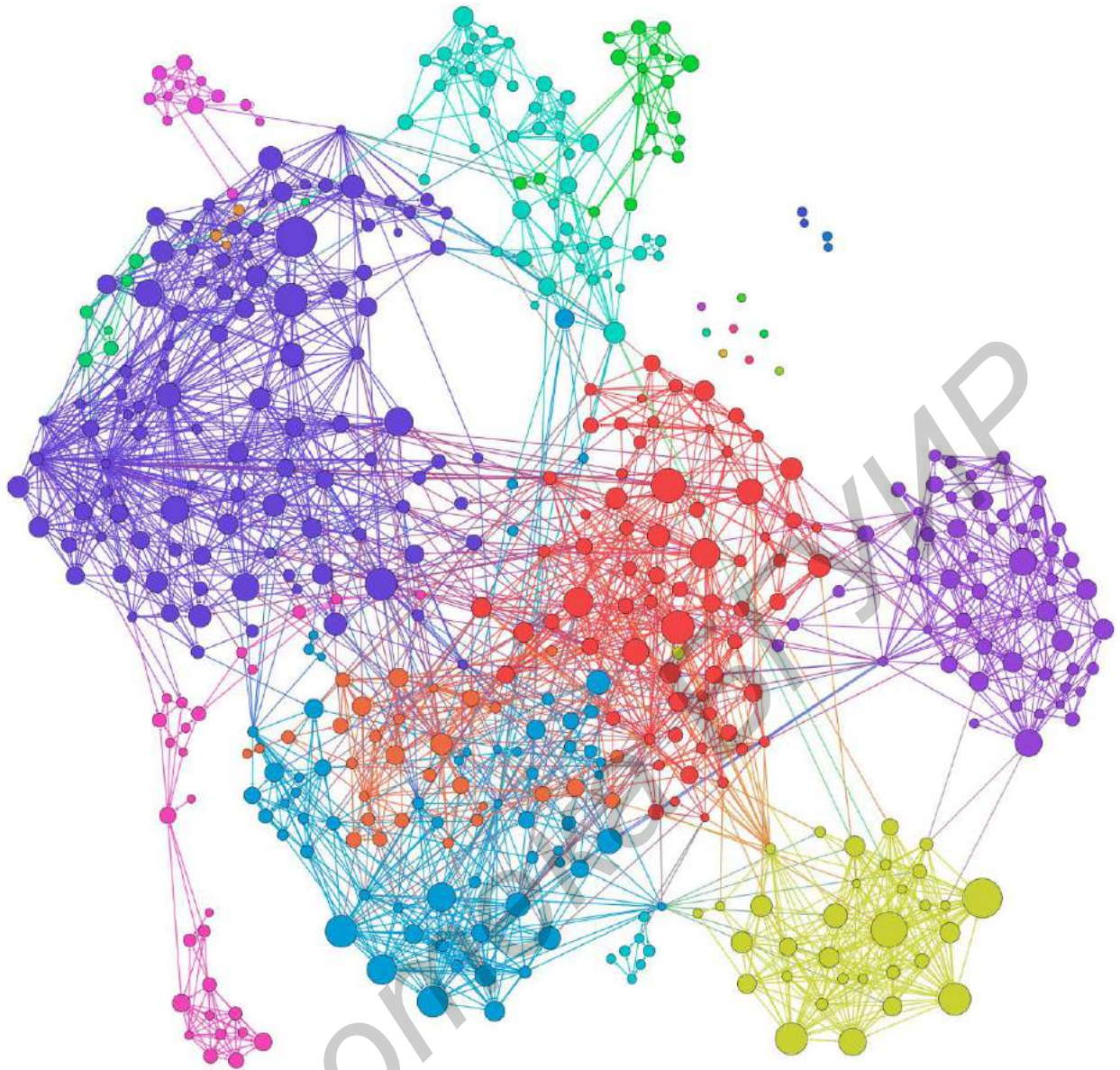


Рисунок 5 – Структура графа наукометрической базы ArXiv для публикаций, которые имеют более 100 ссылок. Размер узлов указан пропорционально индексу их цитирования

ANALYSIS OF INFORMATION EXCHANGE PROCESSES IN SCIENTOMETRIC DATABASES

Potebnia A.V., Pogorilyy S.D.

*Kyiv National Taras Shevchenko University, Kyiv,
Ukraine*

admin@artem.bz.ua

sdp@univ.net.ua

This paper presents the investigation of information propagation processes in scientometric databases. On the basis of the developed mathematical model with the application of the combinatorial optimization methods, we have identified the trends of databases clustering. The paper contains the experimental research of the ArXiv system. As a result, we have obtained the following important results and conclusions:

- The database structure is divided into a number of implicit document communities, which have a relatively low density of the intergroup connections;
- The primary linking function between these communities is performed by the articles of the average citing level;
- The usage of citation indexes is not enough to identify the critical bridge nodes. For this purpose the calculation of the closeness and betweenness centrality metrics is much more appropriate;
- ArXiv system is more vulnerable than the ego-graphs of social networks, but its structural integrity is larger in comparison to the *World Wide Web*;
- In addition, the scientometric base ArXiv and blogosphere LiveJournal have the similar trends of the implicit communities formation and the network mobilization.

Keywords: scientometric database; graph partition problem; modularity function.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822: 004.912

РАСПОЗНАВАНИЕ ИНФОРМАЦИОННЫХ ОПЕРАЦИЙ: МУЛЬТИАГЕНТНЫЙ ПОДХОД

Додонов А.Г., Ландэ Д.В., В.А. Додонов

*Институт проблем регистрации информации НАН Украины,
г. Киев, Украина*

dodonov@ipri.kiev.ua

dwlande@gmail.com

dodonov.vadim@gmail.com

В работе предложено в качестве методологической основы детектирования информационных операций исследование динамики информационных потоков. Предложена модификация модели диффузии информации, модель информационной резервации. Рассмотрены мультиагентные модели распространения информации, позволяющие распознавать информационные операции. Определены достоинства и недостатки данных моделей.

Ключевые слова: информационные сюжеты, информационные операции, информационное пространство, моделирование, мультиагентные системы.

Введение

Информационная операция, термин которой в последнее время применяется все шире, является компонентой информационной войны, содержание которой направлено на реализацию предварительно спланированных психологических воздействий на враждебную, дружескую или нейтральную аудиторию путем информационного влияния на установки и поведение с целью достижения заранее определенных преимуществ [DoD, 2003]. Информационные операции определяются как «акции, направленные на воздействие на информацию и информационные системы противника и защиту собственной информации и информационных систем». Проявления информационных операций встречаются во многих сферах – военной, социальной, экономической. Информационные операции в настоящее время непосредственно связаны с воздействием на людей, манипулированием.

При изучении информационных операций необходимо определить объективные критерии, и в качестве одного таких, можно рассматривать динамику распространения информационных сюжетов в соответствующем фрагменте информационного пространства. Исследованию динамики информационных потоков посвящены многочисленные научные работы [Corso, 2005], [Kleinberg, 2006], [Ландэ, 2006], [Rakesh, 2014]

показано, что в типовых ситуациях динамике распространения новостей, информационного сюжета присущ характер «всплеска», волны с явным периодом возрастания его влияния и плавным спадом.

В результате анализа многочисленных диаграмм поведения ТИП, были выявлены наиболее типичные, базовые профили их поведения (рис. 1) [Ландэ, 2012]. Некоторые сюжеты развиваются следующим образом: после быстрого информационного всплеска подготовки идет плавный спад (например, публикации о стихийных бедствиях), некоторые, напротив предполагают длительную плавную информационную подготовку, после чего идет резкий спад (например, публикации об планируемых заранее мероприятиях). Существуют также тематические потоки, характеризующиеся симметричной кривой динамики, как узкие, кратковременные, так и растянутые во времени.

В случае информационных потоков, которые ассоциируются с конкретными тематическими информационными потоками, необходимо описывать динамику каждого из таких потоков отдельно, принимая во внимание то, что рост одного из них может автоматически приводить к уменьшению других и наоборот. Поэтому ограничение на объемы информации по всем тематикам распространяется и на совокупность всех информационных сюжетов. В случае изучения

общего информационного потока наблюдается явление «перетекания» объемов публикаций из одних, теряющих актуальность информационных сюжетов, в другие.

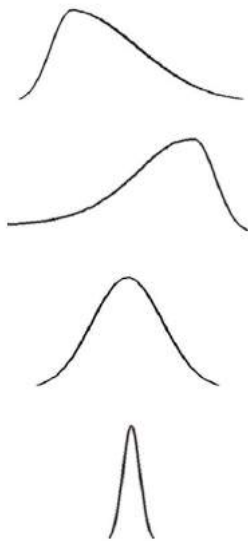


Рисунок 1 – Базовые профили динамики тематических сюжетов новостей

Следует отметить, что выбор тематики информационных сюжетов, позволяющих детектировать информационные операции, является содержательно сложной, неоднозначной задачей.

1. Модель диффузии информации

Жизненный цикл тематических информационных потоков (информационных сюжетов) может описываться, например, моделью диффузии информации (МДИ), построенной с помощью методов клеточных автоматов [Ландз, 2007].

Клеточные автоматы являются полезными дискретными моделями для исследования динамических систем. Дискретность модели, а точнее, возможность представить модель в дискретной форме, может считаться важным преимуществом, поскольку открывает широкие возможности использования компьютерных технологий.

Модель диффузии информации является двумерной, поэтому вся система клеточных автоматов для этого случая будет описываться двумерным массивом. В случае двумерной решетки, элементами которой являются квадраты, ближайшими соседями, входящими в окрестность элемента, можно считать или только элементы, расположенные вверх-вниз и влево-вправо от него, либо добавленные к ним еще и диагональные элементы (окрестность Мура).

В рамках данной модели, которая относится к распространению новостей в информационном пространстве, применяются окрестность Мура и вероятностные правила распространения новостей по заданной тематике.

В рамках МДИ, которая относится к распространению информационных сюжетов в информационном пространстве, применяются вероятностные правила распространения новостей по заданной тематике. Предполагается, что каждая клетка клеточного автомата может иметь различные статусы информированности, а именно, быть в одном из трех состояний: 1 – «свежая новость» (клетка окрашивается в черный цвет); 2 – новость, устаревшая, но сохраненная в виде сведений (серая клетка); 3 – клетка не имеет информации, переданной новостным сообщением (клетка белая, информация не дошла или уже забыта).

МДИ предполагает следующие правила развития информационного сюжета:

- 1) изначально все поле состоит из белых клеток за исключением нескольких стоящих рядом черных, которые первыми «приняли» новость;
- 2) белая клетка может перекрашиваться только в черный цвет или оставаться белой (она может получать новость или оставаться «в неведении»);
- 3) белая клетка перекрашивается, если выполняется условие: $pt > 1$, где p – псевдослучайная величина ($0 < p < 1$), t – количество черных клеток в окрестности;
- 4) если клетка черная, а вокруг нее черные и серые ($s > x$, s – количество черных и серых клеток, x – заданная константа, то она перекрашивается в серый цвет (новость устаревает, но сохраняется как сведения);
- 5) если клетка серая, а вокруг нее исключительно черные и серые ($s > y$, y – заданная константа), то она перекрашивается в белый цвет (забывание сведений при их общеизвестности).

Типичные состояния системы клеточных автоматов, пребывающих в различных состояниях, в зависимости от шагов итерации приведены на рис. 2.

При анализе приведенных графиков следует обратить внимание на такие особенности: 1 – суммарное количество клеток, пребывающих во всех трех состояниях на каждом шагу итерации постоянно и равно размеру поля; 2 – при стабилизации клеточных автоматов соотношение количества серых, белых и черных клеток приблизительно составляет: 0.75 : 0.25 : 0; существует точка пересечения кривых, определяемых всеми тремя последовательностями на уровне 33 % каждая.

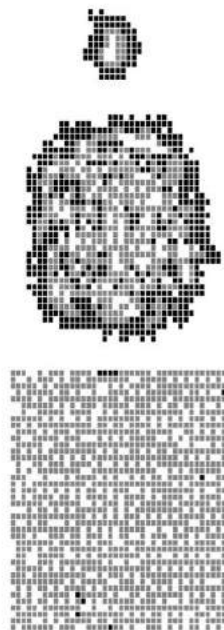


Рисунок 2 – Состояния эволюции системы клеточных автоматов

Полученные в результате аналитического моделирования зависимости количества серых x_g , белых x_w и черных x_b клеток от шага эволюции системы клеточных автоматов, выражаются формулами:

$$x_g = \frac{1}{1 + e^{-\alpha(t-\tau_1)}}; \quad (1)$$

$$x_w = 1 - \frac{1}{1 + e^{-\beta(t-\tau_2)}};$$

$$x_b = \frac{1}{1 + e^{-\beta(t-\tau_2)}} - \frac{1}{1 + e^{-\alpha(t-\tau_1)}}. \quad (2)$$

Базовые профили динамики информационных сюжетов, соответствующие значениям $x = y = 8$ в правилах 4 и 5 модели, были получены при значениях параметров $\alpha = 0,15$, $\beta = 0,25$.

2. Модель информационной резервации

«Информационную резервацию» можно также охарактеризовать как область информационного пространства, находящуюся под непрерывным воздействием информационных операций. В результате этого динамике важнейших информационных сюжетов, протекающих в ИР, свойственны отклонения от характера плавного «всплеска», а именно:

- быстрое прекращение «нежелательного» информационного сюжета (*S*-эффект);
- растягивание периода подъема информационного сюжета (*L*-эффект) с «угодной» администрации ИР тематикой.

Указанные отклонения получаются в том случае, когда изменяются параметры правил,

определяющих поведение модели диффузии информации, соответствующие некоторым жизненным наблюдениям за информационными резервациями. Если сопоставить черным клеткам модели (сообщение активно) нахождение сообщения в оперативной памяти, а серым – нахождение сообщения в архивной памяти, то *S*- или *L*-эффектам будет соответствовать соотношение времен нахождения сообщения в оперативной или архивной памяти, что регулируется параметрами x и y правил 4 и 5. Как уже было отмечено, при значениях параметров $x = y = 8$ модель соответствует естественной динамике развития информационного сообщения вне информационной резервации – ее график принимает вид колоколообразной кривой. При сохранении параметра оперативной памяти ($x = 8$) и уменьшении параметра архивной памяти y , (до $y = 2$), чаще освобождаемые от информации серые ячейки, затем интенсивней принимают ранее забытую информацию, прекращиваясь в результате в черный цвет, т.е. происходит эффект «проталкивания» нового сообщения (соответствующего «политике» информационной резервации) – *L*-эффект.

С другой стороны, при сохранении параметра оперативной памяти ($x = 8$) и уменьшении параметра архивной памяти x (до $x = 2$) происходит быстрое «забывание» не соответствующего информационной резервации сообщения и перевод его основной части в архив – *S*-эффект.

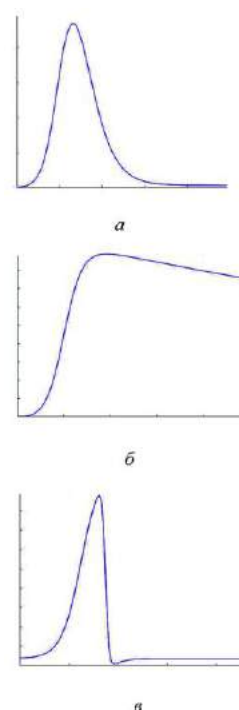


Рисунок 3 – Динамика количества клеток в состоянии «свежая новость»: а – типовая динамика ($\alpha = 0,15$, $\beta = 0,25$); б – растягивание периода актуальности информации ($\alpha = 0,01$, $\beta = 0,25$); в – немедленное прекращение распространения информации ($\alpha = 0,15$, $\beta = 1,5$)

Обобщение аналитической интерпретации МДИ, связанное с изменением коэффициентов в приведенных выше уравнениях (связанных с восприятием информации), которое приводит к искажению типовой динамики, что можно объяснять наличием эффекта информационной резервации (рис. 3).

3. Мультиагентная модель распространения информации

Для создания правдоподобной виртуальной модели распространения информации, прежде всего, необходимо сформировать приближенное к реальности виртуальное информационное пространство, населенное виртуальными информационными агентами. Основной задачей компьютерного моделирования является создание информационного виртуального мира (модели информационного пространства), населенного индивидуальными информационными агентами.

Одной из основных проблем моделирования информационного пространства является проблема описания взаимодействия информационных агентов. Каким образом происходит формирование социумов агентов? Что влияет на отношение агента к той или иной информации? Каковы динамика падения и восстановления информационной репутации, величина информационного следа (остаточная мощность), критическая масса информационного влияния.

Как предположение, можно рассмотреть возможность сравнения информационного воздействия агентов с моделью распространения вирусов, как простейший случай. Информация как вирус может распространяться в общем случае:

- от агента агенту;
- от агента группе (социальные сети, вербальный обмен и др.);
- от информационных центров (производства информации) к отдельному агенту или к группе агентов.

Рассмотрим предложенную авторами мультиагентную модель распространения информации в информационном пространстве. Базовые профили динамики тематических информационных потоков (ТИП) были получены с помощью предложенной авторами мультиагентной модели, в рамках которой отдельные документы, образующие информационный сюжет, ассоциируются с агентами, жизненный цикл агентов – с жизненным циклом документов в информационном пространстве. Предполагается, что в течение времени происходит эволюция популяции агентов, т.е. отдельные агенты могут (рис. 4):

- 1) самозародиться (родиться по причинам, возникающим вне рассматриваемого мультиагентного пространства);
- 2) породить новых агентов;

- 3) «умирать» – исчезать из пространства агентов;
- 4) получать ссылки от других агентов.

Каждый агент обладает «потенциалом», зависящим от его возраста (времени жизни на текущий момент), авторитетности (ссылок, предоставленных на него) и плодовитости (количества порожденных непосредственно им агентов).

Управляющие параметры модели следующие:

- 1) вероятность «самозарождения» P_1 ;
- 2) потенциал агента Pot , зависящий от количества ссылок на него (ns), времени его жизни (t), и количества порожденных им агентов (k):

$$Pot = \frac{ns + k}{t};$$
- 3) вероятность «рождения» от существующего: $P_2 \cdot Pot$;
- 4) вероятность «смерти» агента: P_3 / Pot ;
- 5) вероятность ссылки на агента: $P_4 \cdot Pot$.

Варьирование параметрами управления P_1 , P_2 , P_3 и P_4 позволили смоделировать профили поведения информационных сюжетов. На рис. 2 приведена пример возможной динамики мультиагентной системы: процессы рождения новых агентов от существующих обозначены сплошными стрелками, процессы проставления ссылок на агентов представлены пунктирными стрелками, живые агенты – черными кругами, «мертвые» агенты к моменту $t = 5$ – незаполненными окружностями.

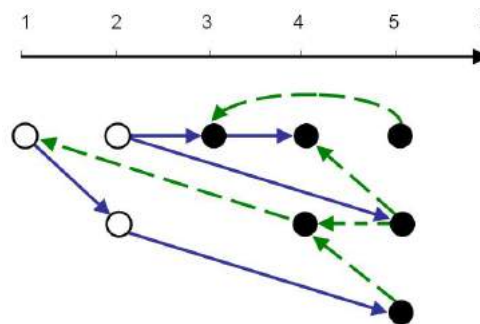


Рисунок 4 – Фрагмент мультиагентного пространства

В предложенной модели учитывается общеизвестная практика проведения информационных кампаний в социальных сетях, заключающаяся в регистрации большого числа аккаунтов-роботов (роя), от имени которых проставляются ссылки (лайки) на материалы, публикуемые от имени аккаунтов из того же роя и на целевые информационные страницы – документы.

В результате проведенных исследований была реализована программа эволюции пространства агентов, исследована эволюция мультиагентной системы при различных значениях параметров, найдены аналогии с реальными тематическими информационными потоками.

4. Модель динамики информационных операций

Предполагается, что системное нарушение типовой динамики некоторых информационных сюжетов в открытом информационном пространстве может свидетельствовать как об информационных операциях [Горбулин, 2009], так и о наличии существовании информационной резервации. При исследовании информационных операций также большое внимание уделяется анализу динамики информационных сюжетов [Додонов, 2013], пользоваться доступными аналитическими средствами, например, вейвлет-анализом. При этом следует ориентироваться на возможные модели информационных атак, например, если эта модель охватывает фазы: «фоновые публикации» — «затишье» — «артподготовка» — «затишье» — «атака», то уже по первым трем компонентам можно с большой вероятностью предсказать будущие события.

Объединяя участки графика, соответствующие началу информационной операции и трендам, связанным с инновационной деятельности (внедрение инноваций можно условно также считать информационными операциями), был получен полный график, соответствующий отображению информационных операций в информационном пространстве (рис. 5). При этом следует ориентироваться на возможные модели информационных атак, однако уже по первым трем компонентам (фазам) можно с большой вероятностью предсказать будущие события.

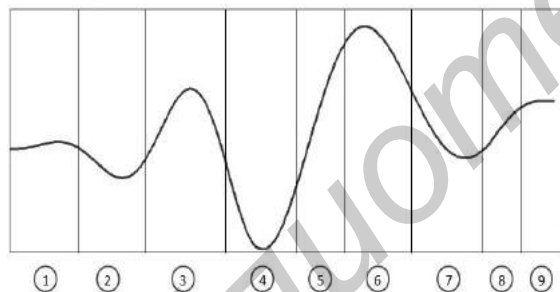


Рисунок 5 – Обобщенная диаграмма, соответствующая всем этапам жизненного цикла информационных операций: 1 – фон; 2 – затишье; 3 – «артподготовка»; 4 – затишье; 5 – атака/триггер роста; 6 – пик завышенных ожиданий; 7 – утрата иллюзий; 8 – общественное осознание; 9 – продуктивность/фон

Отметим, что предложенная модель позволяет отличать информационные потоки, поведение которых определяется естественными закономерностями информационного пространства, от потоков, вызванных влиянием внешних факторов. В частности, в случае информационных резерваций, в качестве индикатора может рассматриваться отклонение трендов динамики некоторых информационных сюжетов от характерных форм распределения, появление периодических зон нестабильности значений, или, наоборот, удивительная локальная стабильность этих значений.

Предложенные модели соответствуют реальным данным, которые экстрагируются системами контент-мониторинга. Таким образом, приведенные зависимости могут быть использованы в качестве шаблонов для выявления информационных операций – как путем анализа ретроспективного фонда сетевых публикаций, так и путем оперативного мониторинга появления некоторых их признаков в реальном времени.

Достоинством таких «шаблонных методов», является простота их реализации (дисперсионные методы, в частности, DFA, методы регрессионного анализа, вейвлет-анализ и т.д.) Методы хорошо приспособлены к ретроспективному анализу. Главный недостаток методов заключается в том, что с их помощью информационную операцию можно распознать лишь на ее финальной стадии.

5. Мультиагентная модель, учитывающая структуру сети распространения информации

Как расширение приведенной выше мультиагентной модели распространения информации можно рассматривать модель, в которой учитывается структура формируемой сети [Пугачев, 2015]. В рамках этой модели каждый агент – источник информации обладает не «потенциалом», а некоторым рейтингом (которому на схемах соответствует размер соответствующего узла). Связями в рассматриваемой сети являются факты перепечатки или «пересказа».

В основе модели лежит предположение, что при проведении информационных операций наиболее рейтинговые источники перепечатывают информацию у наименее рейтинговых, или образуются кластеры низкорейтинговых изданий, перепечатывающих одну и ту же новость.

На рис. 6. приведен пример типичных информационных операций, выявляемых в рамках данной модели.

В рамках формализации этой же модели выбирается несколько десятков параметров топологии сетей распространения информации, таких как диаметр, плотность, кластеризация, посредничество и т.п., которые сравниваются с некоторыми эталонными значениями.

К достоинствам данной модели следует отнести ее формальную строгость и соответствие активно развивающемуся в последнее время направлению Complex Networks, что позволяет ожидать ее дальнейшего развития. К недостаткам следует, по видимому, отнести малую корреляцию с содержательной стороной распознаваемых информационных операций, а также определенную вычислительную сложность при выявлении нечетких информационных дубликатов документов.

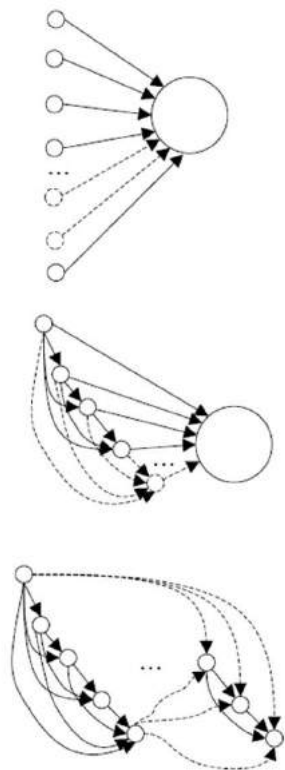


Рисунок 6. – Примеры сетей распространения информации, имеющих признаки информационных операций. Источник – [Пугачев, 2015]

Заключение

Таким образом, в работе в качестве методологической основы детектирования информационных операций предложено исследование динамики информационных потоков.

Рассмотрены некоторые модели информационных потоков, предложена модификация модели диффузии информации, приводящая к типовым профилям динамики информационных потоков в информационных резервуарах, детально рассмотрены мультиагентные модели распространения информации, позволяющие распознавать возможные информационные операции. Определены достоинства и недостатки данных моделей.

Библиографический список

- [Corso, 2005] G.M. Del Corso, A. Gulli, F. Romani. Ranking a stream of news. In Proceedings of 14th International World Wide Web Conference, pp. 97-106, Chiba, Japan, 2005.
- [DoD, 2003] Information operations roadmap. – DoD US. – Washington, D.C.: GPO, 2003.
- [Kleinberg, 2006] Kleinberg J. Temporal dynamics of on-line information streams // Data Stream Management: Processing High-Speed Data Streams. – Springer, 2006.
- [Rakesh, 2014] Rakesh V., Singh D., Vinzamuri B., Reddy C.K. Personalized Recommendation of Twitter Lists Using Content and Network Information // Association for the Advancement of Artificial Intelligence (Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media, 2014.
- [Горбулін, 2009] Горбулін В.П., Додонов О.Г., Ланде Д.В. Інформаційні операції та безпека суспільства: загрози, протидія, моделювання: монографія. – К.: Інтертехнологія, 2009. – 164 с.
- [Додонов, 2013] Додонов А.Г., Ланде Д.В. Мультиагентная модель поведения тематических информационных потоков //

Материалы VI Всероссийской мультиконференции по проблемам управления (30 сентября – 5 октября 2013 г.) – Т. 4. – Ростов-на-Дону: Издательство Южного федерального университета, 2013. – С. 102-107.

[Ланде, 2006] Ланде Д.В., Фурашев В.Н., Брайчевский С.М., Григорьев А.Н. Основы моделирования и оценки электронных информационных потоков: Монография. – К.: Инжиниринг, 2006. – 176 с.

[Ланде, 2007] Ланде Д.В. Модель диффузии информации // Информационные технологии и безопасность. Менеджмент информационной безопасности. Сборник научных трудов Института проблем регистрации информации. – Вып. 10. – 2007. – С. 51-67.

[Ланде, 2012] Ланде Д.В., Брайчевский С.М. Моделирование поведения тематических сюжетов новостей в веб-пространстве // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012 (16-20 октября 2012 г., г. Белгород, Россия): Труды конференции. – Т. 1. – Белгород: Изд-во БГТУ, 2012. – С. 197-204.

[Потемкин, 2015] Потемкин А.В. Выявление информационных операций в средствах массовой информации сети Интернет: диссертация ... кандидата технических наук: 05.13.10 / Потемкин Алексей Владимирович; [Место защиты: Брянский государственный технический университет]. – Брянск, 2015. – 144 с.

RECOGNITION OF INFORMATION OPERATIONS: MULTI-AGENT APPROACH

Dodonov A.G., Lande D.V., V.A. Dodonov

*Institute for Information Recording NAS of
Ukraine, Kiev, Ukraine*

dodonov@ipri.kiev.ua

dwlande@gmail.com

dodonov.vadim@gmail.com

In work research of dynamics of information streams is offered as a methodological basis of detecting of information operations. Modification of model of diffusion of information, model of information reservation is offered. The multi-agent models of dissemination of information, which allows to recognize information operations, are considered. Advantages and disadvantages of these models are defined.

Keywords: *news stories, information operations, information space, modeling, multi-agent systems*



УДК 004.896

МОДЕЛЬ КООПЕРАЦИИ АГЕНТОВ В СЕМИОТИЧЕСКОЙ СРЕДЕ

Кулинич А.А.

*Федеральное государственное бюджетное учреждение науки Институт проблем управления им.
В.А. Трапезникова Российской академии наук,
г. Москва, Россия*

kulinich@ipu.ru

В работе рассмотрена модель кооперации агентов, представленных в знаковой среде их функционирования. Предложена модель формального агента с BDI архитектурой, описанная в терминах знаковой среды. Сформулированы условия кооперации агентов в условиях неполноты их знаний о среде функционирования.

Ключевые слова: Многоагентная система; знаковая среда функционирования; BDI архитектура; концептуальный каркас; кооперация агентов.

Введение

Теоретические основы кооперативного взаимодействия агентов были заложены в 60-е годы прошлого столетия в работе [Цетлин, 1969; Стефанюк и др., 1967; Варшавский и др., 1984]. В основе этих исследований лежит изучение коллективного поведения конечных автоматов, моделирующие агентов с простой реактивной архитектурой, реагирующих на изменение среды, в которой они находятся. С помощью простых реактивных агентов реализуется, как правило, несложное командное поведение агентов – это образование стаи или роя агентов [Карпов, 2014]. Исследуются разнообразные алгоритмы обхода стаей (роем) агентов препятствий, уклонение от столкновения [Павловский, 2002] и т.д. Однако, агенты с реактивной архитектурой могут решать и более сложные задачи, если определены правила их поведения в среде и модели среды их функционирования. Так, например, в работе [Кулинич, 2014] представлена математическая модель командной работы агентов на основе модели поведения людей в малых социальных группах. В рамках этой модели, на основе критериев полезности агентов и взаимного диссонанса, разработаны правила поведения агентов в среде, отражающие закономерности игры в виртуальный футбол, позволяющие моделировать и исследовать командное и эгоистичное поведение агентов.

Для моделирования более сложного и интеллектуального поведения агентов в рамках теории многоагентных систем был предложен ряд теоретических концепций, определяемых как

теории командной работы агентов. Это теория общих намерений [Cohen, 1991] и теория общих планов [Grosz, 1996], ориентированные на BDI (*Belief-Desire-Intention*) архитектуру агентов [Rao, 1995]. В рамках BDI архитектуры описываются «ментальные» состояния агентов в терминах их убеждений (*Belief*), желаний (целей) (*Desire*) и намерений (*Intention*). Задача теорий командной работы заключается в том, чтобы построить методы согласования убеждений, желаний множества агентов, а также их намерений для реализации совместной работы. Такое согласование требует многочисленных коммуникаций между агентами.

Важность коммуникации между агентами при их коллективном решении сложных задач, привело к появлению множества подходов (языков и протоколов коммуникации), основанных на различных принципах. В этих подходах важным элементом, для обеспечения взаимопонимания между агентами являются модели их знаний о предметной области - онтологии. Одинаковые онтологии предметной области у разных агентов позволяет им общаться с помощью знаков, которые, по сути, являются именами понятий их понятийных систем. Знак в онтологии представлен тройкой: имя, смысл, представление (значение) знака. Если у агентов одинаковые онтологии, то при коммуникации достаточно передать агенту только имя знака, которое однозначно определит его смысл, представление и возможную реакцию - действия.

В этой статье исследуются вопросы коммуникации агентов (роботов), которые работают в условиях неопределенности среды их функционирования.

1. Знаковая среда функционирования агентов

Обычно, предполагается, что агенты, выполняя некоторую совместную работу, могут общаться не только между собой (искусственными агентами), и с людьми. В этом случае, онтологии искусственных агентов и людей должны совпадать. Однако, в случаях, когда агенты решают задачу автономно, без общения с людьми, язык их коммуникации может быть значительно упрощен, оставаясь при этом достаточно выразительным для решения агентами поставленной задачи. Упростить коммуникации между агентами можно, если сделать допущения о среде функционирования агентов.

Под средой функционирования агентов будем понимать «среду обитания» агентов, которая включает множество агентов и объектов, обладающие определенными свойствами (параметры агентов и объектов), автономностью, реактивностью (способностью реагировать на изменение состояния среды), проактивностью (способностью действовать самостоятельно для достижения своей цели), коммуникабельностью (способностью к общению и совместным действиям для достижения цели).

Дадим формальное определение среды функционирования агентов. Считаем, что среда функционирования включает множество объектов $B = \{b_j\}$ и агентов $A = \{a_i\}$ и пусть известны множества свойств $F = \{f_i\}$ каждого из объектов этой среды. Пусть для каждого свойства каждого объекта или агента известны множество их возможных значений, $Z = \{Z_i\}$. Считаем, что множество значений свойства – это упорядоченное множество, т.е. $Z_i = \{z_{i1}, \dots, z_{iq}\}$, $z_{iq+1} \succ z_{iq}$, $q = 0 \dots n-1$. Тогда, среду функционирования определим как гиперкуб, полученный прямым произведением множеств значений всех свойств объектов и агентов, $SF = \times_i Z_i$.

При таком определении среды функционирования агентов, каждый ее объект и агенты могут быть представлены в виде точек с координатами – значениями их свойств в пространстве SF , т.е. $A, B = (z_{1e}, \dots, z_{nq})$, где $(z_{1e}, \dots, z_{nq}) \in SF$.

Состоянием среды функционирования будем называть вектор значений всех свойств объектов среды в некоторый момент времени t :

$$Z(t) = (z_{1e}, \dots, z_{1q}, \dots, z_{ne}, \dots, z_{nq}). \quad (1)$$

Считаем, что агенты могут изменять значения своих свойств и значения свойств объектов и других агентов, что является проявлением их реактивных и проактивных способностей.

Среду функционирования агентов будем интерпретировать как семантическое пространство, которое считается признаковой моделью знаний. В работе [Кулинич, 2014] было показано, что в семантическом пространстве можно выделить множество подпространств $\{SS(d^H)\}$, $SS(d^H) \subset SF$,

удовлетворяющее свойствам рефлексивности, антисимметричности и транзитивности. Они образуют решетку $KK = (\{SS(d^H)\}, \cap, \cup)$, структурирующую это семантическое пространство. Такая структуризация семантического пространства называется качественным концептуальным каркасом онтологии и отражает идеализированную структуру знаний об этой среде [Кулинич, 2014]. Все подпространства $SS(d^H)$ в данном случае интерпретируются как классы состояний среды функционирования и характеризуются тройкой: d^H – имя класса; $SS(d^H)$ – содержание класса и $V(d^H)$ – объем класса – это множество объектов среды, значения свойств которых попадает в подпространство $SS(d^H)$. Мощность множества подпространств, включенных в концептуальный каркас, определится как 3^N , где N – общее количество всех свойств агентов и объектов среды функционирования.

Известные закономерности среды функционирования могут быть заданы в виде множества правил, связывающих значения свойств разных объектов этой предметной области и могут быть представлены в виде отображения:

$$W: \times_i Z_i \rightarrow \times_i Z_i. \quad (2)$$

Среду функционирования определим кортежем:

$$\langle A \cup B, Z(t), KK \rangle, \quad (3)$$

где B – множество имен объектов и A – агентов, $Z(t)$ – вектор состояний среды функционирования момент времени t , KK – концептуальный каркас среды функционирования.

В работе [Кулинич, 2014] было определено соответствие φ вектора значений $Z(t)$ и класса состояний среды функционирования, заданных в концептуальном каркасе:

$$\varphi: Z(t) \rightarrow SS(d^H) \subseteq KK, \quad (4)$$

Соответствие (4) позволяет интерпретировать множество векторов $\{Z(t)\}$, $t \in \{1, \dots, n\}$, значения которых попадают в подпространство $SS(d^H)$ как знаковое (понятийное) состояние среды функционирования.

Под знаком в этом случае понимается тройка: $(d^H, SS(d^H), V(d^H))$, где d^H – имя класса состояний; $SS(d^H)$ – содержание класса (определяет подпространство среды функционирования $SS(d^H) \subseteq \{SS(d^H)\}$) и $V(d^H) = \{a_i, b_j\}$, $a_i \in A$, $b_j \in B$, – объем класса – это множество объектов и агентов значения признаков, которых принадлежат подпространству $SS(d^H)$.

Состоянием среды функционирования агентов в терминах знаков концептуального каркаса среды функционирования будем называть вектор имен классов состояний концептуального каркаса, к

которым принадлежат значения свойств объектов и агентов ($A \cup B$):

$$SF(t)=(d_1^H; d_2^H; \dots; d_n^H), H=1, \dots, 3^N. \quad (5)$$

Например, имя класса d_1^1 в позиции первого агента однозначно определяет, содержание класса - подпространство $SS(d^1)$, и объем $V(d^1) = a_1$, $a_1 \in A$, агента значения признаков которого принадлежат подпространству $(z_{1j_1}, \dots, z_{1j_q}) \in SS(d^1)$.

2. Знаковая BDI архитектура агентов

В терминах предложенной модели среды функционирования представим элементы *BDI* (*Belief-Desire-Intention*) архитектуры агентов, т.е. их убеждения (знания о среде функционирования), желания (цели) и намерения (действия).

Убеждения агентов (*Belief*) – это его знания о среде функционирования, включающие две составляющие. Первая – это знания агента о среде функционирования в виде частично упорядоченного множества классов состояний среды функционирования, которые доступны для этого агента, которые формально представляются в виде:

$$BEL_i = \{SS_i(d^H), \leq\} \subseteq KK^W.$$

Вторая составляющая убеждений – это знания о закономерностях среды функционирования, которые ранее определялись как отображение $W(2)$, записанное для векторов значений признаков $\times Z_i$.

Это отображение можно переписать в терминах классов состояний среды функционирования с учетом ранее определенного соответствия (4) в следующем виде:

$$W_j^{BEL}: \varphi(\times Z_{ji}) \rightarrow \varphi(\times Z_{ji}), \quad (6)$$

где $\varphi(\times Z_{ji}) = SS_{ji}(d^H) \in BEL_j$

Отображение (6) может быть представлено множеством правил продукций, отражающих закономерности среды функционирования.

Цели агентов (*Desire*) выражаются целевым состоянием среды функционирования в виде вектора $G_i = (g_{i1}, \dots, g_{in})$, $g_{in} \in Z_i$. По сути, цель каждого агента – это точка в пространстве *FS* и она может быть охарактеризована классом состояний среды функционирования, к которому эта цель принадлежит. В этом случае цель определяется как знак, т.е. тройкой: d^{Gi} – имя класса состояний цели; $SS(d^{Gi})$ – его содержание и $V(d^{Gi})$ – объем класса цели. Цель каждого агента в этом случае может быть представлена в виде вектора знаков, характеризующих классы состояний среды функционирования, в которых должны находиться агенты или объекты.

$$DES_i = (d_1^{GJ}; d_2^{GJ}; \dots; d_n^{GJ})$$

Например, вектор цели $DES_j = (d_1^1; d_n^{10})$ для первого агента означает, что его цель изменить состояние среды так, чтобы сам он находился в классе состояний d_1^1 , со значением свойств в области $SS(d^1)$, а агент с номером n находился бы в классе состояний d_n^{10} , со значением свойств в области $SS(d^{10})$. Положения остальных агентов для первого агента не важны.

Действия агентов (*Intention*). В приведенном выше определении среды функционирования считается, что агенты способны изменить ее состояние. Напомним, что изменение состояния среды выражается в изменении значений свойств объектов или собственных свойств агентов. Это является проявлением их реактивности и проактивности. Активность агентов возможна, благодаря наличию у них некоторого ресурса $U_i^R = (u_{i1}^R, \dots, u_{in}^R)$, $u_{in}^R \in Z_i$

В среде функционирования *FS* ресурсы агента, также могут быть представлены в виде множества имен классов состояний среды функционирования:

$$INT_i = \{d_i^{Uj}\},$$

где d_i^{Uj} – имя класса ресурсов, которое однозначно определяет его содержание $SS(d^{Uj})$ и объем класса ресурсов - $V(d^{Uj})$.

Таким образом, все элементы *BDI* архитектуры агентов могут быть выражены в терминах среды функционирования, в виде знаков, определяющих классы возможных ее состояний, целей и действий.

3. Поведение агента в знаковой среде функционирования

Под поведением агента будем понимать его последовательные действия, направленные на достижение цели. Изменение состояния среды, i -м агентом осуществляется с учетом его знаний о среде W_i и может быть выражено системой конечно-разностных уравнений:

$$Z(t+1) = W_i^\circ(Z(t) \cup U_i(t)), \quad (7)$$

где $^\circ$ - правило вывода, $Z(t+1)$, $Z(t) = (z_1, \dots, z_q)$ – состояние ситуации – это векторы значений признаков в моменты времени t , $U_i(t) \in U_i^R$ – вектор управляющих воздействий в моменты времени t .

Отметим, что в уравнении (7) состояния ситуации в последовательные моменты времени $Z(t+1)$, $Z(t)$ записаны в терминах значений свойств агентов и объектов. В случаях если эти значения заданы в виде линейно упорядоченных множеств значений, значительно упрощается процедура вывода, позволяющее получить новое состояние $Z(t+1)$. Процедура вывода в этом случае основана на композиции *max-min* или *max-product*.

Для общего случая уравнение (7) может быть переписано в терминах классов состояний среды функционирования в виде отображения:

$$W_j^{BEL} : SF(t) \rightarrow SF(t+1).$$

Уравнение динамики в этом случае может быть задано множеством правил «Если, То».

Считаем, что действия агента, направленные на достижение цели, возможны в рамках его убеждений - $BEL_i = \{SS_i(d^H), \leq\} \subseteq KK^W$. Это означает, что множество возможных действий агента $INT_i(t)$ должны принадлежать упорядоченному множеству его убеждений (знаний), т.е. $INT_i \subseteq BEL_i$.

Формально поиск действий для достижения заданного вектора цели DES_i можно свести к решению уравнения $G_i = W_i \circ U_i$ относительно управляющих воздействий U_i , где $G_i = (g_{i1}, \dots, g_{im})$, т.е. к решению обратной задачи. Решение обратной задачи запишется в следующем виде:

$$U^* = G_i \circ W_i^{-1}, \quad (8)$$

где U^* - новые значения свойств среды, позволяющие достичь цели G_i , \circ - процедура обратного вывода. Решение обратной задачи - это множество решений $U^* = \{U_i\}$, где U_i - векторы значений свойств, позволяющие достичь цели.

Для уравнения (8), записанного в терминах значений свойств агентов и объектов, процедуры обратного вывода могут быть основаны на композиции, например, *min-max* или *min-division*.

Учитывая соответствие (4) множество решений может быть представлено в терминах состояний среды функционирования, $INT_i(t) = \{d_i^{Uj}\}$, где $d_i^{Uj} = \varphi(x_i(0) + U_j)$. Тогда решение обратной задачи перепишем в виде:

$$INT_i^* = DES_i \circ W_j^{BEL}.$$

3.1. Возможность самостоятельного достижения цели агентом

Цель любого агента DES_i формулируется в виде вектора, характеризующего состояние среды функционирования агентов. Критерием возможности достижения агентом собственной цели является существование у агента ресурсов для ее достижения. Считается, что множество решений обратной задачи INT_i^* , полученные агентом для достижения цели DES_i и множество его возможных действий INT_i должно содержать общие элементы. Формально этот критерий можно записать так: $INT_i^* \cap INT_i \neq \emptyset$.

Утверждение 1. Если пересечение множеств решений обратной задачи и возможных действий агента не пустое множество, $INT_i^* \cap INT_i \neq \emptyset$, то цель достижима агентом самостоятельно.

Следствие 1. Если все элементы цели агента $\forall d_i^H$ принадлежат упорядоченному множеству его убеждений BEL_i , то цель может быть достигнута агентом самостоятельно. Т.е. $\forall d_i^H \in DES_i | d_i^H \in BEL_i$.

Действительно, если у агента есть необходимые ресурсы и знания для достижения поставленной цели, то он может эту цель достигнуть.

3.2. Возможность достижения цели в кооперации с другими агентами

Интерес представляют случаи, когда в векторе цели i -о агента есть элементы, не принадлежащие его системе убеждений BEL_i . В этом случае, цель может быть достигнута при условии кооперации агента с другими агентами среды функционирования. Формально ситуация когда цель некоторого агента не может быть достигнута без кооперации с другими агентами записывается следующим образом: $\forall d_i^H \in DES_i \& \exists d_i^H \notin BEL_i$.

Обозначим элементы вектора цели агента $\forall d_i^H \in DES_i$, не принадлежащие его системе убеждений как множество $CS_i = \{d_i^{Hc}\}$, где $\forall d_i^{Hc} \in DES_i \& \forall d_i^{Hc} \notin BEL_i$. Далее в имена классов состояний, включенных в вектор цели агента, но не включенный в его систему убеждений будем добавлять верхний индекс «c». Т.е. если в вектор цели включено имя класса состояний d_i^H , то это означает, что $d_i^H \in DES_i \& d_i^H \in BEL_i$, если же $d_i^H \in DES_i \& d_i^H \notin BEL_i$, то в векторе цели оно будет обозначаться именем d_i^{Hc} .

Рассмотрим условия кооперации агентов:

Условие 1. Агент j считается привлекательным для агента i для кооперации, если элемент цели $d_i^{Hc} \notin BEL_i$ этого агента существует в системе убеждений (знаний) агента j , т.е. $d_i^{Hc} \in BEL_j$.

Это условие говорит о том, что для кооперации нужно выбирать агента, который может достичь цели агента, нуждающегося в кооперации. При этом, агент j может отказаться от кооперации, если цель агента i ему не интересна.

Условие 2. Агент j считается привлекательным для кооперации для агента i , если элементы системы убеждений агента i , $d_i^H \in BEL_i$ является также элементами системы убеждений агента j , т.е. $d_i^H \in BEL_j$.

Формально это условие означает, что существует элемент d_i^H , содержащийся в пересечении систем убеждений этих агентов. Под пересечением здесь понимается поэлементное сравнение элементов систем убеждений агентов i и j . Результатом такого является вектор элементы, которого равны 0, если $d_i^H \neq d_j^H$, иначе они равны d_i^H , т.е. $\exists d_i^H \in BEL_i \cap BEL_j$.

Условие 3. Агент j и агент i считаются зависимыми по цели, если агент j является привлекательным для кооперации для агента i и агент j является привлекательным для агента i , т.е. если $d_i^{Hc} \in DES_j \& d_j^{Hc} \in DES_i$

Если все три условия выполняются, кооперация агентов становится возможной. Кооперация агентов, рассматриваемая в этой статье,

заключается в изменении агентом j тех свойств агента i , которые он сам, в силу ограниченности его системы убеждений изменить не может.

Допущение 1. Агент j может изменить свойства только тех агентов или объектов, которые включены в объем $V(d_j^H)$ классов состояний его системы убеждений BEL_j . Т.е. $\exists(a_i \in A \vee b_j \in B) \in V(d_j^H) | d_j^H \in BEL_j$.

В связи с этим сформулируем еще одно условие для взаимодействия агентов.

Условие 4. Агент j может изменить свойства агента i , если агент i включен в объем одного из возможных классов состояний агента j , т.е. $a_i \in V(d_j^H) | d_j^H \in BEL_j$.

4. Пример

Рассмотрим пример, поясняющий приведенные выше теоретические построения. Пусть есть два агента: *Агент 1* и *Агент 2*, имеющие по два свойства: $\{X, Y\}$. Для каждого свойства определено множество значений: $X=\{x_i\}$, $x_i \in [0,1]$; $Y=\{y_j\}$, $y_j \in [0,1]$. Тогда каждый агент может быть определен на плоскости $SF=[0,1] \times [0,1]$ точкой, векторами значений его свойств, т.е. «Агент 1» - (x_{1b}, y_{1j}) и «Агент 2» - (x_{2b}, y_{2j}) , где $(x_{1b}, y_{1j}); (x_{2b}, y_{2j}) \in SF$.

На рисунке 1 показана декомпозиция пространства SF на 9 подпространств, которые могут быть представлены как частично упорядоченное множество в виде концептуального каркаса (решетки) KK . Каждому подпространству $SS(d^0), \dots, SS(d^8)$ присвоено имя - d^0, \dots, d^8 , которое обозначает класс состояний среды функционирования. Например, подпространство с именем d^0 определяется как подпространство: $SS(d^0) = [0,33; 0,66] \times [0,33, 0,66]$. Если агент 1 или 2 будут иметь значения свойств $(x_{1b}, y_{1j}) \in SS(d^0)$, то в этом случае будет определен объем - $V(d^0) = \{\langle \text{Агент 1} \rangle\}$. Таким образом в каждой точке среды функционирования определен знак в виде тройки: $d^0, SS(d^0), V(d^0)$ - имя, содержание и объем.

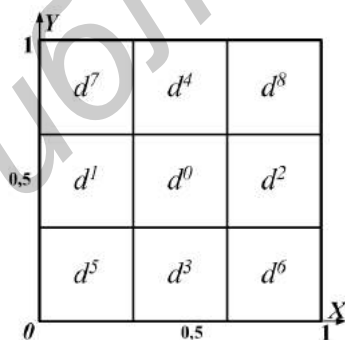


Рисунок 1 – Знаковая среда функционирования

Состояние среды функционирования задано вектором $Z(t) = ((x_{1b}, y_{1j}); (x_{2b}, y_{2j}))$ или в терминах знаковой среды функционирования: $SF(t) = (d_1^I, d_2^J)$, $I, J = 1, \dots, 3^N$, N -число факторов.

Опишем теперь каждого агента в терминах его BDI архитектуры. Считаем, что каждый из агентов имеет неполные знания о среде функционирования. Пусть убеждения (знания) агента 1 - $BEL_1 = \{d^0, d^1, d^2\}$, а агента 2 - $BEL_2 = \{d^1, d^5, d^7\}$. Подпространства среды функционирования, включенные в систему убеждений агентов 1 и 2, показаны на рисунке 2 и затемнены.

Пусть агент 1 находится в подпространстве с именем d^0 , $(x_1(0)=0,5, y_1(0)=0,5)$. Считается, что он может изменить значение только своего свойства X . В этом случае вектор управляющих воздействий равен: $U_1^R = (u_{11}^R, u_{12}^R)$, где $u_{11}^R = -0,5, u_{12}^R = 0,5$. Чтобы перейти агенту 1 в позицию d^1 нужно применить действие u_{11}^R , т.е. $x_1(t+1) = x_1(0) + u_{11}^R = 0$, а в позицию d^2 действие u_{12}^R , т.е. $x_1(t+1) = x_1(0) + u_{12}^R = 1$. Чтобы остаться агенту в позиции d^0 , он не должен предпринимать никаких действий, т.е. $u_{10}^R = 0$. Множество возможных действий агента $U_1(t) = \{-0,5; 0; 0,5\}$ представим в терминах знаков в виде: $INT_1 = \{d^1, d^0, d^2\}$. Уравнение динамики агента 1 запишем в виде: $x_1(t+1) = x_1(0) + (u_{21} \vee u_{22} \vee u_{20})$.

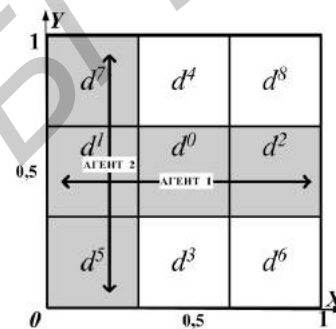


Рисунок 2 – Системы убеждений агентов 1 и 2

Аналогично, для агента 2, находящегося в подпространстве d^1 , $(x_2(0)=0, y_2(0)=0,5)$ запишем множество возможных действий - это $U_2(t) = \{-0,5; 0; 0,5\}$, или в терминах знаков: $INT_2 = \{d^1, d^5, d^7\}$. Уравнение динамики агента 2 запишем в виде: $y_2(t+1) = y_2(0) + (u_{21} \vee u_{22} \vee u_{20})$.

Пусть у агента 1 есть цель $G_1 = (x_1=1,0; y_1=0,5)$. Эта цель в терминах знака будет выглядеть так: $DES_1 = (d^2)$. Поскольку в этом примере значения координаты y_1 для цели и текущего состояния равны, $y_1 = y_1(0) = 0,5$, то наши дальнейшие рассуждения касаются только координаты x_1 . Выбор действия агентом 1 для перехода в целевое состояние определяется решением обратной задачи. В этом случае решение уравнения $u_1^* = G_1 - x_1(0) = 0,5$ дает единственное решение, которое означает, что агенту 1 нужно увеличить свою координату x_1 на 0,5. В этом случае знаковое выражение для этого действия следующее $INT_1^* = (d^2)$.

Согласно утверждению и следствию 1 имеем, что $INT_1^* \cap INT_1 \neq \emptyset = d^2$, что означает, что агент может достичь этой цели самостоятельно. Аналогично можно описать самостоятельные действия агента 2, позволяющие ему перемещаться в рамках тех подпространств, которые включены в его систему убеждений BEL_2 .

Рассмотрим теперь вопросы кооперации агентов, для достижения целей, не принадлежащих системам их убеждений. При рассмотрении вопросов кооперации агентов, будем считать, что одни агенты могут менять значения некоторых свойств других агентов, разумеется, с их согласия.

Пусть цель агента 1 - $DES_1=(d_1^5)$. Для этой цели выполняется условие $d_1^5 \in DES_1$ & $d_1^5 \notin BEL_1$. Это означает, что достичь этой цели агент 1 самостоятельно не может, т.к. он не может самостоятельно изменить свое свойство Y . Далее мы будем опускать подробности коммуникации между агентами в процессах поиска партнера. Считаем, что у агента 1 есть информация об агенте 2 для того чтобы проверить условия возможной кооперации.

Условие 1 для цели агента 1 и системы убеждений агента 2 выполняется, т.е. $d_1^{5c} \in BEL_2$, условие 2 для системы убеждений агента 1 и агента 2 также выполняется, т.е. $d_1^1 \in BEL_1 \cap BEL_2 \neq \emptyset$. Это значит, агент 2 привлекателен для кооперации. Условие 3 – условие взаимной полезности будет выполнено, только в случае если цели у агента 2 равна - $DES_2=(d_2^5)$. И, наконец, кооперация агентов будет возможна, и агент 2 сможет изменить свойство Y агента 1, если выполнится условие 4. Для этого необходимо, чтобы агент 1 переместился в позицию d_1^1 , которая была определена при проверке условия 2. Таким образом, если агент 1 переместится в класс состояний d_1^1 и обратится к агенту 2 с просьбой изменить его свойство Y , то цель агента 1 - $DES_1=(d_1^5)$ будет достигнута совместными усилиями двух агентов.

Заключение

В статье рассмотрена формальная модель агентов BDI архитектурой в терминах знаковой модели среды их функционирования. Сформулированы формальные условия для кооперации агентов.

Библиографический список

- [Цетлин, 1969] Цетлин М.Л. Исследования по теории автоматов и моделированию биологических систем. – М.: Наука, 1969. – 316 с.
- [Стефанюк, 1967] Стефанюк В.Л., Цетлин М.Л. О регулировке мощности в коллективе радиостанций // Проблемы передачи информации. – 1967. – Т. 3, №4. – С. 59–67.
- [Варшавский, 1984] Варшавский В.И., Поспелов Д.А. Оркестр играет без дирижера. – М.: Наука 1984. – 208 с.
- [Карпов, 2014] Карпов В.Э. Процедура голосования в однородных коллективах роботов // XIV национальная конференция по искусственному интеллекту с международным участием КИИ-2014 (24-27 октября 2014 г., Казань, Россия): Труды конференции. Т.2. – Казань: Изд-во РИЦ «Школа», 2014, – 341 с. 159-167.
- [Павловский, 2002] Павловский В.Е., Кирикова Е.П. Моделирование управляемого адаптивного поведения гомогенной группы роботов // Искусственный интеллект. – 2002. – №4. – С. 596–605.
- [Кулинич, 2014] Кулинич А. А. Модель командного поведения агентов (роботов): когнитивный подход / Управление большими системами. Выпуск 51. М.: ИПУ РАН, 2014. С.174-196.

[Cohen, 1991] Cohen P., Levesque H.J. Teamwork. Nous, 25(4), (1991) Special Issue on Cognitive Science and Artificial Intelligence, P. 487-512.

[Grosz, 1996] Grosz B., Kraus S. Collaborative Plans for Complex Group Actions // Artificial Intelligence. – 1996. – №86. – P. 269–358.

[Rao, 1995] Rao A.S., Georgeff M.P. BDI Agents: From Theory to Practice // Proc. First International Conference on Multi-Agent Systems (ed. V.Lesser). – AAAI Press/The MIT Press, 1995. – P. 312–319.

[Городецкий, 2011] Городецкий В.И. Теория, модели, инфраструктуры и языки спецификации командного поведения автономных агентов. Обзор (Часть 1, Часть 2) // Искусственный интеллект и принятие решений. – 2011. – №2, №3 – С. 19–30, С. 34–47.

[Кулинич, 2014] Кулинич А.А. Концептуальные каркасы онтологий слабо структурированных предметных областей // Искусственный интеллект и принятие решений. 2014. - № 4. - С. 31-41.

MODEL OF AGENTS COOPERATION IN THE SEMIOTIC ENVIRONMENT

Kulinich A. A.

*V.A. Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences. Russia, Moscow.*

kulinich@ipu.ru

The model of the agents' cooperation presented in the sign environment is considered. The model of the formal agent with BDI the architecture, described in terms of the sign environment is offered. Conditions of the agents' cooperation, in the conditions of incompleteness of their knowledge about the sign environment are formulated.

Introduction

Theories of command work of agents with BDI architecture are considered. It is the theory of the general intentions and the theory of collaborative plans. To simply the agents communications processes it is offered to investigate the agents command work in the sign environment.

Main Part

The model of the sign environment of agents is offered. In terms of the sign environment the BDI architecture of agents is formulated.

Conditions of independent achievement by agents of the purpose and a condition of achievement of the purpose at cooperation with other agents are considered.

The example of the description of the BDI agents' architecture and an example of teamwork for purpose achievement is resulted.

Conclusion

In article the formal model of agents with BDI architecture in terms of the sign environment model is considered. Formal conditions for cooperation of agents are formulated.



УДК 004.891

МОДИФИЦИРОВАННЫЙ МЕТОД ЛОГИЧЕСКОГО ВЫВОДА ЗНАНИЙ НА ОСНОВЕ НЕЧЕТКОЙ ОНТОЛОГИИ И БАЗЫ ПРЕЦЕДЕНТОВ

Мошкин В.С., Ярушкина Н.Г.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

PostForVadim@yandex.ru

jng@ulstu.ru

В данной работе представлены результаты применения методологии интеграции систем продукционных правил с нечеткой онтологией на примере решения задачи оценки состояния локальной вычислительной сети в процессе искусственного повышения трафика. Помимо этого, рассмотрена возможность модификация данного алгоритма за счет параллельного запуска механизма анализа прецедентов в процессе логического вывода знаний.

Ключевые слова: онтология, прецедент, логический вывод, нечеткие онтологии.

Введение

Для обеспечения высокого качества принимаемых решений в процессе проектирования сложных технических систем необходимы модели, ориентированные на описание объектных структур рассматриваемой предметной области, позволяющие идентифицировать, анализировать (включая наглядное представление) и манипулировать всем многообразием объектов и отношений, имеющихся в предметной области (ПрО).

Однако состояние дел в сфере информационно-аналитического обеспечения процесса проектирования сложных систем недостаточно отвечает потребностям управления в современных производственных организациях, и существует ряд научных проблем, требующих системного решения:

- необходимость разработки семантического базиса анализа хранимых знаний в рамках решения задач автоматизированного проектирования;
- отсутствие интегративных концептуальных моделей, использующих различные подходы хранения знаний о предметной области;
- необходимость универсализации процесса автоматизированной обработки хранимых знаний при решении проектных задач различного рода;
- необходимость одновременного использования разноаспектных описаний особенностей рассматриваемой предметной области;

- необходимость решения проблемы учета нечеткости в человеческих рассуждениях;
- необходимость учета прошлого опыта решения задач, подобных поставленной перед проектировщиком.

Таким образом, на сегодняшний день актуальной является задача интеграция различных форм и алгоритмов представления и логического вывода знаний с целью поддержки принятия решений в процессе проектирования сложных систем.

1. Нечеткость в логическом выводе знаний

1.1. Преимущества использования нечетких алгоритмов в ЭС

При попытке формализовать человеческие знания, необходимые в процессе логического вывода рекомендаций в ЭС, возникает проблема использования традиционного математического аппарата для их описания. Для решения данной проблемы были введены в процесс логического вывода элементы нечеткости как реализация аппарата нечеткой (fuzzy) алгебры, предложенной американским исследователем Лотфи Заде в начале 70-х годов XX века.

Быстрый рост областей применимости алгоритмов нечеткого логического вывода обусловлен главными преимуществами fuzzy-систем:

- возможность оперировать нечеткими входными данными;
- возможность нечеткой формализации критериев оценки и сравнения.;
- возможность проведения качественных оценок как входных данных, так и выходных результатов;
- возможность проведения быстрого моделирования сложных динамических систем и их сравнительный анализ с заданной степенью точности.

В таблице 1 представлен ряд систем, использующих один или несколько алгоритмов нечеткого логического вывода [Trausan-Matu, 2008], [Nunes et al, 2012], [Lei, 2012].

Таблица 1.1 Примеры экспертных систем на основе

Наименование	Применение	Базовый алгоритм
Risk Analysis Expert System(RA_X)	Экспертная система анализа рисков	Мамдани
Adaptive-Network-Based Fuzzy Inference System (ANFIS)	Система прогнозирования состояния морских волн	Суджено
Fuzzy inference system in river flow forecasting	Система прогнозирования течения реки	Мамдани, Такаги–Суджено–Канг
СУ БПЛА	Система управления беспилотными летательными аппаратами	Мамдани
FIS To Fault Diagnosis	Система тестирования подшипников в условиях повышения нагрузки	Суджено

Несмотря на то, что с каждым годом область применения систем, основанных на использовании алгоритмов нечеткого логического вывода, расширяется, каждый из используемых алгоритмов имеет значительные недостатки в случае рассмотрения поставленной задачи через призму сложной предметной области:

- не учитываются особенности взаимосвязей объектов рассматриваемой предметной области;
- все алгоритмы являются однотактными и не предполагают возможности нахождения выхода при изменении входных значений в случае итерационных алгоритмов;
- не учитывается возможная иерархия и синонимия объектов;
- алгоритмы не содержат элементов обучаемости, что значительно снижает потенциал их развития и повышает трудозатраты экспертов на полноценное описание особенностей поведения объектов рассматриваемой предметной области.

В связи с этим возникает задача разработки и реализации гибридных алгоритмов логического вывода с использованием нечеткости,

обеспечивающих исправление перечисленных недостатков. Одним из таких решений является применение класса нечетких онтологий.

1.2. Формальная модель нечеткой онтологии

Переход от обычных онтологий к нечетким выглядит вполне естественным, поскольку понятия и отношения естественного языка, представляющие собой исходный материал для построения онтологии неоднозначны, неточны и не имеют жестких границ. Поэтому адекватным средством формализации онтологий могут служить модели на базе лингвистических переменных, нечеткие множества, нечеткие отношения, нечеткие графы и нечеткие деревья, нечеткие ограничения, нечеткие реляционные и алгебраические системы [НГС, 2007].

В общем виде любая нечеткая FuzzyOWL-онтология может быть представлена следующим образом:

$$I = (I_f, C_f, P_f, D_f, Q_f, L_f, Mod_f). \quad (1)$$

где

- I_f – множество объектов классов онтологии;
- C_f – множество нечетких классов онтологии;

$$C_f = \{C_f^A, C_f^C\}, \quad (2)$$

где C_f^A – множество абстрактных классов, C_f^C – множество конкретных классов онтологии;

- P_f – множество свойств объектов:

$$P_f = \{P_f^A, P_f^C\}, \quad (3)$$

где P_f^A – множество конкретных свойств, т.е. свойств объектов (Object Property), P_f^C – множество абстрактных свойств, т.е. свойств типа данных (Datatype Property);

- D_f – множество аксиом онтологии:

$$D_f = \{A_f^{ABox}, A_f^{TBox}, A_f^{RBox}\}, \quad (4)$$

где A_f^{ABox} – множество утверждений об индивидах, A_f^{TBox} – множество терминологических аксиом, A_f^{RBox} – множество аксиом отношений (иерархия отношений). Часть аксиом может быть подклассом множества нечетких аксиом, которые предполагают истинность утверждения с определенной степенью.

- O_f – множество значений степеней, которые могут быть добавлены к объекту нечеткой аксиомы:

$$O_f = \{LD_f, MD_f, ND_f, Var_f\}, \quad (5)$$

где LD_f – множество лингвистических переменных, MD_f – множество степеней модификации, ND_f – множество численных значений степеней, Var_f – множество переменных.

- L_f – множество операторов нечетких логик соответствующих типов.

$$L_f = \{L_f^{Luk}, L_f^{Zad}, L_f^{Goed}, L_f^{Prod}\}, \quad (6)$$

где L_f^{Luk} – множество операторов логики Лукасевича, L_f^{Zad} – множество операторов логики Заде, L_f^{Goed} – множество операторов логики Геделя, L_f^{Prod} – множество операторов продукционной логики. Определяются с помощью встроенного отношения *hasSemantics*;

- Mod_f – множество «нечетких модификаторов», т.е. функций модификации функций принадлежности, нечетких классов и нечетких отношений. Функции могут быть линейными или треугольными.

1.3. Применение алгоритма логического вывода на основе нечеткой онтологии

Взаимосвязь нечеткой онтологии предметной области и набора продукционных правил в процессе логического вывода осуществляется с помощью формирования запросов к онтологии, генерируемых системой анализа при выполнении набора правил [Ярушкина и др., 2015].

Базовый алгоритм формирования вывода на основе интеграции онтологии и систем продукции представлен на рис.1

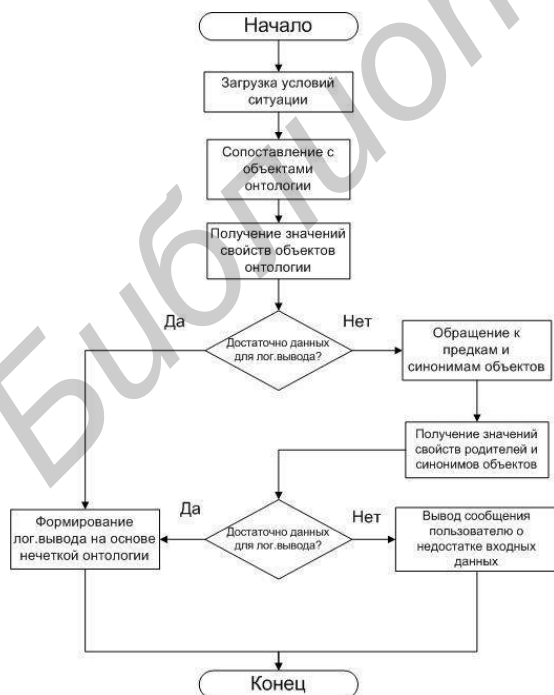


Рисунок 1 – Алгоритм вывода с использованием нечеткой онтологии

С целью сравнения методики взаимодействия онтологического анализа и механизмов логического вывода на основе четкой и нечеткой онтологии был проведен ряд экспериментов, в рамках которых были смоделированы возможные проблемные ситуации, возникающие в процессе работы локальной вычислительной сети при искусственном повышении загрузки каналов связи.

Объектом для проводимых экспериментов стала ЛВС центра разработки электронных мультимедиа технологий (ЦЭМТ) Ульяновского государственного технического университета. Сравнительная характеристика методик логического вывода в процессе моделирования проблемных ситуаций работы ЛВС представлена в таблице 2.

Таблица 2. Сравнение методик логического вывода на основе четких (OWL+SWRL) и нечетких (FuzzyOWL+SWRL) онтологий.

Рекомендации	OWL+SWRL	FuzzyOWL+SWRL
Нет решений	2	1
Несколько решений (в том числе верное или близкое к верному)	4	24
Несколько неверных решений	-	-
Одно верное (близкое к верному) решение	22	4
Одно неверное решение	2	1

Несмотря на то, что включение нечеткости в описание предметной области значительно снижает риски потери возможных результатов работы блока логического вывода и увеличивает гибкость процесса вывода, наличие большого количества вариантов решения задачи не дает пользователю в полной мере положиться на какой-либо из предложенных вариантов [Мошкин и др., 2015].

В этом случае необходима методика, позволяющая пользователю получать конкретное решение, основанное на практическом опыте предыдущих пользователей, решавших в прошлом аналогичную задачу.

2. Использование прецедентов в процессе логического вывода

2.1. Определение понятия рассуждения по прецедентам

При решении сложных задач в области автоматизированного проектирования пользователь должен получать всестороннюю поддержку со стороны системы, имея возможность выбора решения не только посредством использования экспертных знаний, хранящейся в базе системы, но и с учетом опыта решения данной задачи специалистами в прошлом. Такой опыт может быть формализован в виде набора прецедентов, которые должны обрабатываться и учитываться в рамках решения задачи параллельно основному алгоритму.

Согласно [Leake, 1996], **рассуждения по прецедентам** (case-based reasoning, CBR) — это метод формирования умозаключений, опирающийся не на логический вывод от исходных посылок (логические рассуждения), а на поиск и анализ случаев формирования подобных умозаключений в прошлом. Проверка корректности умозаключения может являться частью CBR-процесса.

С точки зрения решения задач, **рассуждения по прецедентам** — это метод получения решения путем поиска подобных проблемных ситуаций в памяти, хранящей прошлый опыт решения задач, и адаптации найденных решений к новым условиям. Применение CBR для решения задач оправдано в случае выполнения следующих условий, касающихся природы прикладной области:

1. Подобные задачи должны иметь подобные решения (*принцип регулярности*). В этом случае накопленный опыт решения задач может служить отправной точкой процесса поиска решения для новых подобных задач.

2. Виды задач, с которыми сталкивается решатель, должны иметь тенденцию к повторению. Это условие гарантирует, что для многих проблем в будущем будет существовать аналог в прошлом опыте [Pal, Shiu, 2004].

Процесс рассуждений по прецедентам может быть схематически представлен в виде цикла. В работе [Aamodt, 1994] приведено описание типового процесса рассуждений по прецедентам, который состоит из следующих этапов: формирование прецедента новой задачи (прецедента запроса); поиск наиболее похожих прецедентов и в БД прецедентов; повторное использование решений найденных прецедентов и их адаптация к новой задаче; проверка предлагаемого решения на адекватность и сохранение нового решения в БД прецедентов.

Существует аналогия между системами, основанными на правилах и прецедентах. И те, и другие необходимо каким-то образом индексировать, чтобы обеспечить эффективное извлечение. И те, и другие выбираются в результате сопоставления, причем выбор и ранжирование производятся на основе знаний, хранящихся в фоновых структурах, наиболее универсальными из которых являются онтологии.

Принципиальные различия между методиками логического вывода на основе систем продукций и прецедентов были в полной мере изложены в [Kolodner, 1993]. Базовые отличия данных методик:

1. Правила представляют собой шаблоны, т.е. содержат переменные и не описывают непосредственно решение, прецеденты же в свою очередь являются константами и оперируют конкретными объектами базы знаний.

2. Применение правил представляет собой итерационный цикл — последовательность шагов, приводящих к решению. Прецедент можно

рассматривать как приближенный вариант полного решения.

3. Правило выбирается на основе точного сопоставления антецедента и входных данных. Прецедент выбирается посредством частичного сопоставления, причем учитываются еще и знания о сущности характеристик, по которым выполняется сопоставление.

2.2. Формальная модель системы вывода, основанной на анализе прецедентов

К основным преимуществам рассуждений по прецедентам можно отнести отсутствие необходимости полного и углубленного рассмотрения знаний о конкретной предметной области; возможность напрямую использовать опыт, накопленный системой, без интенсивного привлечения эксперта в той или иной предметной области; возможность применения эвристик, повышающих эффективность решения задач [Дворянкин и др., 2008].

Формальная модель системы вывода, основанного на прецедентах, может быть представлена в виде упорядоченной тройки:

$$CBR = \{Cases, I, S_{CASE}\}, \quad (7)$$

где *Cases* — база прецедентов;

I — онтология предметной области (нечеткая);

S_{CASE} — алгоритм поиска подходящего прецедента.

Структура прецедента может быть представлена в следующем виде:

$$Case = \{Index_{CASE}, D(Index_{CASE}), Eff(D(Index_{CASE}))\}, \quad (8)$$

где *Index_{CASE}* — индекс прецедента, т.е. описание начальной ситуации;

D(Index_{CASE}) — множество решений поставленной задачи;

Eff(D(Index_{CASE})) — множество оценок эффективности принятого решения задачи.

Так как прецеденты, в отличие от правил, оперируют не переменными, а конкретными объектами классов, а также значениями свойств этих объектов, то, согласно (1), начальная ситуация описывается следующим множеством:

$$Index_{CASE} = \{I_f, P_f\}, \quad (9)$$

где *I_f* — множество объектов классов онтологии;

P_f — множество значений свойств соответствующих объектов.

Аналогична и модель решения задачи: решением является объект вспомогательного класса «Рекомендации» разработанной онтологии, который в качестве вывода в процессе работы алгоритма выдает лингвистическое значение свойства типа данных (DatatypeProperty) «имеетОписание» выбранного объекта класса «Рекомендации».

2.3. Модификация алгоритма логического вывода с использованием прецедентов

Для внесения в базу знаний начального набора прецедентов может использоваться универсальный редактор OWL-онтологии Protégé 4.x. Данная программная система предполагает возможность представления знаний в форме:

- нечетких онтологий (с использованием Fuzzy OWL Plugin);
- SWRL-правил;
- набора прецедентов в OWL-формате.

Для обработки прецедентов в OWL-формате в настоящее время используется java-фреймворк jColibri. Данный набор библиотек является бесплатным и свободно распространяемым. Корректный анализ прецедентов посредством использования фреймворка jColibri предполагает необходимость создания в OWL-онтологии трех базовых классов:

- CBR-CASE - класс, содержащий экземпляры прецедентов;
- CBR-DESCRIPTION - класс, описывающий ограничения, накладываемые на прецеденты;
- CBR-INDEX - хранит структуру прецедента, т.е. классы-потомки и их экземпляры, участвующие в процессе поиска подходящего прецедента.

Перечисленные вспомогательные классы онтологии напрямую согласуются с моделью вывода по прецедентам, представленной в (7) и (8).

Главное особенностью модифицированного алгоритма вывода рекомендаций (рис. 2) является проведение параллельного и независимого логического вывода результатов анализа на основании базы правил и базы прецедентов.

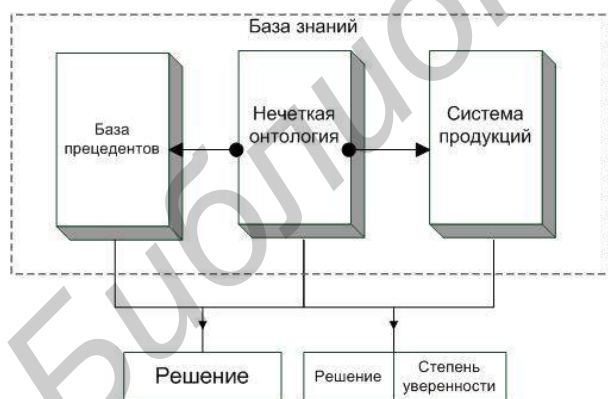


Рисунок 2 – Схема модифицированного алгоритма логического вывода с использованием базы прецедентов

Таким образом, пользователь получает возможность принятия решений как с учетом анализа закономерностей конкретной области, так и на основании опыта пользователей, уже сталкивавшихся с подобными задачами.

Помимо этого, наличие базы прецедентов позволяет внести элемент обучаемости данного алгоритма.



Рисунок 3 – Схема алгоритма формирования базы прецедентов

Главным преимуществом алгоритма формирования базы прецедентов, схема которого представлена на рисунке 3, является возможность пользователем выбора решения, который оказался верным при решении данной задачи.

Решение, полученное в результате логического вывода на основе базы продукционных правил и выбранное пользователем в качестве правильного, будет занесено в базу прецедентов с исходными условиями задачи. В результате данного действия, при решении аналогичной задачи в следующий раз пользователь получит эту рекомендацию в качестве априорной, т.е. полученной на основе анализа опыта решения подобной задачи.

Заключение

Таким образом, предложенный в данной работе модифицированный алгоритм логического вывода с использованием в качестве базы знаний нечеткой FuzzyOWL-онтологии предметной области, набора продукционных правил, а также включающий в себя процесс анализа прецедентов, позволяет:

- снизить риски потери возможных результатов работы блока логического вывода, основанного на анализе предметной онтологии;
- увеличить гибкость процесса вывода рекомендации, предоставляя пользователю более широкий выбор вариантов, упорядоченных по степени их релевантности;
- приблизить формализованное представление знаний о предметной области к естественному для человека варианту их представления и восприятия;
- обеспечить получение пользователем возможности принятия решений как с учетом анализа закономерностей конкретной области, так и на основании опыта пользователей, уже сталкивавшихся с подобными задачами;
- обеспечить возможность обучаемости

системы за счет динамического формирования базы прецедентов.

Библиографический список

[Дворянкин и др., 2008] Дворянкин А. М., Сипливая М. Б., Жукова И. Г. Интеграция рассуждений по прецедентам и онтологии в интеллектуальной системе поддержки инженерного анализа в области контактной механики // Известия Волгоградского государственного технического университета.. Волгоград. – 2008. -№4. - Том 2.

[Мошкин и др., 2015] Мошкин В.С., Ярушкина Н.Г. Логический вывод на основе нечетких онтологий // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VIII-й Международной научно-практической конференции (Коломна, 18-20 мая 2015 г.). В 2-х томах. Т1. – М.: Физматлит, 2015. – С. 259–267.

[НГС, 2007] Нечеткие гибридные системы/ И.З.Батыршин, А.О.Недосекин, А.А.Стецко, В.Б.Тарасов, А.В.Язенин, Н.Г.Ярушкина. – М.: Физматлит, 2007.

[Ярушкина и др., 2015] Ярушкина Н.Г., Мошкин В.С. Применение алгоритма логического вывода на основе FuzzyOWL-онтологии // Радиотехника. – 2015. – № 6. – С. 68–72.

[Aamodt, 1994] Aamodt, A. Case-based reasoning: foundational issues, methodological variations, and system approaches/ A. Aamodt, E. Plaza// Communications. – 1994. – № 7. – С. 39–59.

[Kolodner, 1993] Kolodner J. L. (1993). Case-Based Reasoning. Los Altos, CA: Morgan Kaufmann.

[Leake, 1996] David B. Leake (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park, CA : AAAI Press/MIT Press, 1996, ISBN 0-262-62110-X.

[Pal, Shiu, 2004] Sankar K. Pal, Simon C. K. Shiu. Foundations of Soft Case-Based Reasoning. New Jersey: Wiley, 2004, ISBN: 978-0-471-64466-8.

[Trausan-Matu, 2008] Trausan-Matu, Stefan. A Framework for an Ontology-Based Information System for Competence Management. *Economy Informatics*, 1-4/2008, p.105.

[Nunes et al, 2012] Isabel L. Nunes and M'ario Simões-Marques Applications of Fuzzy Logic in Risk Assessment The RA_X Case, *Fuzzy Inference System Theory and Applications*, Dr. Mohammad Fazle Azeem (Ed.), 2012. – pp. 21-40.

[Lei, 2012] Yaguo Lei. The Hybrid Intelligent Method Based on Fuzzy Inference System and Its Application to Fault Diagnosis, *Fuzzy Inference System Theory and Applications*, Dr. Mohammad Fazle Azeem (Ed.), 2012. – pp. 153-170.

MODIFIED KNOWLEDGE INFERENCE METHODS BASED ON FUZZY ONTOLOGY AND SET OF USE CASES

Moshkin V.S., Yarushkina N.G.

*Ulyanovsk State Technical University, Russian
Federation*

PostForVadim@yandex.ru

jng@ulstu.ru

This paper presents the results of applying the methodology of integration of production rules with fuzzy ontology as an example of solving the problem assessment of the local area network in an artificial rise in traffic. In addition, we consider the possibility of a modification of the algorithm by the parallel launch of the mechanism analysis of precedents in the process of inference of knowledge.

Introduction

To ensure the high quality of decisions need to be assigned a model of object-oriented description of the structures of the subject area, allowing to identify,

analyze (including the visual representation) and manipulate all variety of objects and relations existing in the subject area (ABM).

Therefore, today is a very urgent task of integrating various forms of algorithms and representations and inference of knowledge to support decision-making in the design of complex systems.

Main Part

Fuzzy ontology consisting of five levels reduce the subjective component evaluation data domain concepts of individual experts on the basis of its competence in a given subregion this area.

There are the most versatile approach to the construction of fuzzy ontologies - Methodology of Fuzzy OWL.

The experience can be formalized as a set of precedents that must be evaluated and taken into account within the framework of solving the problem of parallel to the main algorithm.

The arguments by precedents (case-based reasoning, CBR) - a method of forming conclusions, is not based on inference from the assumptions (logical reasoning), and on the search and analysis of cases, the formation of such conclusions in the past. Validation of conclusions can be part of CBR-process.

The presence of the base use case allows you to introduce an element of the learning algorithm.

Conclusion

Thus, we proposed in this paper, a modified algorithm for inference using a knowledge base of fuzzy FuzzyOWL-domain ontology, a set of production rules, and including the process of analysis of precedents, you can:

- reduce the risks of loss of the possible outcomes of the unit of logical inference, based on the analysis of ontology;
- increase the flexibility of the process output recommendations, giving the user a wide range of options, ordered according to their relevance;
- provide users the opportunity to obtain decision-making both in terms of the analysis of patterns of a particular area, and based on the experience of users who are already faced with similar tasks;
- enable the learning system through the formation of dynamic database precedent.



УДК 004.8

АЛГОРИТМ ПРЕДОБРАБОТКИ И ВОССТАНОВЛЕНИЯ АНКЕТНЫХ ДАННЫХ

Сибирев И.В., Афанасьева Т.В.

*Ульяновский государственный технический университет,
г. Ульяновск, Россия*

ivan.sibirev@yandex.ru

tv.afanasjeva@gmail.com

В статье предлагается алгоритм, восстанавливающий недостающие анкетные данные, на основе методов нечёткой кластеризации.

Ключевые слова: предобработка анкетных данных; восстановление данных; нечёткая кластеризация.

Введение

В настоящее время актуальными проблемами являются проблемы обработки данных, в частности, результатов анкетирования, данных, снятых с множества датчиков, результатов целевого запроса и т.д. Данные могут восприниматься как «кадр», характеризующий текущее состояние изучаемой системы, процессы и тенденции изучаемого объекта или явления.

В последние годы возникло множество фирм, берущихся за обработку анкетных данных: составление анкет, проведение анкетирования, перевод анкет в электронный вариант, повторное проведение анкетирования (если это необходимо), обработку анкетных данных вручную или при помощи программных средств. Некоторые фирмы берутся за разработку программного обеспечения для обработки анкетных данных.

Проблемой этапа предобработки анкетных данных является неполнота при заполнении анкет или недоступность отдельных данных.

Существуют два основных подхода при работе с недостающими данными. Первый подход – восстановление данных из сторонних источников информации. Это может быть повторное анкетирование, задание уточняющих вопросов, восстановление информации с использованием внешних баз данных. В IT мире в подавляющем большинстве случаев это реализуется с помощью многообразных систем контроля версий или резервного копирования, как в масштабах отдельного рабочего места, так и в масштабах отдельного сервера или DATA-центра.

Второй подход – редукция (удаление) данных. Если часть анкетных данных отсутствует, при этом невозможно провести повторное анкетирование или задать уточняющие вопросы, тогда в дело вступает редукция. Удаляются те вопросы и те анкеты, в которых не заполнены («повреждены») данные. Часто удаляют наиболее незаполненные вопросы и анкеты. Позиции, оставшиеся пустыми, заполняют, например, самыми типичными ответами по каждому вопросу для данного анкетирования, либо используют статистические модели восстановления данных. Обзор статистических моделей восстановления данных в кластерном анализе приведен в работе [Busse и др, 2005]. Редукция и грубое заполнение пустых мест значительно ухудшают качество анализируемых материалов.

Цель некоторых исследований, обрабатывающих большие объёмы данных, – не в точной конкретизации каждого отдельного параметра, но – в описании общей картины, состояний и тенденций, в том числе, в терминах нечёткой логики.

Нами предлагается алгоритм восстановления данных на стадии предобработки, позволяющий избежать избыточной редукции. Данный алгоритм относится к неточным, но он позволяет заполнить недостающие данные с большей точностью, чем заполнение типичным представителем или средним арифметическим, приводит к меньшим искажениям общей картины при обработке данных методами многомерного анализа (кластеризация, факторный анализ и т.д.).

Данный алгоритм основан на методах кластерного анализа (см.[Дюран и др, 1977], [Жамбю, 1988], [Ким и др., 1989], [Райзин и др, 1980], [Малышев и др, 1991], [Мандель, 1988] и др.). Мы разбиваем анкетные данные на несколько

кластеров. Затем с помощью нечёткой кластеризации (см. [Мальшев и др, 1991]) получаем коэффициенты принадлежности каждой отдельной анкеты к каждому кластеру. В дальнейшем, опираясь на таблицу принадлежности, уточняем незаполненные анкетные данные.

1. Алгоритм восстановления данных

Опишем алгоритм и проиллюстрируем его на примере вычислительного эксперимента.

Шаг 0. Входные данные – числовые или символ N, который обозначает незаполненный ответ на вопрос.

Таблица 1– Исходные данные

Параметры	П1	П2	П3	П4	П5	П6
Анкета 1	1	65	3	N	14	5
Анкета 2	N	23	72	N	7	26
Анкета 3	1	23	43	N	N	52
Анкета 4	124	45	N	N	57	N
Анкета 5	N	23	72	N	23	35
Анкета 6	59	56	43	N	45	12
Анкета 7	N	N	N	N	N	N

Шаг 1. Редукция. Удаляем полностью незаполненные параметры и анкеты.

Таблица 2. Данные после редукции

Параметры	П1	П2	П3	П5	П6
Анкета 1	1	65	3	14	5
Анкета 2	N	23	72	7	26
Анкета 3	1	23	43	N	52
Анкета 4	124	45	N	57	N
Анкета 5	N	23	72	23	35
Анкета 6	59	56	43	45	12

Шаг 2. Временно заполняем все незаполненные ответы средними значениями по параметру.

Таблица 3. Первичное заполнение недостающих данных

Параметры	П1	П2	П3	П5	П6
Анкета 1	1	65	3	14	5
Анкета 2	46,25	23	72	7	26
Анкета 3	1	23	43	29,2	52
Анкета 4	124	45	46,6	57	26
Анкета 5	46,25	23	72	23	35
Анкета 6	59	56	43	45	12

Шаг 3. Кластеризация анкетных данных. Может проводиться одним из методов кластерного анализа. Мы будем использовать центроидный метод [].

Шаг 3.1. Подбор количества кластеров.

Постараемся подобрать количество кластеров так, чтобы не было кластеров, для которых какой-

либо параметр изначально был полностью неизвестным.

Ниже представлены разбиения на 2, 3, 4, 5 кластеров:

x5: C1:A1; C2:A2; C3:A4; C4:A6; C5: A3, A3; - нет;
 x4: C1:A4; C2:A6; C3:A3, A5; C4:A1, A2; - нет;
 x3: C1:A3, A5; C2:A1, A2; C3:A4, A6; - да;
 x2: C1:A4, A6; C2:A1, A2, A3, A5; - да;
 где x2, ..., x5 – разбиения на 2, ..., 5 кластеров, C1, C2, ... – первый, второй и т.д. кластеры, A1, A2, ... – анкета номер 1,2 ... , C2:A1, A2, A3, A5; -означает что анкеты 1,2,3,5 попали в кластер 2.

После каждого разбиения выписано «да» или «нет», что означает отсутствие или присутствие в данном разбиении кластеров с полностью неизвестным параметром, соответственно.

Выберем разбиение, в котором количество кластеров было бы наибольшим при отсутствии кластеров с полностью неизвестным параметром. То есть – выбираем разбиение со словом «да», где наибольшее количество кластеров. Этим условиям в нашем примере соответствует разбиение на три кластера.

Шаг 3.2. Кластеризация центроидным методом на выбранное количество кластеров.

Таблица 4. Кластеризация центроидным методом на 3 кластера

	C1	П1	П2	П3	П5	П6
Анкета 3		1	23	43	29,2	52
Анкета 5		46,25	23	72	23	35
C2						
Анкета 1		1	65	3	14	5
Анкета 2		46,25	23	72	7	26
C3						
Анкета 4		124	45	46,6	57	26
Анкета 6		59	56	43	45	12

C1, C2, C3- названия кластеров. П1, П2, П3, П4, П5 – названия параметров.

Шаг 4. Кластеризуем FCM-методом FCM-метод [Мальшев и др, 1991] – метод нечёткой кластеризации, применяется в паре с другим методом кластеризации, в нашем случае с центроидным методом. Результатом нечёткой кластеризации является таблица коэффициентов принадлежности анкет к кластерам.

Таблица 5. FCM кластеризация

FCM	C1	C2	C3
Анкета 1	0.927	0.024	0.049
Анкета 2	0.047	0.041	0.913
Анкета 3	0.385	0.098	0.517
Анкета 4	0.018	0.95	0.032
Анкета 5	0.018	0.019	0.963
Анкета 6	0.249	0.383	0.372

Шаг 5. Уточнение временно заполненных анкетных данных.

Шаг 5.1. Для каждого кластера (из пункта 3.2) подсчитываем среднее значение каждого параметра по этому кластеру.

Таблица 6. Среднее значение каждого параметра по каждому кластеру

С1	П1	П2	П3	П5	П6
Анкета 3	1	23	43	29,2	52
Анкета 5	46,25	23	72	23	35
Среднее значение	23,63	23	57,5	26,1	43,5
С2					
Анкета 1	1	65	3	14	5
Анкета 2	46,25	23	72	7	26
Среднее значение	23,625	44	37,5	10,5	15,5
С3					
Анкета 4	124	45	46,6	57	26
Анкета 6	59	56	43	45	12
Среднее значение	91,5	50,5	44,8	51	19

Новое значение временно заполненных параметров анкеты вычислим как сумму произведений средних значений этого параметра для каждого кластера (из таблицы 6) на коэффициенты принадлежности данной анкеты кластеру (из таблицы 5). Результаты вычислений приведены в таблице 7.

Таблица 7. Результаты уточнения неизвестных параметров на первой итерации

Параметры	П1	П2	П3	П5	П6
Анкета 1	1	65	3	14	5
Анкета 2	85,62	23	72	7	26
Анкета 3	1	23	43	37,44	52
Анкета 4	124	45	38,09	57	16,12
Анкета 5	88,99	23	72	23	35
Анкета 6	59	56	43	45	12

Первая итерация закончена, переходим к шагу 3.

В нашем примере, спустя несколько итераций, были получены следующие анкетные данные.

Таблица 8. Результаты уточнения неизвестных параметров на четвёртой итерации.

Параметры	П1	П2	П3	П5	П6
Анкета 1	1	65	3	14	5
Анкета 2	52.93	23	72	7	26
Анкета 3	1	23	43	41.13	52
Анкета 4	124	45	47.96	57	23.72
Анкета 5	53.87	23	72	23	35
Анкета 6	59	56	43	45	12

Построим таблицу значений восстанавливаемых параметров по окончании каждой итерации.

Таблица 9. Значения неизвестных параметров после каждой итерации.

Итерация	A2П1	A3П5	A4П3	A4П6	A5П1
0	46.25	29.2	46.6	26	46.25
1	85.62	37.44	38.09	16.12	88.99
2	58.81	41.18	47.75	22.77	58.69
3	44.93	40.81	47.95	23.65	43.69
4	52.93	41.13	47.96	23.72	53.87

Здесь A4П6 – шестой параметр четвёртой анкеты.

Рисунок 1 иллюстрирует сходимость значений восстанавливаемых параметров при возрастании номера итерации. На горизонтальной оси – число итераций, на вертикальной оси – значения параметров. Каждая ломаная на рисунке соответствует значениям одного параметра на разных итерациях.

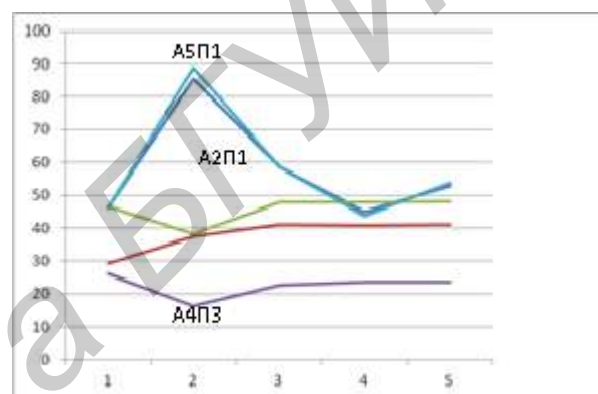


Рисунок 1. Генерируемые данные на первых нескольких итерациях

Наблюдается стремление значений параметров к некоторым предельным значениям при увеличении числа итераций.

2. Вычислительный эксперимент

Для изучения свойств алгоритма требуется апробация на реальных данных. Коллективом ученых Ульяновского государственного технического университета (Н.Г. Ярушкина, Т.В. Афанасьева, О.В. Шиняева, К.В. Святос, Л.М. Валкин, Д.А. Ефремов, К.Г. Калимуллин и др.) в рамках проекта «Исследование ИТ-кластера Ульяновской области», проведено анкетирование руководителей ИТ-предприятий г. Ульяновска. Собраны данные по 87 предприятиям по 39 вопросам анкеты [Ярушкина и др., 2013]. Нами проводились обработка и кластерный анализ этих данных [Афанасьева и др., 2014].

В результате редукции, с заданным порогом заполнения, количество обрабатываемых анкет сократилось с 87 до 49, количество обрабатываемых параметров сократилось с 39 до 33, то есть лишь 48% из потенциально заполняемых позиций подверглось дальнейшей обработке и анализу. Остальные 52% потенциально заполняемых позиций «оказались за бортом», причём из них были

заполнены 5% от общего числа позиций. Иными словами, в редуцированных данных было 10% заполненных позиций от числа удалённых.

С использованием алгоритма, описанного в этой статье, можно было не редуцировать данные вообще, т. к. не нашлось полностью незаполненных параметров или анкет. При этом классификационное значение мало заполненных параметров при кластеризации ничтожно.

Заключение

Данный алгоритм предназначен для восстановления данных на стадии предобработки, в частности, анкетных данных. В настоящий момент область применимости – вещественные данные, но в дальнейшем она может быть расширена за счёт некоторых модификаций.

Алгоритм не генерирует новую информацию, а распространяет известную информацию на незаполненные места анкет. Чем больше известной информации, тем информативнее будет результат.

Алгоритм не способен восстановить аномальные значения, но способен восстановить данные с учётом средних тенденций в этом и других кластерах. Данный алгоритм – более тонкий инструмент, чем «заливание» всех пустующих полей анкет средним арифметическим.

В текущей версии каждая итерация данного алгоритма может потребовать значительных вычислительных затрат. Спасает ситуацию быстрая сходимость алгоритма, что делает общее время вычислений приемлемым для практического использования. У этого алгоритма широкие возможности оптимизации быстродействия.

При использовании данного алгоритма нам удастся:

- сэкономить время и средства на повторном анкетировании;
- «спасти» значительную часть заполненных данных, которые были бы удалены при обычной редукции;
- восстановить незаполненные данные с точностью, позволяющей дальнейшую обработку методами многомерного анализа.

Данный алгоритм может быть использован на этапе предобработки данных как для исследования с использованием пакетного режима обработки данных, так и в исследованиях с использованием KDD систем.

Библиографический список

- [Афанасьева и др., 2014] Программа «Сегментация и кластеризация рынка IT» / Т.В. Афанасьева, И.В. Сибирев // Инновации в науке. Сб.ст. по материалам XXIX междунар. науч.–практ. конф. №1. – Новосибирск : СибАК, 2014. – С. 46-53
- [Дюран и др., 1977] Кластерный анализ / Б. Дюран, П. Одедл. – М.: Статистика, – 1977. – 128 с.
- [Жамбю, 1988] Жамбю, М. Иерархический кластер-анализ и соответствия. Пер. с фр. / М. Жамбю– М.: Финансы и статистика, 1988. – 342 с.

[Ким и др., 1989] Факторный, дискриминантный и кластерный анализ: Пер с англ./ Дж. О. Ким. [и др.]; – М.: Финансы и статистика, 1989. – 215с.

[Райзин и др., 1980] Классификация и кластер. /Под ред. Дж. Вэн. Райзина. – М. : Мир, 1980, – 390 с.

[Мальшев и др., 1991] Нечеткие модели для экспертных систем в САПР / Н.Г. Мальшев и др. – М.: Энергоиздат, 1991. – 136 с.

[Мандель, 1988] Мандель, И. Д. Кластерный анализ / И. Д. Мандель– М.: Финансы и статистика. 1988. – 176с.

[Ярушкина и др., 2013] Исследование модели для экспертных систем Ульяновской области / Н. Г. Ярушкина [и др.]; – Ульяновск : УлГТУ, 2013. – 137 с.

[Busse и др., 2005] Cluster Analysis of Heterogeneous Rank Data. / L.M. Busse, P. Orbanz, J.M. – Zurich: Buhmann Institute of Computational Science, ETH Zurich, 8092

[Давыдов, 2015] Knowledge Discovery and Data Mining в системной социологии. [Режим доступа 2015] http://www.isras.ru/Davydov_Knowledge.html

ALGORITHM FOR PREPROCESSING AND RECOVERY OF QUESTIONNAIRES DATA

Sibirev I.V., Afanasyeva T.V.

Ulyanovsk State Technical University, Ulyanovsk, Russia

ivan.sibirev@yandex.ru;

tv.afanasjeva@gmail.com

The article proposes an algorithm for recovery missing questionnaires data. This algorithm bases on fuzzy clustering methods.

Introduction

The actual problem is preprocessing of data and recovery of missing questionnaires for allowing work by methods of multidimensional analysis based on fuzzy logic.

Main Part

We present algorithm for recovery missing questionnaires data in the preprocessing stage, avoiding excessive reduction.

This algorithm is based on the methods of cluster analysis. We divide data into several clusters. Then using fuzzy clustering we obtain the coefficients of membership of each questionnaire to each cluster. We based on the table of coefficients membership for upgrade questionnaires data.

The algorithm is illustrated by the example of the computational experiment.

We consider example where the results of the algorithm and the classical reduction are compared. The data are compared on the basis of a real experiment for clustering the IT-companies of the Ulyanovsk region.

Conclusion

This algorithm allows: to save time and resources on re-survey; to «save» main data part that would be reduced; to restore missing questionnaires data with precision, allowing work by methods of multidimensional analysis.



УДК 007:519.816

О ФОРМАЛИЗАЦИИ ТЕМПОРАЛЬНЫХ РАССУЖДЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Вагин В.Н. *, Еремеев А.П. *, Гулякина Н.А. **

*Национальный исследовательский университет «МЭИ», г. Москва, Россия

vagin@apmat.ru, eremeev@apmat.ru

**Белорусский государственный университет информатики и радиоэлектроники,

г. Минск, Республика Беларусь

guliakina@bsuir.by

Рассматриваются возможности формализации рассуждений (вывода) с учетом фактора времени в плане их использования в современных и перспективных интеллектуальных системах различного назначения, включая системы реального времени. Работа выполнена при финансовой поддержке РФФИ и БРФФИ.

Ключевые слова: интеллектуальная система, темпоральный вывод, прецедент, сеть Петри, дерево решений, моделирование, технология OSTIS.

Введение

Современные интеллектуальные системы, (ИС) особенно системы реального времени, типичными представителями которых являются ИС поддержки принятия решений реального времени (ИСППР РВ), относятся к классу сложных динамических систем, реализуемых структурно и функционально на основе парадигмы семиотической системы [Вагин и др., 2001; 2015].

Формально ИС (ИС РВ) *семиотического типа* может быть задана набором [Вагин и др., 2015]:

$$SS = \langle M, R(M), F(M), F(SS) \rangle,$$

где $M = \{M_1, \dots, M_n\}$ – множество формальных или логико-лингвистических моделей, реализующих определенные как чисто вычислительные (например, решение задачи математической оптимизации), так и интеллектуальные функции (например, моделирование правдоподобных рассуждений на основе экспертных знаний);

$R(M)$ – правила выбора необходимой модели или совокупности моделей в текущей ситуации, т.е. правила, реализующие отображение

$$R(M): S \rightarrow M,$$

где S – множество возможных ситуаций (состояний), которое может быть и открытым, или

$S' \rightarrow M$, где S' – некоторое множество обобщенных ситуаций (состояний), например, нормальных (штатных), аномальных или аварийных, при попадании в которые происходит смена модели;

$F(M) = \{F(M_1), \dots, F(M_n)\}$ – множество правил модификации моделей M_i , $i = 1, \dots, n$. Каждое правило $F(M_i)$ реализует отображение

$$F(M_i): S'' \times M_i \rightarrow M'_i,$$

где $S'' \subseteq S$, M'_i – некоторая модификация модели M_i ;

$F(SS)$ – правило модификации собственно системы SS , ее базовых конструкций M , $R(M)$, $F(M)$ и при необходимости самого правила $F(SS)$, т.е. $F(SS)$ реализует целый ряд отображений (или комплексное отображение)

$$F(SS): S''' \times M \rightarrow M', S''' \times R(M) \rightarrow R'(M),$$

$$S''' \times F(M) \rightarrow F'(M),$$

$$S''' \times F(SS) \rightarrow F'(SS),$$

где $S''' \subseteq S$, $S''' \cap S' = \emptyset$, $S''' \cap S'' = \emptyset$, т.е. правила модификации данного типа применяются в ситуациях, когда имеющихся множеств моделей, правил выбора и правил модификации недостаточно для вывода (поиска решения). Причем для модификации $F(SS)$ могут быть использованы как внутренние средства порождения моделей и правил (гипотез), так и внешние.

Известно, что в ИС РВ, как в сложной динамической системе, необходимо наличие развитых средств представления фактора времени и моделирования временных (темпоральных) зависимостей как при представлении данных и знаний, так и при моделировании рассуждений, содержащих фактор времени (темпоральных рассуждений) [Еремеев и др., 2003; 2010; 2015а; 2015б]. В данной работе рассматривается ряд

методов и моделей в контексте формализации темпоральных зависимостей и темпорального вывода в ИС РВ.

1. Формализация рассуждений на основе темпоральных прецедентов

При поиске решения плохо формализованных задач, специфичных для ИС РВ типа ИСППР РВ, возникает необходимость применения методов правдоподобных рассуждений (вывода), позволяющих найти приемлемое решение в заданных временных ограничениях. Один из таких методов базируется на том факте, что человеку свойственно при решении новой, неизвестной задачи попытаться в первую очередь использовать решения, которые принимались ранее в подобных (аналогичных) случаях и при необходимости адаптировать их к рассматриваемому. Данный подход лег в основу методов рассуждений на основе аналогий и прецедентов (Case-Based Reasoning, CBR) [Еремеев и др, 2010; Montani at all, 2013].

Данные методы достаточно хорошо проработаны и используются на практике в ИС различного назначения, однако во многих случаях при выводе используются «мгновенные» снимки ситуаций (параметров), а история их изменения не учитывается. Но природа физических процессов такова (что специфично для динамических систем типа ИС РВ), что к одному и тому же значению ключевых параметров контролируемый процесс может перейти разными путями, от которых зависит дальнейшая динамика их изменения. Поэтому остро встает вопрос о разработке новых методов и средств моделирования рассуждений с учетом фактора времени, т.е. темпоральных рассуждений.

Рассмотрим темпоральное расширение метода рассуждений на основе прецедентов (на примере метода ближайшего соседа) возможностью учета поведения контролируемого объекта во времени [Куриленко и др., 2009]. При учете фактора времени появляется возможность проанализировать проблемную ситуацию более глубоко в динамике, а также отследить процесс изменения значений ключевых параметров.

Наиболее простым способом адаптации существующих алгоритмов вывода на основе прецедентов к учету фактора времени является подмена «мгновенного» снимка ключевых параметров «историей» их изменения. Эта история может быть представлена в виде матрицы значений, составленной на определенный период наблюдения с определенным интервалом (тактом). Далее для реализации вывода матрица преобразуется в вектор путем последовательной записи в него элементов ее столбцов. Таким образом, в прецедент включаются не только значения параметров в текущий момент времени, но и их значения за некоторый период времени до этого. При этом эксперту (лицу, принимающему решения, ЛПР) может представляться более удобная для интерпретации и заполнения матричная форма, а ее преобразование в

массив для применения того или иного CBR-метода ИС РВ может выполняться автоматически.

Другой способ введения фактора времени в процесс рассуждений основывается на учете решений, полученных на прошлых этапах при выборе решения на текущем этапе. Проиллюстрируем его на примере расширения метода ближайшего соседа. В первую очередь следует определить отрезок времени для анализа проблемной ситуации (т.е. глубину анализа). Далее рассматриваемый промежуток времени разбивается на N равных отрезков с некоторым шагом (тактом).

В сформированной базе прецедентов, где каждый прецедент представляется в параметризованном виде [Варшавский и др, 2009]

$$\text{CASE} = (x_1, x_2, \dots, x_n, R),$$

где x_1, x_2, \dots, x_n – параметры ситуации, описывающей данный прецедент; $x_1 \in X_1, x_2 \in X_2, \dots, x_n \in X_n$, где n – количество параметров для описания прецедента, а X_1, \dots, X_n – области допустимых значений соответствующих параметров, R – решение (диагноз, рекомендации ЛПР), происходит сравнение значений параметров ситуации в каждый момент времени i , где $i = 0, 1, \dots, N$.

Значения параметров прецедентов сравниваются между собой по методу ближайшего соседа с выбранной метрикой (например, Евклидовой). Для всех моментов i получаются варианты прецедентов с определенными интегральными оценками, которые могут быть выбраны как соответствующие ситуации в этой точке для текущего значения порога подобия. На следующем этапе по интегральным оценкам в моментах i выбирается наиболее вероятный прецедент или группа прецедентов, удовлетворяющих поисковому условию. Каждому результирующему прецеденту в соответствие ставится выбранная мера сходства, которая может, в частности, задаваться правилом: «При совпадении всех параметров в описании прецедента и текущей ситуации степень сходства равна 1, а каждый совпавший параметр дает вклад равный $1/n$, где n – число параметров в описании прецедента и текущей ситуации». Отметим, что, используя значения параметров в точках i , можно с помощью интерполяции построить прогноз развития проблемной ситуации.

На результат поиска прецедентов, удовлетворяющих проблемной ситуации, оказывает влияние несколько факторов. Во-первых, это выбор метрики. В каждом конкретном случае выбор производится по-разному, в зависимости от целей ЛПР, физической и статистической природы используемой при управлении сложным объектом информации, а также других ограничений и факторов, влияющих на процесс поиска решения. Во-вторых, влияние оказывает выбор шага, с которым разбивается на промежутки рассматриваемый отрезок времени. Вариация размеров шага дает различное количество дискретных точек, по которым происходит

сравнение значений параметров прецедентов. Это оказывает влияние не только на точность выбора прецедента, но и на скорость выполнения поискового алгоритма, а также на величину погрешности, с которой может быть построен прогноз. И, в-третьих, важным фактором оказывается выбор порогового значения, определяющего степень сходства.

Предложенные подходы могут быть использованы как по отдельности, так и в комбинации друг с другом. В практических приложениях важно правильно (адекватно) определить величину N и размер такта.

Изложенный метод на основе темпоральных прецедентов можно развить и улучшить в плане применения в ИС РВ. Очевидное улучшение – переход к моделям, позволяющим хранить меньший объем информации и допускающих рассуждения с использованием как метрических, так и качественных темпоральных зависимостей на основе темпоральной версии задачи согласования ограничений [Еремеев и др, 2010].

Определим метрическую точечную задачу согласования темпоральных ограничений (МЗСТО) как $Z=(V,D,C)$, где $V = \{V_1, V_2, \dots, V_m\}$ – конечное множество темпоральных переменных, соответствующих моментам времени; D – область значений темпоральных переменных (является множеством вещественных или целых чисел); C – конечное число бинарных темпоральных ограничений вида $C_{ij} = \{[a_1, b_1], \dots, [a_k, b_k]\}$, где интервалы попарно не пересекаются. Бинарные ограничения позволяют определить разрешенную дистанцию между моментами времени и интерпретируются как $(a_1 \leq V_j - V_i \leq b_1) \cup \dots \cup (a_k \leq V_j - V_i \leq b_k)$. Для решения МЗСТО требуется скорректировать все ограничения так, чтобы они не противоречили друг другу. Если подобная корректировка возможна, то МЗСВО считается согласованной (иначе – несогласованной).

Используем формализм МЗСВО для описания наблюдаемых ситуаций и прецедентов. При этом фиксируется как сам факт наступления тех или иных событий, их порядок, так и время их появления (метрика).

Далее будем предполагать, что МЗСТО согласованы и преобразованы в минимальный вид. Для вычисления соответствия наблюдаемой ситуации и прецедента могут быть применены как метод с жесткими, так и метод с мягкими темпоральными ограничениями.

Метод с жесткими ограничениями предполагает точное соответствие наблюдаемых событий и событий, имеющих в прецеденте, а также для каждого метрического ограничения C_{ij} в прецеденте и ограничения C^*_{ij} в наблюдаемой ситуации должны выполняться условия $C_{ij} \cap C^*_{ij} = C_{ij}$. При этом для того, чтобы события в прецеденте и анализируемой ситуации соответствовали друг другу, предлагается использовать для их нумерации числа, получаемые

в результате сортировки по имени параметра и времени.

Среди достоинств данного метода – высокая скорость вывода и высокая точность результата. Метод полезен для систем, в которых временные характеристики протекающих процессов (время наступления событий и их порядок) достаточно стабильны. Однако для рассматриваемой МЗСТО в целом он подходит лишь частично в виду достаточно сильной жесткости условий.

Так называемый «мягкий» метод может рассматриваться в трех вариантах: 1) условие $C_{ij} \cap C^*_{ij} = C_{ij}$ заменяется на пару $C_{ij} \cap C^*_{ij} \neq \emptyset$ и $C^*_{ij} \subseteq C_{ij}$; 2) условие $C_{ij} \cap C^*_{ij} = C_{ij}$ заменяется на $C_{ij} \cap C^*_{ij} \neq \emptyset$; 3) условие $C_{ij} \cap C^*_{ij} = C_{ij}$ заменяется на условие близости границ в ограничениях: $|\text{lo}(C_{ij}) - \text{lo}(C^*_{ij})| < \varepsilon \wedge |\text{hi}(C_{ij}) - \text{hi}(C^*_{ij})| < \varepsilon$.

Дальнейшее смягчение условий возможно при анализе степени сходства с учетом изменения истории изменений каждого параметра по отдельности. Тут возможны разные стратегии (например, учитывающие ограничения между событиями изменения значений разных параметров с большим весом, чем ограничения, наложенные на события полученные из-за изменения одного параметра и т.п.).

Вычисление прецедента на основе выборки похожих ситуаций (обучение) в данном случае может быть выполнено на основе смягчения ограничений C_{ij} в одном из вариантов так, чтобы выполнялись условия его подобия другим ситуациям.

Отметим, что оба метода могут быть адаптированы для работы с неточной информацией, а именно: в случае отсутствия значений какого-либо из параметров соответствующие события и ограничения, в которых они присутствуют, могут быть удалены из прецедента, после чего обычным методом может быть оценена степень подобия.

Таким образом, организация прецедентов на основе МЗСТО позволяет учитывать, как последовательности событий, так и их длительности, т.е. как качественную, так и количественную информацию. В случае, если анализ длительностей не принципиален, а важен порядок событий, то можно воспользоваться качественной темпоральной логикой – например, точечной логикой, для которой в работах [Куриленко 2009; Еремеев и др, 2010, Куриленко и др., 2014] предлагаются достаточно быстродействующие алгоритмы вывода. В этом случае для представления прецедента может использоваться точечная ЗСТО (ТЗСТО), а определение степени сходства наблюдаемой ситуации и прецедента может основываться на решении задачи логической эквивалентности соответствующих минимальных ЗСТО.

2. Временные зависимости и их модели на основе темпоральных сетей Петри

Объекты, для мониторинга и управления, которыми используются ИС РВ (ИСППР РВ), в общем случае являются «неалгоритмическими» параллельными системами с недетерминированным поведением, в которых отдельные компоненты функционируют независимо, взаимодействуя друг с другом в некоторые моменты времени. Среди многих существующих методов описания и анализа подобных параллельных систем существует подход, который основан на использовании сетевых моделей, восходящих к сетям специального вида, предложенным К. Петри для моделирования асинхронных информационных потоков в системах преобразования данных [Котов, 1984]. *Сети Петри* (СП) зарекомендовали себя как удобный, наглядный, и в то же время математически строгий формализм для моделирования и анализа. СП позволяют естественно описывать синхронизацию, параллелизм, конфликт и причинную зависимость, а также наглядно представлять структуру и функционирование сложных систем.

СП изначально предназначались для моделирования изучения в динамике поведения системы параллельных процессов. Однако позже область использования этого аппарата существенно расширилась и сегодня формализм СП применяется для решения достаточно сложных теоретических задач, в том числе, из области искусственного интеллекта и ИС. Одной из работ, посвященных этому вопросу, является статья [Вагин и др., 1987], в которой обосновывается возможность использования СП для параллельного дедуктивного вывода. Следует, однако, отметить, что решение подобных «интеллектуальных» задач с помощью классического формализма СП зачастую требует построения крайне громоздких моделей. Поэтому активно развивается направление, связанное с разработкой различных модификаций СП, базирующихся на классических СП и обладающих дополнительными свойствами, необходимыми для эффективного решения определенных задач.

Отсутствие возможности учета фактора времени в классических СП не позволяет эффективно использовать их в контексте формализации темпоральных зависимостей и темпорального вывода в ИС РВ. Значительная часть исследований в области развития и модификации аппарата СП посвящена разработке различных темпоральных расширений. На кафедре Прикладной математики НИУ «МЭИ» в качестве базового формализма для использования в ИС РВ предложен аппарат *раскрашенных сетей Петри реального времени с поддержкой темпоральной логики Аллена* (РСП РВ ТЛА) [Еремеев и др., 2013].

Пусть L – язык типизированных выражений, которые построены из переменных и констант с использованием только операции сложения комплектов, U – конечная модель этого языка, элементами которой являются различимые фишки.

Обозначим как $U^{[1]} \cong \bigcup_{n \in \mathbb{N}_0} U^{[n]}$ множество всех конечных упорядоченных подмножеств вида $U^{[n]} \cong \{[U[1], U[2], \dots, U[n]] \mid (\forall i \in 1..n) U[i] \in U\}$, где $(\forall i', i'' \in 1..n) (i' \neq i'' \Rightarrow U[i'] \neq U[i''])$. Множество всех возможных комплектов (мультимножеств) из элементов множества U обозначим как $U^{\cup} \cong \{(u_1, u_2, \dots, u_n) \mid n \in \mathbb{N}_0, (\forall i \in 1..n) u_i \in U\}$.

Тип элемента $x \in U$ обозначается как $\xi(x)$, тип выражения $\theta \in L$ как $\xi(\theta)$. Множество переменных, входящих в выражение θ , обозначается через $\nu(\theta)$. Определим РСП РВ ТЛА как кортеж

$$RTCPN \equiv \langle \Sigma, P, T, \dot{F}, \dot{\xi}, \dot{\gamma}, \pi, \varepsilon_{\Sigma}, \varepsilon_T, m_0 \rangle,$$

где: Σ – конечное множество типов (цветов), $|\Sigma| \in \mathbb{N}$, $\Sigma \subseteq U^{[1]}$; $P \equiv [p_1, p_2, \dots, p_{|P|}]$ – конечное упорядоченное множество мест, $|P| \in \mathbb{N}$;

T – конечное множество переходов, $|T| \in \mathbb{N}$, $P \cap T = \emptyset$;

$\dot{F} \subseteq (P \times T) \cup (T \times P)$ – непустое множество дуг, $|\dot{F}| \in \mathbb{N}$;

$\dot{\xi} : P \rightarrow \Sigma$ – функция, ставящая в соответствие каждому месту $p \in P$ тип $\xi(p) \in \Sigma$: каждая фишка $u \in U$ в p должна быть этого типа, $u \in \dot{\xi}(p)$;

$$\dot{\gamma} : T \rightarrow Bool,$$

где $Bool = \{true, false\}$ – защитная (охранная) функция, ставящая в соответствие переходу $t \in T$ некоторое логическое выражение;

$\pi : T \rightarrow \mathfrak{R}_0$ – функция приоритетов переходов;

$\varepsilon_{\Sigma} : \dot{F} \rightarrow \dot{L}$ – функция весовых выражений дуг, ставящая в соответствие каждой дуге выражение языка \dot{L} , такое, что

$$(\forall p \in P)(\forall f \in \dot{F})$$

$$((f \Leftarrow t, p \succ \vee f \Leftarrow p, t) \supset \supset \xi(\varepsilon_{\Sigma}(f)) \subseteq \dot{\xi}(p));$$

где $\varepsilon_T : \dot{F} \rightarrow \mathfrak{R}_0$ – функция темпоральных выражений дуг;

$m_0 \in \dot{M}$ – начальная маркировка (состояние) сети, где $\dot{M} \equiv \{m \mid m : P \rightarrow (U^{\cup} \cup \{\emptyset\}) \times \mathfrak{R}\}$ – множество всех возможных маркировок.

Полагаем, что для всех $m \in \dot{M}$ и $p \in P$ $m(p) \equiv \langle \mu(p), \dot{t}(p) \rangle$, $\mu(p) \in (U^{\cup} \cup \{\emptyset\})$, $\dot{t}(p) \in \mathfrak{R}$. Для любой фишки $u \in \mu(p)$ выполняется $u \in \dot{\xi}(p)$. Второй компонент $\dot{t}(p)$ определяет временную метку места. Для перехода $t \in T$ обозначим V_t множество переменных языка \dot{L} , которые

встречаются в выражениях функций ε_Σ и ε_T входных и выходных дуг перехода и в выражении защитной функции $\dot{\gamma}(t)$. Подстановка перехода $t \in T$ – функция $\beta: V_i \rightarrow U^()$, такая, что $(\forall v \in V_i) \beta(v) \in \xi(v)$. Запись $\dot{\gamma}(t)_\beta$ обозначает вычисление защитной функции перехода $t \in T$ в подстановке β , $\varepsilon_\Sigma(f)_\beta \in U^()$ и $\varepsilon_T(f)_\beta$ обозначают вычисления функций соответственно весовых и темпоральных выражений дуги $f \in \dot{F}$ в подстановке β .

Ключевым свойством РСП РВ ТЛА является поддержка работы с качественными темпоральными зависимостями: другие темпоральные модификации СП позволяют учитывать только количественные. О важности учета временных зависимостей обоих типов при работе с ИС РВ говорится, например, в [Еремеев и др., 2003].

В РСП РВ ТЛА для оперирования качественной временной информацией используются *интервалы*. Пусть $T_n \cong [\tau_0, \tau_1, \dots, \tau_n]$ – некоторое конечное упорядоченное подмножество моментов $\tau_i \in T_D$, $i \in 0..n$, $T_D \cong \{\tau \mid \tau = k * \tau', k \in \mathbb{N}_0, \tau' \in \mathcal{R}_+\}$, где $(\forall i, j \in 0..n)(i > j \supset \tau_i > \tau_j)$. Назовем интервалом конечное упорядоченное подмножество $Int \cong [\tau_0, \tau_1, \dots, \tau_n]$, $\tau_i \in T_D$, $i \in 0..n$, для которого $(\forall i \in 1..n)(\exists k \in \mathbb{N})(\tau_i = k * \tau' \supset \tau_{i-1} = (k-1) * \tau')$. Пусть $\rho: P \times T_D \rightarrow U^() \times \mathcal{R}$ – функция, определяющая маркировку сети в момент времени $\tau \in T_D$, а $U_p \cong \{\dot{u} \mid \dot{u} \in U^(), (\forall u \in \dot{u}) u \in \xi(p)\}$ – множество комплектов фишек. Определим функцию $\zeta: P \times U_p \rightarrow T_D$:

$$\begin{aligned} \zeta(p, \dot{u}) &\cong [\tau_0, \tau_1, \dots, \tau_n], (\forall i \in 0..n) \tau_i \in T_D, \\ (\forall i, j \in 0..n)(i > j \supset \tau_i > \tau_j), \\ (\forall i \in 0..n)(\rho(p, \tau_i) = m(p) \Leftarrow \mu(p), \dot{\tau}(p) \succ \dot{u} \subseteq \mu(p) \wedge \dot{\tau}(p) \leq 0). \end{aligned}$$

Подмножество $\zeta(p, \dot{u})$ можно представить как кортеж интервалов: $\zeta(p, \dot{u}) \Leftarrow Int_0, Int_1, \dots, Int_n >$, $(\forall i \in 0..n) Int_i = [\tau_{i_0}, \tau_{i_1}, \dots, \tau_{i_n}]$. Определим функцию $\chi: \zeta \times T_D \rightarrow Int_D \cup \{\emptyset\}$, ставящую в соответствие моменту времени $\tau \in T_D$ интервал $Int_i = [\tau_{i_0}, \tau_{i_1}, \dots, \tau_{i_n}]$ из кортежа $\zeta(p, \dot{u}) \Leftarrow Int_0, Int_1, \dots, Int_m >$:

$$\begin{aligned} \chi(\zeta(p, \dot{u}), \tau) &\cong [(\exists Int_i = [\tau_{i_0}, \tau_{i_1}, \dots, \tau_{i_n}] \in \zeta(p, \dot{u})) \\ (\tau_{i_0} \leq \tau \wedge (\forall Int_j = [\tau_{j_0}, \tau_{j_1}, \dots, \tau_{j_n}] \in \zeta(p, \dot{u})) \\ (j > i \supset \tau_{j_0} > \tau)) \rightarrow Int_i : \emptyset]. \end{aligned}$$

В каждый момент работы модели на основе предлагаемого аппарата рассматриваются текущие или последние завершившиеся интервалы. В

качестве защитных функций переходов $\dot{\gamma}: T \rightarrow Bool$ используются формулы темпоральной логики Аллена (ТЛА) [Allen, 1983] $\varphi: (Int_D \cup \{\emptyset\}) \times (Int_D \cup \{\emptyset\} \times T_D) \rightarrow Bool$:

$$\varphi = \chi(\zeta(p_1, \dot{u}_1), \tau) \{r_1, r_2, \dots, r_n\} \chi(\zeta(p_2, \dot{u}_2), \tau),$$

где $r_i \in B$, $i \in 1..|B|$, $p_1, p_2 \in P$, $\dot{u}_1 \in U_{p_1}$, $\dot{u}_2 \in U_{p_2}$, B – множество базисных отношений ТЛА.

Подобная модификация аппарата СП позволяет работать с темпоральными зависимостями между объектами сложных систем, как количественными, так и качественными. В настоящее время формализм РСП РВ ТЛА применяется на кафедре прикладной математики НИУ «МЭИ» в плане исследований и разработки методов и инструментальных средств конструирования ИСПР РВ семиотического типа на основе нетрадиционных логик. Рассматриваются вопросы анализа и верификации подобных сетей [Еремеев и др., 2015; Королев, 2015]. Планируется дальнейшее расширение области использования данного аппарата – в том числе, и в задаче формализации темпорального вывода.

3. Формирование обобщенных понятий с использованием темпоральных деревьев решений

Во многих случаях для описания поведения сложных систем приходится использовать сотни независимых атрибутов, которые необходимо анализировать, чтобы наиболее точно смоделировать поведение системы. В такой ситуации крайне важно решать задачу обобщения для получения компактных описаний классов ситуаций на объекте управления. Однако следует учесть, что важным параметром, отражающим динамику поведения сложной системы, является время. Обобщение информации, отражающей изменение ситуаций во времени, требует использования специальных методов, таких как, например, темпоральные деревья решений.

В рассматриваемой задаче информация о функционировании объекта управления представлена множеством результатов измерений, поступающих с датчиков; таким образом, состояние сложного объекта и его компонентов описывается набором признаков (количественных и качественных), при этом одним из признаков является время. Целью работы является получение обобщенных описаний классов ситуаций, возникающих на объекте, причем с каждым классом ситуаций связываются определенные действия по управлению объектом, называемые далее восстановительными действиями.

Рассмотрим возможные ситуации на объекте. Введем понятия: C_n – множество ситуаций, возникающих на сложном техническом объекте, которые диагностируются как нормальные; C_f – множество ситуаций, в которых наблюдаются

неисправности на объекте. Предлагается сформировать описание понятий C_n и C_f в рамках введенной модели. На основе полученных обобщенных описаний классов C_n и C_f необходимо выработать рекомендации по выбору восстанавливающего действия на сложном техническом объекте; такое действие должно переводить систему из состояния «неисправность» в состояние «норма».

Постановка задачи обобщения приведена в [Вагин и др., 2008]. Дано O множество объектов, представленных в некоторой интеллектуальной системе. Пусть $V \subset O$ множество объектов системы, относящихся к определенному классу (множество положительных объектов).

$O = V \cup W$; $V \cap W = \emptyset$, W образует множество отрицательных объектов относительно данного класса.

Пусть имеется обучающая выборка $K = K^+ \cup K^-$, такая, что $K^+ \subset V$, $K^- \subset W$, $K^+ \cap K^- = \emptyset$.

Необходимо на основе анализа обучающей выборки построить понятие, разделяющее положительные и отрицательные примеры. Понятие Q считается сформированным, если удалось построить решающее правило, которое для любого примера из обучающей выборки указывает, принадлежит ли этот пример понятию, или нет. Сформированное понятие может быть представлено в виде логической функции, дерева решений, набора продукционных правил вида «ЕСЛИ условие ТО искомое понятие».

Далее предлагается использовать метод деревьев решений, который широко используется в задачах классификации объектов, представимых признаковыми описаниями. Таким образом, формирование обобщенного понятия Q предлагается выполнить путем построения дерева решений. Однако деревья решений в их классическом виде имеют некоторые ограничения, в частности, невозможность работы с поведением объекта или системы во времени, невозможность принятия решений с течением времени. Без учета фактора времени не удастся проследить динамику изменения состояния системы. Предлагается расширить признаковое описание объектов: введем понятие «время» как один из атрибутов, используемых при построении дерева решений. Будем далее использовать дискретное время: $t = 0, 1, 2, \dots$.

Неформально *темпоральное дерево решений* (ТДР) – это дерево, в котором вершины-листья помечены видом неисправности и предлагаемым в данной ситуации восстановительным действием, а промежуточные вершины, называемые далее внутренними вершинами, помечены именами атрибутов, в совокупности с видом проверки и временной меткой. Дуги ТДР определяют переходы по результатам проверок значений атрибутов во внутренних вершинах.

Дадим формальное определение. ТДР — это взвешенный ациклический ориентированный граф $Ttemp = (V_{temp}, E_{temp})$. Во множестве вершин V_{temp} выделим вершину $v_0 \in V_{temp}$ – корень дерева. Все вершины разделим на два класса: $V_i \subseteq V_{temp}$ – множество внутренних вершин (узлов) дерева; V_l включает в себя такие вершины, из которых выходят дуги; $V_e \subseteq V_{temp}$ – множество внешних, конечных, вершин дерева (листьев); V_e включает в себя такие вершины, из которых дуги не выходят. V_i и V_e образуют разбиение множества вершин V ТДР:

$$V_i \cap V_e = \emptyset,$$

$$V_i \cup V_e = V_{temp}.$$

Внутренние вершины V_i дерева взвешены (помечены) парой

$$\langle a, tc \rangle,$$

где a – имя атрибута; tc – временная метка.

Вершины-листья V_l взвешены (помечены) названием или номером ситуации из $C_n \cup C_f$. Если ситуация отнесена к классу C_n вершина – лист помечается названием или номером предлагаемого восстановительного действия.

Каждая дуга e ТДР взвешена условием «атрибут[tc]=значение_атрибута» (для качественных значений) либо «атрибут[tc] σ значение_атрибута» (для количественных значений атрибутов $\sigma = \{ \geq, >, =, \leq, < \}$). Здесь «атрибут» – имя атрибута в вершине, из которой исходит дуга e , «значение атрибута» – одно из возможных значений (количественное или качественное) признака «атрибут»; tc – момент времени, в который необходимо проводить эту проверку, $0 \leq tc < t^*$.

Таким образом, основным отличием ТДР от обычных деревьев решений является наличие метки времени в каждом внутреннем узле дерева. Проверка значения атрибута во внутреннем узле дерева производится только в том случае, если момент времени, которым помечен набор значений датчиков, совпадает с временной меткой в этом узле.

ТДР предлагается использовать для решения задач диагностики сложного технического объекта; примером такой задачи может служить задача бортовой диагностики автомобиля. Данные, поступающие с различных датчиков, описывают состояние всей системы в конкретный момент времени, а использование ТДР позволяет проследить изменение состояния системы за некоторый интервал времени и выявить неисправности или неблагоприятные тенденции, которые могут по истечении некоторого времени повлечь неисправности.

На основании анализа известного алгоритма для построения ТДР, изложенный в [Console et al., 2003] (назовем его CPD как сокращение от фамилий авторов – Console-Picardi-Dupre) был разработан оригинальный алгоритм Temporal ID3, который

является расширением алгоритма ID3 [Quinlan, 1986], учитывающим фактор времени. По сравнению с алгоритмом CPD, на ТДР не накладывается никаких ограничений по временным меткам в узлах. Однако не учитывается стоимость восстановительных действий при выборе наблюдения на каждом шаге. Кроме того, снимается ограничение на уменьшение временных меток при движении от корня дерева к листьям, что приводит к необходимости сохранять некоторые значения датчиков для дальнейшего использования при проведении диагностики.

Исходными данными как для первоначального обобщения, так и для диагностики, является таблица наблюдений. Также может использоваться некоторая модель для восстановительных действий: например, может быть введен частичный порядок над действиями — тогда при наличии неразличимых ситуаций и сравнимых действий достаточно будет выполнить наиболее сильное из действий. Общая схема алгоритма построения ТДР Temporal_ID3 приведена ниже. На вход алгоритма подаются:

- 1) таблица с ситуациями;
- 2) наблюдения в виде множества пар <датчик, временная метка>;
- 3) модель восстановительных действий.

Алгоритм Temporal_ID3 (S: Таблица с ситуациями, O: Наблюдения, M: Модель восстановительных действий)

Результат: ТДР \tilde{T}

Начало

Если для всех ситуаций из S восстановительные действия совпадают,

то вернуть Лист(S, M)

Пусть D — минимальный крайний срок для ситуаций из S.

Если ситуации из S неразличимы на основе показаний датчиков с меткой времени $t \leq D$, *то вернуть* Лист(S, M).

Выбрать наблюдение $\langle s^*, t' \rangle$, которое будет проверяться в данном узле дерева.

Пусть $s^*_1, s^*_2, \dots, s^*_n$ — различающиеся показания датчика s^* в момент времени t' , а $S^*_j, j=1, 2, \dots, n$ — подмножества ситуаций из S, состоящие из ситуаций с показанием s^*_j датчика s^* в момент времени t' .

Вернуть ТДР с корнем, помеченным выбранным наблюдением $\langle s^*, t' \rangle$, и дугами, помеченными $s^*_1, s^*_2, \dots, s^*_n$, соединяющими

корень соответственно с ТДР

Temporal_ID3($S^*_1, O \setminus \{ \langle s^*, t' \rangle \}$, M)

Temporal_ID3($S^*_2, O \setminus \{ \langle s^*, t' \rangle \}$, M)

...

Temporal_ID3($S^*_n, O \setminus \{ \langle s^*, t' \rangle \}$, M)

конец

В результате выполнения такой рекурсивной процедуры на выходе получаем ТДР. Функция Лист(S, M) на основе ситуаций и модели

восстановительных действий строит лист дерева, помечая его множеством восстановительных действий, полученным согласно модели.

При выборе наблюдения для разбиения используется критерий "прирост информативности" Куинлана [Quinlan, 1986]. Так как ограничение на уменьшение временных меток при движении от корня дерева к листьям снято, при построении ТДР показания датчиков можно рассматривать как обычные атрибуты [Антипов и др., 2010]. Алгоритм строит такое ТДР, в котором с каждым узлом ассоциирован атрибут, являющийся наиболее информативным среди всех атрибутов, еще не рассмотренных на пути от корня дерева к листьям.

Было проведено моделирование процесса диагностики на основе использования ТДР. В случае обнаружения неисправности либо отказа, с помощью ТДР предлагается выбрать возможное восстановительное действие, способное предотвратить аварию. Как было показано выше, действия (восстановительные действия) характеризуются некоторой стоимостью, которая отражает уменьшение функциональности системы. Таким образом, основная цель процедуры диагностики заключается в выборе оптимального действия при динамическом изменении параметров.

При проведении эксперимента было выявлено, что ТДР, построенные с использованием алгоритмов CPD и Temporal ID3, правильно определяют значительное количество некорректных ситуаций и выбирают нужные восстановительные действия. При этом следует отметить, что одним из недостатков алгоритма CPD является большое число ложных срабатываний — так называемых ошибок первого рода. Поэтому минимизация размера ТДР не должна быть единственным критерием при его построении.

4. Реализация формальных моделей темпоральных рассуждений средствами технологии OSTIS

Технология OSTIS (Open Semantic Technology for Intelligent Systems) представляет собой технологию компонентного (модульного) и платформенно независимого проектирования совместимых интеллектуальных систем различного назначения [Голенков и др., 2015].

Технология OSTIS ориентирована на проектирование интеллектуальных систем, имеющих сложноструктурированные базы знаний и реализующие параллельные модели обработки знаний. Это делает технологию OSTIS достаточно перспективной для реализации моделей темпоральных рассуждений. Особую важность здесь имеет то, что различные модели темпоральных рассуждений могут быть оформлены как компоненты интеллектуальных систем, многократно используемые и легко интегрируемые в различные интеллектуальные системы.

В интеллектуальных системах, построенных по технологии OSTIS (такие системы будем называть ostis-системами), база знаний представляет собой сложноструктурированную семантическую сеть, в которой имеются элементы, не только обозначающие внешние сущности и связи между ними, но и обозначающие различные классы элементов семантической сети, различные фрагменты этой сети, различные связи между указанными классами и фрагментами.

Основой формального представления моделей темпоральных рассуждений средствами технологии OSTIS является специально разработанная для этого иерархическая система предметных областей и соответствующих им формальных онтологий. В число указанных предметных областей входят:

- предметная область временных сущностей;
- предметная область ситуаций и событий;
- предметная область ситуаций и событий в семантической памяти (в памяти, где хранится база знаний ostis-системы);
- предметная область действий и задач;
- предметная область действий формальных агентов, работающих над семантической памятью.

Заключение

В докладе представлены некоторые основные результаты, полученные в научных группах НИУ «МЭИ» (кафедра прикладной математики) и БГУИР (кафедра интеллектуальных информационных технологий) по проблематике формализации темпоральных рассуждений. Совместные исследования проводятся в плане разработки эффективных методов и моделей для использования их в перспективных интеллектуальных системах различного назначения, в частности, ИС РВ и ИСППР РВ семиотического типа, предназначенных для помощи ЛПП в проблемных ситуациях и в условиях достаточно жестких временных ограничений.

Библиографический список

- [Вагин и др., 2001] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // Изв. РАН. Теория и системы управления. – 2001. – № 6. – С. 114-123.
- [Вагин и др., 2015] Вагин В.Н., Еремеев А.П. Научная школа искусственного интеллекта в Московском энергетическом институте на базе кафедры прикладной математики: становление и развитие // Вестник МЭИ. – 2015. – № 2. – С. 29-37.
- [Еремеев и др., 2003] Еремеев А.П., Троицкий В.В. Модели представления временных зависимостей в интеллектуальных системах поддержки принятия решений // Известия АН. Теория и системы управления. – 2003. – № 5. – С. 75-88.
- [Еремеев и др., 2010] Еремеев А.П., Куриленко И.Е. Средства темпорального вывода для интеллектуальных систем реального времени // В кн.: Интеллектуальные системы. Коллективная монография. Выпуск 4 / Под. ред. В.М. Курейчика. – М.: Физматлит, 2010. – С. 222-252.
- [Еремеев и др., 2015 а] Еремеев А.П., Королев Ю.И. Анализ и верификация моделей процессов в сложных динамических системах // Российская академия наук. Искусственный интеллект и принятие решений. – 2015. – № 1. – С. 45-56.
- [Еремеев и др., 2015 б] Еремеев А.П., Королев Ю.И. Анализ и верификация раскрашенных сетей Петри реального времени с поддержкой логики Аллена // Открытые семантические технологии проектирования интеллектуальных систем: материалы IV Междунар.

научно-техн. конф. – Минск: БГУИР, 2014. – С. 461-464.

[Montani et al., 2013] Stefania Montani, Lakhmi C. Jain. Successful Case-based Reasoning Applications-2. – Springer, 2013.

[Куриленко, 2009] Куриленко И.Е. Применение временной логики при построении интеллектуальной системы управления крупными парковочными комплексами // Сб. док. научно-практ. конф. ИММВИ 2009. Т.2 – М.: ФизМатЛит, 2009. – С. 171-180.

[Варшавский и др., 2009] Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений. // Российская академия наук. Искусственный интеллект и принятие решений. – 2009. – №2. – С. 45-57.

[Куриленко и др., 2014] Куриленко И.Е., Шорникова Д.А. Реализация подсистемы временного вывода на базе точно-интервальной временной логики для систем поддержки принятия решений реального времени // Четырнадцатая нац. конф. по искусственному интеллекту с междунар. участием КИИ-2014: Тр. конф. Т.1. – Казань: Изд-во РИЦ «Школа», 2014. – С. 50-58.

[Котов, 1984] Котов В.Е. Сети Петри. – М.: Наука, 1984.

[Вагин и др., 1987] Вагин В.Н., Захаров В.Н., Розенблом Л.Я. Логический вывод на интерпретированных сетях Петри // Техническая кибернетика. – 1987. – № 5. – С. 187-195.

[Еремеев и др., 2013] Еремеев А.П., Королев Ю.И. Реализация интеллектуальных систем реального времени на основе сетей Петри с поддержкой темпоральных зависимостей // Программные продукты и системы. – 2013. – № 3 (103). – С. 88-94.

[Королев, 2015] Королев Ю.И. Верификация моделей процессов в динамических системах по методу Model Checking // Открытые семантические технологии проектирования интеллектуальных систем. (OSTIS-2015): материалы V Междунар. науч.-техн. конф. – Минск: БГУИР, 2015. – С. 545-548.

[Вагин и др., 2008] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В.Н. Вагина, Д.А. Поспелова. 2-е издание дополненное и исправленное. – М.: Физматлит, 2008. – 712 с.

[Console et al., 2003] Console L., Picardi C., Duprè D. Temporal decision trees: model-based diagnosis of dynamic systems on-board // Journal of artificial intelligence research. 2003. №19.

[Quinlan, 1986] Quinlan J.R. Induction of decision trees // Machine learning, 1986. №1

[Антипов и др., 2010] Антипов С.Г., Фомина М.В. Метод формирования обобщенных понятий с использованием темпоральных деревьев решений // Искусственный интеллект и принятие решений. № 2, 2010. – С.64-76

[Голенков и др., 2015] Голенков В.В., Гулякина Н.А. Семантическая технология компонентного проектирования систем, управляемых знаниями // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар.науч.-техн.конф.- Мн.: БГУИР, 2015. – с. 57-78.

ABOUT THE TEMPORAL REASONING FORMALIZATION IN THE INTELLIGENT SYSTEMS

Vagin V.N.*, Eremeev A.P.*, Guliakina N.A.**,
*National Research University «MPEI»,
Moscow, Russia*

vagin@appmat.ru; eremeev@appmat.ru

*Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus
guliakina@bsuir.by*

The formalizing of reasoning (inference), taking into account the time factor, and its usage in modern and perspective intelligent systems of a different kind including real-time systems are considered in this paper. The work is supported by RFBR and BRFB.

Keywords: intelligent system, temporal reasoning, precedent, Petri-net, temporal decision tree, modeling, technology OSTIS.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 007:519.816

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ СИСТЕМ НА ОСНОВЕ ТЕМПОРАЛЬНЫХ СЕТЕЙ ПЕТРИ

Еремеев А.П., Королев Ю.И.

*Национальный исследовательский университет «МЭИ»,
г. Москва, Россия*

Eremeev@appmat.ru

KorolevYu@gmail.com

В работе рассматривается вопрос разработки программного инструментария, предназначенного для моделирования сложных динамических систем и позволяющего оперировать временными (темпоральными) зависимостями. Моделирование осуществляется на основе формализма раскрашенных сетей Петри реального времени с поддержкой логики Аллена. Описывается прототип инструментария в среде *G2*, разработка программных средств на языках высокого уровня. Приведен пример использования созданного программного обеспечения в задаче разработки моделей объектов электроэнергетической системы.

Ключевые слова: интеллектуальная система поддержки принятия решений, темпоральные сети Петри, моделирование систем, разработка инструментария.

Введение

В настоящее время активно разрабатываются методы и программные средства проектирования математического и программного обеспечения перспективных интеллектуальных систем поддержки принятия решений (ИСППР), включая ИСППР реального времени (ИСППР РВ) семиотического типа, способных к развитию и адаптации к управляемому объекту и к изменениям внешней среды [Вагин и др., 2001]. Такие системы предназначены для помощи лицам, принимающим решения (ЛПР), при управлении и мониторинге сложных динамических систем (ДС) в условиях жестких временных ограничений и при наличии неопределенности в поступающей информации. ИСППР РВ относятся к классу динамических интеллектуальных систем (ДИС) [Осипов, 2008], основанных на использовании методов искусственного интеллекта и поиска решений на основе знаний. Одним из основных блоков ДИС является подсистема моделирования [Вагин и др., 2008], а потому разработка программных средств моделирования сложных ДС является важной задачей, возникающей при создании ИСППР РВ.

Учет времени необходим при решении задач диагностики и мониторинга сложного объекта, планирования действий, прогнозирования последствий решений, управления в реальном времени, обучения ЛПР. Поэтому при разработке

ИСППР РВ особое внимание должно уделяться средствам оперирования темпоральными зависимостями, как количественными, так и качественными. В ряде современных коммерческих инструментальных комплексов для построения сложных ДС (*G2*, *RTworks* и др.) имеются средства отображения времени и темпоральных зависимостей, однако они весьма примитивны и не соответствуют сегодняшним требованиям.

В основу системы моделирования современных ИСППР РВ необходимо заложить достаточно мощный формальный аппарат. Разработка абсолютно нового формализма сопряжена со значительными трудностями, поэтому эффективным путем является выбор существующего формализма и его модификация с целью разработки математического и программного обеспечения для моделирования и анализа сложных ДС, а также включения этих средств в состав ИСППР РВ. В качестве базового формализма для создания моделей, ориентированных на использование в составе ИСППР РВ семиотического типа, предлагается аппарат на основе раскрашенных сетей Петри (РСП), позволяющий оперировать темпоральной информацией [Еремеев и др., 2013].

В данной статье рассматриваются вопросы разработки и использования программного инструментария для моделирования сложных ДС (включая ДИС), в основе которого лежит формализм модифицированных РСП.

1. Базовый математический формализм

РСП [Jensen, 1992] – графоориентированный язык для проектирования, описания, имитации и контроля распределенных и параллельных систем. Графическими примитивами отображается течение процесса, а конструкциями специального языка имитируется необходимая обработка данных. Главным отличием РСП от классических сетей Петри (СП) является типизация данных, основанная на понятии множества цветов, которое аналогично типу в декларативных языках программирования. Класс РСП совпадает по выразительной мощности с классом СП, однако построенная на их базе модель получается, как правило, более компактной и структурированной.

Для практического использования РСП разработана система *CPN Tools*, которая поддерживает простой подкласс РСП, допускающий использование временных задержек на фишках. Подобный подход эффективен, если при моделировании предполагается соответствие каждой фишки некоторому ресурсу с определенными темпоральными ограничениями. Однако при создании моделей сложных ДС рационально использовать фишки как индикаторы состояния определенного объекта системы. Кроме того, простая поддержка временных меток в РСП делает крайне сложной задачу учета темпоральных зависимостей между объектами модели. Поэтому подход к обработке темпоральной информации, используемый в РСП, неэффективен при решении задачи моделирования сложных ДС.

В качестве эффективного средства моделирования и анализа процессов в сложных ДС (в том числе и в собственно ДИС) предлагается использовать аппарат раскрашенных сетей Петри реального времени с поддержкой темпоральной логики Аллена (РСП РВ ТЛА) [Еремеев и др., 2013]. Формализм РСП РВ ТЛА является расширением подкласса РСП РВ [Szpyrka, 2008] и обладает следующими ключевыми особенностями:

- каждому месту сети p ставится в соответствие временная метка $\tau(p)$. Фишки-токены места, временная метка которого положительна, недоступны для переходов сети;
- каждая дуга f имеет два параметра: весовое и временное выражения, причем результатом вычисления весового выражения $\varepsilon_2(f)$ является комплект фишек, результатом вычисления временного выражения $\varepsilon_1(f)$ является неотрицательное вещественное число;
- каждому переходу t ставится в соответствие некоторое логическое выражение типа $\{true, false\}$ – защитная функция $\gamma(t)$. Переход t может сработать только в том случае, если в результате вычисления выражения $\gamma(t)$ получается значение $true$.
- каждому переходу t ставится в соответствие приоритет $\pi(t)$, что позволяет напрямую моделировать детерминированный выбор.

При вычислении значений весовых и временных выражений дуг сети и защитных функций переходов используется функция подстановки β , которая замещает каждую переменную в выражениях определенным значением соответствующего типа. Функционирование модели на основе РСП РВ ТЛА осуществляется путем перехода сети из одного состояния в другое. Такой переход происходит вследствие одного из следующих событий:

- срабатывание перехода t в некоторой подстановке β ;
- уменьшение значений временных меток всех мест на фиксированную величину τ (временной шаг работы модели).

При этом безусловный приоритет имеет срабатывание перехода. Уменьшение значений временных меток позволяет только дожидаться момента, когда может сработать очередной переход.

Правила оперирования темпоральной информацией в РСП РВ ТЛА позволяют моделировать процессы функционирования ДС и оперировать как количественными, так и качественными темпоральными зависимостями. Для отображения количественных зависимостей используются конструкции временных меток мест и временных выражений дуг. Для работы с качественными зависимостями используются защитные функции переходов $\gamma(t)$, которые могут задаваться формулами темпоральной логики Аллена (ТЛА) [Allen, 1983] вида:

$$\varphi ::= \langle p_1 \rangle \langle \langle u_1 \rangle \{ r_1, r_2, \dots, r_n \} \langle p_2 \rangle \langle \langle u_2 \rangle \rangle. \quad (1)$$

Обозначение $\langle p \rangle \langle \langle u \rangle \rangle$ определяет временной интервал, на котором в месте сети p находится комплект фишек u . В каждый момент работы модели рассматриваются текущие, или последние завершившиеся интервалы. Формула ТЛА истинна ($true$), если между двумя интервалами соблюдаются отношения r_1, r_2, \dots, r_n , и ложна ($false$) в противном случае. Базисные отношения ТЛА приведены в [Allen, 1983].

Более подробные формальные определения рассматриваемого базового формализма РСП РВ ТЛА и описание принципов функционирования моделей на его основе приведены, в частности, в [Еремеев и др., 2015a]. Вопросы анализа и верификации таких моделей рассмотрены в работе [Еремеев и др., 2015b].

Преимуществом использования средств ТЛА является возможность использовать не только количественные временные меры, но и качественные отношения между интервалами, определяющими состояния отдельных объектов системы. Это позволяет, в частности, разбивать сети сложных моделей на более простые несвязные подсети и адекватно отражать неопределенность, присущую сложным ДС, что является одним из основополагающих принципов при проектировании перспективных ИСППР РВ и ДИС в целом.

2. Разработка инструментария

Компьютерное моделирование сложных ДС актуально как для теоретических исследований, так и для практического применения. Современные инструментальные средства конструирования ДИС, в частности, мощный инструментальный комплекс конструирования систем реального времени *G2* (*Gensym Corp.*, США) позволяют реализовывать подобные проекты.

2.1. Реализация прототипа в среде G2

G2 – это объектно-ориентированная интегрированная среда для разработки и сопровождения сложных приложений – интеллектуальных систем реального времени, использующих базы знаний. В отличие от систем, ориентированных на какую-то одну методологию или конкретную предметную область, *G2* интегрирует в себе множество взаимодополняющих методов искусственного интеллекта, что упрощает и ускоряет процесс разработки приложений и позволяет делать их достаточно универсальными. Программные продукты, разработанные с помощью *G2*, являются независимыми от аппаратного обеспечения, на котором они работают (т.е. являются переносимыми). Возможность простого манипулирования графическим представлением объектов в *G2* и составления схем, являющихся отображением технологических цепочек или алгоритмов обработки данных, обеспечивает базовые средства для построения проблемно-ориентированных языков визуального программирования. Реализованы версии *G2* для всех типов рабочих станций и персональных ЭВМ, а также обеспечивается возможность интеграции *G2* с большим числом программных и аппаратных средств самых разных фирм.

В системе *G2* был разработан прототип инструментария для моделирования процессов в сложных ДС на основе РСП РВ ТЛА. При этом был введен ряд допущений:

- в модели отсутствуют переменные, это приводит к тому, что любая подстановка перехода β является по сути тривиальной: $\beta = ()$;
- создание только безопасных сетей (не более одной фишки в каждом месте), весовыми выражениями дуг $\varepsilon_{\Sigma}(f)$ являются только единичные фишки;
- упрощенный алгоритм разрешения конфликта переходов;
- защитные функции переходов $\gamma(t)$ на основе ТЛА необходимо реализовывать в отдельных процедурах;
- временной шаг модели τ зафиксирован и равен 1 секунде.

В качестве примера рассмотрим модель системы экстренного торможения поезда, описанную в [Еремеев и др., 2013] (рисунок 1). В кабине машиниста каждые 60 секунд загорается световой сигнал, чтобы проверить, контролирует ли он

идущий поезд. Если машинист проигнорирует световой сигнал, то через 6 секунд включается звуковой сигнал. Затем, если машинист не деактивирует его в течение 3 секунд, срабатывает механизм аварийного торможения.

Формулы ТЛА применены в данном случае как защитные функции переходов *DisactLS* и *DisactSS*, обозначающих своевременную реакцию машиниста (ЛПП) на световой и звуковой сигнал соответственно:

- *Driver(active) {d} LightSig(on)* – переход *DisactLS* срабатывает, когда фишка *active* в месте *Driver* появится в тот момент, когда в месте *LightSig* будет находиться фишка *on*;
- *Driver(active) {d} SoundSig(on)* – переход *DisactSS* срабатывает, когда фишка *active* в месте *Driver* появится в тот момент, когда в месте *SoundSig* будет находиться фишка *on*.

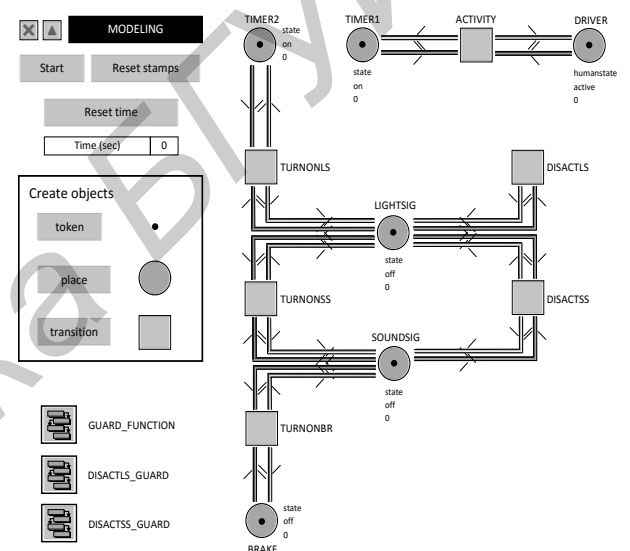


Рисунок 1 – Модель системы экстренного торможения поезда

Разработка прототипа инструментария в среде *G2* и создание на его базе моделей процессов позволили подтвердить целесообразность применения РСП РВ ТЛА для реализации моделей сложных ДС. Однако использование среды *G2*, несмотря на удобство разработки, существенно сужает область применимости полученного программного продукта. Поэтому было принято решение разработать расширенную версию инструментария на языках высокого уровня в среде *Microsoft Visual Studio (MVS)*.

2.2. Разработка на языках высокого уровня

Среда *MVS*, в частности, ориентирована на разработку приложений для платформы *.NET Framework*, основой которой является общезыковая среда исполнения *Common Language Runtime*, подходящая для разных языков программирования. Функциональные возможности этой среды доступны в любых языках программирования, использующих эту среду. Платформа *.NET* рассчитана на работу под операционными системами семейства *Microsoft Windows*, однако

существуют независимые проекты, позволяющие запускать программы *.NET* на других операционных системах (в частности, семейства *UNIX*).

Другой особенностью *MVS* является возможность использования в одном проекте нескольких языков программирования. Поддержка современных объектно-ориентированных языков высокого уровня, в частности, *C++* и *C Sharp*, делает эту среду перспективным инструментом для разработки наукоемких приложений.

Задачу разработки программы-инструментария можно разбить на две подзадачи:

- разработка функциональной логики работы приложения, «модели», содержащую всю наукоемкую часть задачи;
- разработка графического интерфейса пользователя (*graphical user interface, GUI*) – «представления», с помощью которого разработчик создает и настраивает «модель».

Для разработки *GUI* было принято решение использовать наиболее современную графическую *.NET* систему *Windows Presentation Foundation (WPF)* для построения клиентских приложений *Windows*. В основе *WPF* лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. Графической технологией, лежащей в основе *WPF*, является *DirectX*, в отличие от *Windows Forms*, где используется *GDI/GDI+*. Производительность *WPF* выше, чем у *GDI+* за счёт использования аппаратного ускорения графики через *DirectX*.

Применение *WPF* повлекло за собой решение использовать шаблон *Model View ViewModel (MVVM)* – «модель»-«представление»-«модель представления» [Gossman, 2005]. «Модель представления», с одной стороны, является абстракцией «представления», а с другой – предоставляет данные из «модели». Классический шаблон «модель»-«представление»-«контроллер» (*Model View Controller*), при использовании которого изменения в пользовательском интерфейсе не влияют непосредственно на модель, а предварительно проходят через контроллер, в данном случае неудобен, поскольку в технологии *WPF* широко используется концепция «связывания данных», позволяющая связывать данные с визуальными элементами интерфейса в обе стороны.

В рамках предложенного подхода было разработано 3 группы элементов:

- элементы графического интерфейса (пространство имен *RTCPNtoolkit.View*);
- элементы логики функционирования (*RTCPNtoolkit.Model*);
- элементы, обеспечивающие взаимодействие (*RTCPNtoolkit.ViewModel*).

При этом первые две группы независимы друг от друга, а организация взаимодействия ложится на элементы третьей группы. Связь пространств имен *RTCPNtoolkit.View* и *RTCPNtoolkit.ViewModel* осуществляется преимущественно за счёт применения механизмов связывания данных (*data binding*), связь пространств имен *RTCPNtoolkit.Model* и *RTCPNtoolkit.ViewModel* осуществляется за счёт применения механизмов создания и отслеживания событий (*event-driven programming*). Таким образом, использование шаблона *MVVM* позволило полностью отделить задачу разработки функционирования моделей на основе РСП РВ ТЛА от задачи разработки *GUI*.

Тестирование разработанных инструментальных средств было проведено с использованием комплекса заранее подготовленных примеров. Большая часть этих примеров представляет собой искусственные конструкции, позволяющие проверить корректность разработанных алгоритмов функционирования моделей на основе РСП РВ ТЛА, в частности, алгоритмов определения допустимости перехода и разрешения переходов. Проведенное функциональное тестирование позволило усовершенствовать ряд шагов алгоритмов, повысив производительность программного продукта. На разработанный программный комплекс получено свидетельство о государственной регистрации программы для ЭВМ № 2015616435 «Инструментарий для разработки моделей систем на основе темпоральных сетей Петри с поддержкой логики Аллена».

3. Применение инструментария

Предложенный формальный аппарат РСП РВ ТЛА и соответствующий программный инструментарий на его основе был применен в рамках НИР по модернизации электродинамической модели МЭИ (ЭДМ МЭИ), проводимых в плане приоритетных исследований по «интеллектуальной энергетике» с созданием комплекса соответствующих ИСПП РВ для мониторинга, диагностики и управления энергообъектами.

ЭДМ МЭИ позволяет адекватно воспроизводить различные нормальные и аварийные режимы электроэнергетической системы (ЭЭС) и предназначена для исследования поведения новых управляемых элементов ЭЭС, для испытания натуральных устройств автоматического регулирования, противоаварийного управления, релейных защит в условиях, максимально приближенных к реальным, а также проверки и доработки алгоритмов для микропроцессорных регуляторов и устройств [Шаров и др., 2005]. Однако устаревание оборудования и аппаратных компонентов создает сложности в обслуживании и настройке алгоритмов управления. Поэтому в настоящее время проводятся масштабные работы по модернизации оборудования и, в частности, замене аналоговых устройств программируемыми контроллерами.

В рамках проводимых исследований особое внимание уделяется разработке моделей объектов ЭЭС. Разработка осуществляется с применением современных контроллеров *CompactRIO* – многофункциональной реконфигурируемой встраиваемой системой сбора данных и управления. При программировании контроллеров применяется *LabVIEW*, графическая среда разработки и платформа для выполнения программ. В отличие от *SCADA*-систем, которые ориентированы на разработку автоматизированных систем управления технологическим процессом, среда *LabVIEW* в большей степени ориентирована на решение задач в области автоматизированных систем научных исследований.

Программа *LabVIEW* является виртуальным прибором (ВП). В процессе моделирования объекта ЭЭС разрабатываются модели отдельных составляющих элементов этого объекта, каждая из которых является проектом (группой ВП) в среде *LabVIEW*. Архитектура таких приложений соответствует архитектуре контроллеров *CompactRIO* и представлена на рисунке 2. Каждое приложение использует три процессора:

- процессор ПК под управлением *Windows*;
- процессор блока *CompactRIO* под управлением ОС жесткого реального времени;
- программируемую логическую интегральную схему (ПЛИС).

Рациональным подходом является разбиение общей задачи моделирования на отдельные подзадачи для каждого процессора, в соответствии с функциональными характеристиками этих подзадач и техническими характеристиками процессоров. Обработка поступающих (или моделируемых) сигналов осуществляется на уровне ПЛИС, контроллер реального времени обрабатывает задачи, связанные с управлением ВП нижнего уровня. ВП на *Windows* используется для задания параметров моделирования и управления всем приложением.

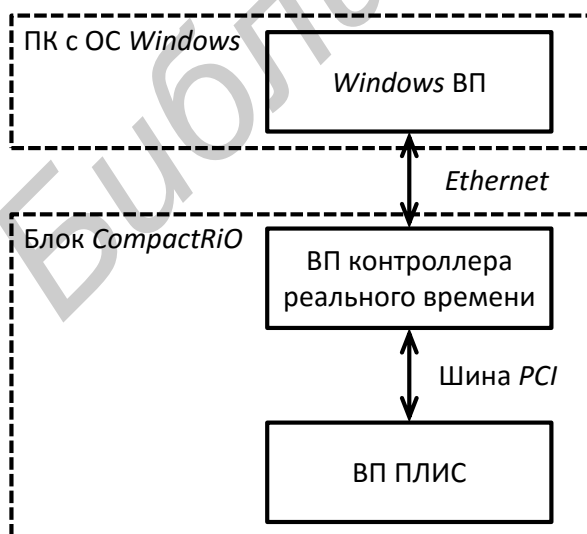


Рисунок 2 – Архитектура приложения *LabVIEW* для моделирования отдельного элемента объекта ЭЭС

Такая архитектура позволяет отдельно разрабатывать каждый элемент моделируемого объекта с использованием современных методов и подходов из области ЭЭС. Однако при этом возникает задача объединения таких подмоделей в единую модель объекта сети, для решения которой был применен подход на основе РСП РВ ТЛА. Архитектура разработанного приложения приведена на рисунке 3.

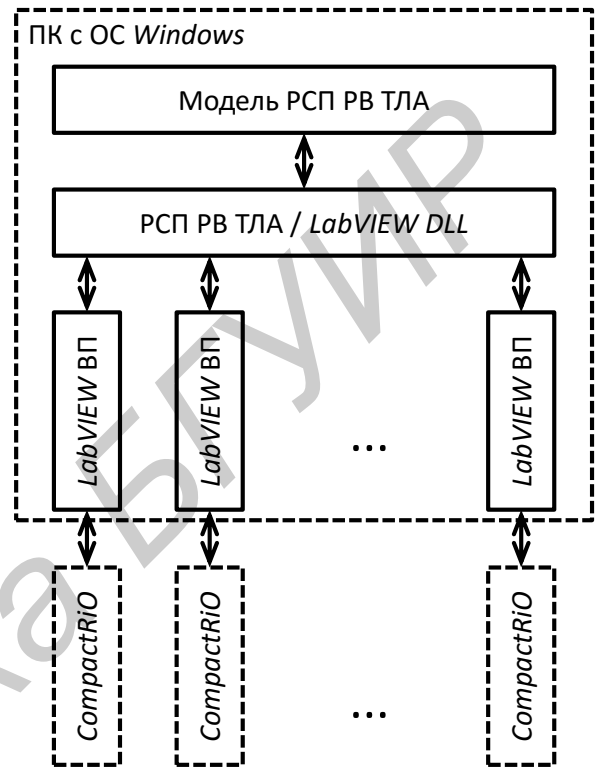


Рисунок 3 – Архитектура приложения для моделирования объекта ЭЭС

Проблема организации взаимодействия между ВП верхнего уровня и созданным инструментарием решается с помощью библиотеки *DLL* (*Dynamic Link Library*) – промежуточной библиотеки, позволяющей осуществлять двухстороннюю связь. При этом *DLL*-библиотека поддерживает работу с разделяемой памятью (*shared memory*), что позволяет увеличить скорость прохождения данных между отдельными элементами приложения. Данный подход предпочтителен, так как в его основе лежит модульный принцип: можно легко заменить одну *LabVIEW*-модель элемента объекта другой (при этом необходимо контролировать только интерфейсную часть ВП, который взаимодействует с промежуточной библиотекой). Поэтому предложенная архитектура удобна при проведении научных исследований в области ЭЭС.

Архитектура приложения позволяет использовать разработанную связующую сеть РСП РВ ТЛА в двух режимах работы:

- в режиме тестирования и отладки, когда сеть РСП РВ ТЛА рассматривается отдельно от *LabVIEW*-моделей, параметры функционирования

которых задаются с помощью вспомогательных подсетей. Использование формул ТЛА позволяет разрабатывать модели таким образом, чтобы эти подсети были не связаны напрямую (дугами) с основной сетью модели;

- в режиме моделирования, когда связующая сеть РСП РВ ТЛА получает данные непосредственно из *LabVIEW*-моделей. При этом модель фактически включается в контур управления и реализуется принцип модельно-управляемой разработки (*model-driven development*).

Предложенная архитектура применима не только для решения задачи моделирования объектов ЭЭС, но и в других приложениях, в том числе и в составе ДИС, а также при разработке их подсистем моделирования. Эксперименты (испытания) показали эффективность подхода к разработке моделей сложных ДС на основе аппарата РСП РВ ТЛА на примере моделирования функционирования объектов ЭЭС для использования в составе ДИС типа ИСППР РВ.

Заключение

В статье затронут вопрос разработки программного обеспечения для создания моделей сложных ДС на основе модификации РСП. Дано краткое описание базового формализма, приведено описание прототипа инструментария для моделирования на основе РСП РВ ТЛА, разработанного в среде *G2*. Функционирование программного приложения проиллюстрировано на примере. Отмечены недостатки комплекса *G2* – высокая стоимость комплекса и существенное сужение области применимости разрабатываемого приложения.

Рассмотрена полная и достаточно универсальная в плане как применения в составе ДИС типа ИСППР РВ семиотического типа, так и автономного использования, версия программного продукта в среде *Microsoft Visual Studio* на языке *C Sharp*. Предложена архитектура разработанного инструментария на основе шаблона *Model-View-ViewModel*. Приведен пример использования формализма РСП РВ ТЛА и инструментария моделирования на его основе в рамках работ по разработке моделей объектов ЭЭС. Предложена модульная архитектура приложения, показано, что модели на основе РСП РВ ТЛА удобно использовать при проведении научных исследований в области ЭЭС. Сделан вывод о том, что подход, при котором сеть РСП РВ ТЛА используется для объединения и организации взаимодействия отдельных модулей, можно использовать и в других задачах как автономно, так и в составе ДИС типа ИСППР РВ.

Работа выполнена при поддержке Министерства образования и науки Российской Федерации (государственное задание № 2.737.2014/К), РФФИ (проекты №14-01-00427, 15-07-04574, 16-37-00309) и Фонда содействия инновациям.

Библиографический список

- [Вагин и др., 2001] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // *Известия РАН. Теория и системы управления*, № 6, 2001. – С. 114-123.
- [Вагин и др., 2008] Вагин В.Н., Еремеев А.П. Исследования и разработки кафедры прикладной математики по конструированию интеллектуальных систем поддержки принятия решений на основе нетрадиционных логик // *Вестник МЭИ*, № 5, 2008. – С. 16-26.
- [Еремеев и др., 2013] Еремеев А.П., Королев Ю.И. Реализация интеллектуальных систем реального времени на основе сетей Петри с поддержкой темпоральных зависимостей // *Программные продукты и системы*, № 3 (103), 2013. – С. 88-94.
- [Еремеев и др., 2015a] Еремеев А.П., Королев Ю.И., Кутепов В.П., Фальк В.Н. Формализм сетей Петри с поддержкой темпоральных зависимостей // *Труды Конгресса по интеллектуальным системам и информационным технологиям "IS&IT'15"*. Научное издание в 3-х томах. – Таганрог: Изд-во ЮФУ, 2015. – Т. 2. – С. 118-131.
- [Еремеев и др., 2015b] Еремеев А.П., Королев Ю.И. Анализ и верификация моделей процессов в сложных динамических системах // *Искусственный интеллект и принятие решений*. – 2015. – № 1. – С. 45-56.
- [Осипов, 2008] Осипов Г.С. Динамические интеллектуальные системы // *Искусственный интеллект и принятие решений*, № 1, 2008. – С. 47-54.
- [Шаров и др., 2005] Шаров Ю.В., Бейм Р.С., Карымов Р.Р., Сыромятников С.Ю. Электродинамическая модель электроэнергетических систем // *Электрические станции*, № 5, 2005. – С. 58-63.
- [Allen, 1983] Allen J.F. Maintaining knowledge about temporal intervals // *Communications of the ACM*. – 1983. – Vol. 26. – No. 11. – P. 832-843.
- [Gossman, 2005] Gossman J. Introduction to Model/View/ViewModel pattern for building WPF apps [Электронный ресурс] // *MSDN Blogs*. [сайт]. [2005]. URL: <http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx> (дата обращения: 07.04.2015).
- [Jensen, 1992] Jensen K. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. – Vol 1-3. – Springer-Verlag, 1992-1997.
- [Szyrka, 2008] Szyrka M. Modelling and Analysis of Real-Time Systems with RTCP-Nets // *Petri Net, Theory and Applications*. I-Tech Education and Publishing. – 2008. – P. 17-40.

DEVELOPMENT OF THE SOFTWARE FOR SYSTEMS MODELLING BASED ON TEMPORAL PETRI NETS

Eremeev A.P., Korolev Y.I.

National Research University «Moscow Power Engineering Institute», Moscow, Russia

Eremeev@appmat.ru

KorolevYu@gmail.com

In the paper the software development of the toolkit for complex dynamic system modelling that allows to operate with temporal dependencies is described. Modelling is based on the formalism of real-time coloured Petri nets supporting Allen's logic. The toolkite prototype designed in *G2* IDE and the development of software in high level languages are considered. The example of using the designed software to model objects of electric power system is given. The possibility of using the designed software in intelligent decision support systems is mentioned.

Keywords: intelligent decision support system, temporal Petri nets, system modelling, toolkit design.



УДК 007:519.816

РАЗРАБОТКА МЕТОДОВ РАССУЖДЕНИЙ НА ОСНОВЕ ТЕМПОРАЛЬНЫХ ПРЕЦЕДЕНТОВ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Куриленко И.Е.

*Национальный исследовательский университет «МЭИ»,
г. Москва, Россия*

ivan@appmat.com

Рассматривается расширение возможностей средств рассуждений на основе прецедентов возможностью учета фактора времени. Работа выполнена при финансовой поддержке РФФИ.

Ключевые слова: интеллектуальная система поддержки принятия решений, моделирование правдоподобных рассуждений, рассуждения на основе темпоральных прецедентов, темпоральная логика.

Введение

В настоящее время значительное внимание уделяется разработке систем, ориентированных на решение задач в открытых и динамических предметных областях (ПО) [Рыбина, 2010, Вагин и др., 2001, Вагин и др., 1999, Лисьев и др., 2011]. Следует отметить, что большинство таких практических задач плохо формализованы [Варшавский и др., 2006]. При поиске решения плохо формализованных задач возникает необходимость применения правдоподобных методов вывода, позволяющих найти приемлемое решение [Варшавский и др., 2009, Montani at all, 2013]. Один из таких методов базируется на том факте, что человеку свойственно на первом этапе при решении новой, неизвестной задачи пытаться использовать решения, которые принимались ранее в подобных случаях и при необходимости адаптировать их к рассматриваемому [Eremeev at all, 2013]. Данный подход с использованием накопленного опыта лег в основу методов и моделей рассуждений на основе аналогий и прецедентов (CBR — Case-Based Reasoning) [Варшавский и др., 2006, 2009, Montani at all, 2013].

Данные методы хорошо проработаны и внедрены на практике, однако во многих случаях при выводе используются «мгновенные» снимки ключевых параметров, а история их изменения не учитывается. Учитывая, что природа физических процессов такова, что к одному и тому же значению ключевых параметров контролируемый процесс может перейти разными путями, которые и будут во многом определять дальнейшую динамику их изменения, остро встает вопрос о разработке новых

методов рассуждений с учетом фактора времени [Eremeev at all, 2009, 2013, Варшавский и др., 2012].

В рамках данной работы рассматриваются формальные модели, расширяющие метод рассуждений на основе прецедентов (на примере метода ближайшего соседа) возможностью учета поведения контролируемого объекта во времени [Борисов и др., 2005, Куриленко и др., 2009]. При учете фактора времени появляется возможность рассмотреть проблемную ситуацию в динамике, то есть текущая ситуация сравнивается не с какими-либо фиксированными значениями прецедентов, а отслеживается процесс изменения значений, что позволяет строить предположения не по одному параметру сходства, а более глубоко проанализировать проблемную ситуацию.

1. Темпоральное расширение метода ближайшего соседа

Наиболее простым способом адаптации существующих алгоритмов вывода на основе прецедентов к учету фактора времени является подмена «мгновенного» снимка ключевых параметров «историей» их изменения. Эта история может быть представлена в виде матрицы значений, составленной на определенный период наблюдения с определенным интервалом (тактом).

Для вывода в этом случае могут использоваться имеющиеся алгоритмы и методы. Для этого матрица перепаковывается в вектор путем последовательной записи в него элементов ее столбцов. Таким образом, в прецедент включаются не только значения параметров в текущий момент времени, но и их значения на некоторый период времени до

этого. При этом эксперту в интерфейсе программы может представляться более удобная для интерпретации и заполнения матричная форма, а ее преобразование в массив для применения того или иного CBR-метода система может выполнять автоматически.

Иной способ внедрения фактора времени в процесс рассуждений основывается на учете решений, полученных на прошлых этапах при выборе решения на текущем этапе. Проиллюстрируем его на примере расширения метода ближайшего соседа. В первую очередь следует определить отрезок времени для анализа проблемной ситуации (глубину анализа). Далее рассматриваемый промежуток времени разбивается на N равных отрезков с некоторым шагом (тактом).

В сформированной базе прецедентов, где каждый прецедент представляется в параметризованном виде [Варшавский и др, 2006], происходит сравнение значений параметров ситуации в каждый момент времени i , где $i = 0, 1, \dots, N$. Значения параметров прецедентов сравниваются между собой по методу ближайшего соседа с выбранной метрикой, к примеру, Евклидовой. Для всех точек i получаются варианты прецедентов с определенными интегральными оценками, которые могут быть выбраны как соответствующие ситуации в этой точке для текущего значения порога подобия. На следующем этапе по интегральным оценкам в моментах i выбирается наиболее вероятный прецедент или группа прецедентов, удовлетворяющих поисковому условию. Каждому результирующему прецеденту в соответствие ставится мера сходства, которая определяется по правилу: «При совпадении всех параметров в описании прецедента и текущей ситуации степень сходства будет равна 1, а каждый совпавший параметр дает вклад равный $1/n$, где n – число параметров в описании прецедента и текущей ситуации».

Отметим, что используя значения параметров в точках i можно с помощью интерполяции построить прогноз развития проблемной ситуации (на уровне параметров).

На результат поиска прецедентов, удовлетворяющих проблемной ситуации, оказывает влияние несколько параметров. Во-первых, это выбор метрики. В каждом конкретном случае этот выбор производится по-разному, в зависимости от целей ЛПР, физической и статистической природы используемой информации при управлении сложным объектом и других ограничений и факторов, влияющих на процесс поиска решения. Во-вторых, влияние оказывает выбор шага, с которым разбивается на промежутки рассматриваемый отрезок времени. Вариация размеров шага дает различное количество дискретных точек, по которым происходит сравнение значений параметров прецедентов. Это оказывает влияние не только на точность выбора прецедента, но и на скорость выполнения

поискового алгоритма, а так же на величину погрешности, с которой может быть построен прогноз, если это необходимо. И, в-третьих, важным оказывается выбор порогового значения, определяющего степень сходства.

Предложенные методы могут быть использованы как по отдельности, так и в комбинации друг с другом. В практических приложениях важно правильно определить величину N и размер такта. При этом следует учитывать вероятность скачкообразного изменения поведения параметров контролируемого объекта или процесса. Если такое возможно, то предпочтительнее использовать первый метод.

2. Использование темпоральной логики при рассуждении на основе прецедентов

Изложенные в предыдущем разделе методы можно улучшить. Очевидное улучшение – переход к моделям, позволяющим хранить меньше информации, допускающих рассуждения с использованием как метрической, так и качественной информации.

Определим метрическую точечную задачу согласования временных ограничений (МЗСВО) как $Z=(V,D,C)$, где $V = \{V_1, V_2, \dots, V_m\}$ – конечное множество временных переменных, соответствующих моментам времени; D – область значений временных переменных (является либо множеством вещественных чисел, либо множеством целых чисел); C – конечное число бинарных временных ограничений вида $C_{ij}=\{[a_i, b_i], \dots, [a_k, b_k]\}$, где интервалы попарно не пересекаются. Бинарные ограничения позволяют определить разрешенную дистанцию между моментами времени и интерпретируются как $(a_i \leq V_j - V_i \leq b_i) \cup \dots \cup (a_k \leq V_j - V_i \leq b_k)$. Для решения задачи требуется скорректировать все ограничения так, чтобы они не противоречили друг другу. Если подобная корректировка возможна МЗСВО считается согласованной (иначе – несогласованной).

Используем формализм МЗСВО для описания наблюдаемых ситуаций и прецедентов. При этом фиксируется как сам факт наступления тех или иных событий, их порядок, так и время их появления (метрика).

Для задания структуры темпорального прецедента на основе МВЛ вводится определение ситуации как $S = \langle Z, P, a \rangle$, где Z – МЗСВО; $P = \{P_1, \dots, P_k\}$ – набор параметров контролируемого объекта; $a: V \rightarrow P$ – функция, сопоставляющая каждой временной переменной (событию) набор параметров, которые характеризовали состояние объекта или процесса в данный момент времени. Прецедент определяется как ситуация, дополненная диагнозом и рекомендациями ЛПР – $\langle Z, P, a, R \rangle$. Далее будем предполагать, что МЗСВО – Z согласованы и преобразованы в минимальный вид. При поиске прецедента для наблюдаемой ситуации

используется алгоритм, учитывающий временные характеристики и состояние контролируемого объекта или процесса в момент наблюдения тех или иных событий.

Предложен обобщенный улучшенный алгоритм извлечения темпоральных прецедентов, построенный для моделей представления прецедентов, использующих временные логики, представляющие информацию о времени в виде временных ограничений. В этом алгоритме на первом этапе проверяется сходство временной структуры. Алгоритм контроля этого сходства может быть разбит на два этапа. На первом проверяется необходимое условие - пересечение каждого ограничения в наблюдаемой ситуации с соответствующим ограничением в прецеденте должно быть не пусто (в противном случае констатируется отсутствие временного сходства). Если на первом этапе не произошло усиления имеющихся ограничений, то необходимости во втором этапе нет, и временное сходство имеется. Если произошло уточнение временных ограничений - то требуется проверить согласованность полученной модели. Если она согласована (непротиворечива), то есть сходство - иначе сходства нет. Алгоритм, используемый на втором этапе может быть улучшен, если использовать методы пошагового уточнения задачи согласования временных ограничений. В некоторых случаях может использоваться более сильное требование идентичности временных ограничений (в этом случае проверка согласованности не требуется). После проверки сходства временной структуры проверяется сходство параметров.

Для вычисления соответствия временной структуры наблюдаемой ситуации и прецедента могут быть предложены несколько методов - метод с жесткими ограничениями и метод с мягкими ограничениями. Метод с жесткими ограничениями предполагает точное соответствие наблюдаемых событий и событий, имеющихся в прецеденте, а также для каждого метрического ограничения C_{ij} в прецеденте его пересечение с ограничением C_{ij}^* в наблюдаемой ситуации должны равняться C_{ij} . При этом для того, чтобы события в прецеденте и анализируемой ситуации соответствовали друг другу, предполагается использовать для их нумерации числа, получаемые в результате сортировки по имени параметра и времени. Этот метод полезен для систем, в которых временные характеристики протекающих процессов (время наступления событий и их порядок) достаточно стабильны. В случае использования метода с мягкими ограничениями снимается условие, что для каждого метрического ограничения C_{ij} в прецеденте его пересечение с ограничением C_{ij}^* в наблюдаемой ситуации должны равняться C_{ij} . Оно заменяется на то, что это пересечение не пусто и C_{ij}^* вложено в C_{ij} . Возможно дальнейшее смягчение условий при анализе степени сходства с учетом изменения истории изменений каждого параметра по отдельности. Тут возможны разные стратегии

(например, учитывающие ограничения между событиями изменения значений разных параметров с большим весом, чем ограничения, наложенные на события полученные из-за изменения одного параметра). Вычисление прецедента на основе выборки похожих ситуаций (обучение) в данном случае может быть выполнено на основе смягчения ограничений C_{ij} в одном из вариантов так, чтобы выполнялись условия его подобия другим ситуациям. Для определения подобия параметров контролируемого объекта или процесса в момент наступления того или иного события может использоваться подходящий на практике метод оценки сходства (например, метод ближайшего соседа). Организация прецедентов на основе метрической ЗСВО позволяет учитывать как последовательности событий, так и их длительности.

Если анализ длительностей не принципиален, а важен порядок событий можно воспользоваться качественной временной логикой - например, точечной временной логикой, для которой в работах [Еремеев и др., 2005, 2007, 2009, 2010, Куриленко 2009, Куриленко и др., 2014] предлагаются быстродействующие алгоритмы вывода. В этом случае для представления прецедента может использоваться точечная ЗСВО, а определение степени сходства наблюдаемой ситуации и прецедента может основываться на решении задачи логической эквивалентности соответствующих минимальных ЗСВО.

Естественным развитием рассмотренных моделей является обеспечение возможности указания в моменты изменения не всех параметров наблюдаемого объекта, а только изменяющихся.

Исходя из практических соображений перспективным выглядит построение модели представления прецедента на основе расширенной точно-интервальной временной логики. Эта логика позволяет работать как с качественной информацией, заданной в виде ограничений «момент-момент», «момент-интервал», «интервал-интервал», так и с метрической информацией: длиной временного интервала и точным положением момента на временной оси. Рассмотрим метод представления прецедентов с использованием выражений расширенной точно-интервальной временной логики. ЗСВО для расширенной точно-интервальной временной логики задается, как $Z=(V, D_1, D_2, C)$, где V - конечное множество временных переменных, соответствующих моментам или интервалам времени; D_1 - область значений временных переменных - моментов времени (множество вещественных или целых чисел), D_2 - область значения временных переменных - интервалов времени (множество пар вещественных или целых чисел); C - конечное число бинарных временных ограничений.

При вычислении соответствия временной структуры наблюдаемой ситуации и прецедента существенно важным является быстродействие

алгоритмов сопоставления событий как временных ограничений. Для ускорения процесса временного вывода решатель использует алгоритмы сокращения перебора на графах, основанные на объединении ребер и вершин, которые имеют идентичную метрическую информацию или могут быть представлены сверткой базовых бинарных ограничений между примитивами точечно-интервальной временной логики. Разработан эффективный способ представления темпоральной информации и временных ограничений, как бинарных масок-векторов, который позволяет компактно хранить темпоральную информацию. Каждое ограничение C_{ij} между временными интервалами в расширенной точечно-интервальной временной логике может быть сведено к четверке точечных временных ограничений между моментами, которые являются граничными точками интервалов. Затем каждое ограничение четверки, являющееся уже точечным ограничением, кодируется особым образом и помещается в битовую маску-вектор. В основе алгоритма вычисления выполнимых ограничений лежит преобразование битовых масок-векторов и получение матрицы дизъюнктивных ограничений. Искомое ограничение C_{ij} является элементом $[i,j]$ данной матрицы после завершения работы алгоритма вывода.

Использование такой структуры темпоральных прецедентов позволяет учитывать одновременно и метрическую и качественную информацию о времени.

Рассмотренные модели легко расширить для возможности обработки неточной или неопределенной информацией. Например, может быть определена временная мера подобия для гетерогенных последовательностей событий, основанных на временной модели вероятностной сети временных ограничений. Как индикатор сходства исходных последовательностей событий в этом случае может использоваться общая неопределенность возможных ограничений. Менее сходные последовательности производят более неопределенные временные ограничения и наоборот. Эти ограничения являются неопределенными вследствие неточности, порожденной временными различиями последовательностей. В качестве основа темпорального прецедента в этом случае используется нечеткая временная логика. Рассмотрим множество событий, связанных с неточностью и неопределенностью знаний и рассматриваемых, как подмножества множества X , где событие наступает. Пустое множество связано с не наступлением события. Определим модель, которая использует потенциал теории возможностей [Zadeh et al, 1978] как средство для представления и рассуждения с неопределенными точечными временными ограничениями. Неопределенность ограничений выражается в неопределенной временной связи. Таким образом, неопределенность представляется в виде вектора с участием трех возможных способов выражения

значения относительной правдоподобности трех основных ограничений до ($<$), в то же время ($=$), и после ($>$), которое может содержаться между двумя точками. Помимо перечисленных существуют дизъюнктивные ограничения «до или в то же время», «после или в то же время», а также возможен случай полного отсутствия информации (когда, любое из трех основных ограничений возможно). Для двух временных точек i и j , вероятностное ограничение r_{ij} между ними представлено в виде нормализованного вектора $r_{ij}=(P_{i<j}, P_{i=j}, P_{i>j})$, где $P_{i<j}$ (соответственно $P_{i=j}, P_{i>j}$) является вероятностью того, что $i<j$ (соответственно, $i=j, i>j$). Над этими векторами определяются базовые операции над временными ограничениями (композиция, отрицание, дополнение и пересечение), что позволяет определить алгебру временных ограничений. Временная модель вероятностной сети временных ограничений является видом задачи согласования временных ограничений (далее ЗСВО) и определяется как $PTCN = \langle TP, R \rangle$, где TP - множество моментов времени, R представляет собой множество возможных временных ограничений, определенных над множеством моментов времени. Используем введенные определения для построения способа представления прецедентов с использованием нечеткой темпоральной логики. Будем описывать ситуацию множеством событий (e). Каждое событие состоит из параметризованного описания области (параметр, атрибут, значение) и временной компоненты, указывающей, когда произошло событие. Прецедент представляет собой набор гетерогенных событий, описывающих временную последовательность и рекомендации. Любая последовательность событий определяется порядком, с учетом временной компоненты. Тем не менее, каждое событие может быть сравнимо с событием из другой последовательности и, следовательно, порядок является одним из наиболее актуальных критериев сходства.

3. Программная реализация

Предлагаемые методы реализованы в прототипе СBR-системы, используемой для диагностики штатных ситуаций в ИС управления крупными парковочными комплексами (ИС УП) [Борисов и др., 2005, Куриленко, 2009]. Операции, протекающие в штатном режиме на точках доступа, в таких комплексах являются однотипными [Куриленко, 2009]. В результате для выделения штатных ситуаций можно применить механизмы вывода на основе темпоральных прецедентов. Разработанная система диагностики с применением средств вывода на основе темпоральных прецедентов внедрена на практике на ряде объектов в России и СНГ.

В ее задачи входит: выделение сбоев в работе ИС УП и подготовка экспертной оценки наблюдаемых сбоев по базе типовых проблемных

ситуаций, определение аномальных ситуаций, которые возникли в процессе эксплуатации ИС УП, но не были учтены при разработке управляющих правил, пресечение попыток противодействия обслуживающего персонала и посетителей. Система реализует функции: поиска и выделения сбоев в работе парковочного комплекса, которые относятся к типовым проблемным ситуациям, а также обеспечивает определение аномальных ситуаций, которые возникли в процессе эксплуатации ИС УП, но не были учтены при разработке управляющих правил.

Созданный прототип может функционировать в трёх режимах:

- режим обучения – обучение ИС УП на основе данных из файлов и знаний экспертов;
- “on-line” - режим – события поступают напрямую на вход системы;
- “off-line” - режим – события читаются из протоколов (логов) работы системы.

Система состоит из блока обучения, блока вывода, базы прецедентов, и трёх интерфейсов: интерфейса взаимодействия с объектом, экспертом и пользователем. ИС содержит базу типовых ситуаций (как штатных, так и известных нештатных). Решатель на основе прецедентов соотносит временные диаграммы в последовательностях операций за определенный период с моделями, хранящимися в этих базах. В случае, если выявляется типовая нештатная ситуация, ее данные заносятся в базу нештатных ситуаций. Если же выделяется ситуация, которая не описана ни в базе штатных типовых ситуаций, ни в базе нештатных типовых ситуаций, то эта ситуация заносится в отдельную базу, содержимое которой анализируется разработчиками (экспертами) ИС УП.

Проведенные эксперименты показали применимость предложенных методов на практике.

Заключение

Применение в составе современных ИС методов построения рассуждений на основе темпоральных прецедентов позволяет находить более точные и корректные решения, сделать прогнозирование динамики развития ПО и, таким образом, расширить круг решаемых задач. Интеграция расширенных методов рассуждений на основе прецедентов с учетом фактора времени в ИСППР для мониторинга и управления различными сложными техническими объектами позволит лицам, принимающим решения своевременно выбрать необходимую стратегию поведения для устранения проблемной ситуации за счет сокращения поискового пространства и уменьшения времени реакции системы.

Библиографический список

- [Вагин и др., 2001] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // Изв. РАН. Теория и системы управления. 2001. № 6. - С. 114-123.
- [Рыбина, 2010] Рыбина Г.В. Основы построения интеллектуальных систем. — М.: Финансы и статистика; ИНФРА-М, 2010. — 432 с.
- [Лисьев и др., 2011] Лисьев Г.А., Попова И.В. Технологии поддержки принятия решений. — 2-е издание, стереотипное. — М.: Издательство «ФЛИНТА», 2011. — 133 с.
- [Вагин и др., 1999] Вагин В.Н., Еремеев А.П. Конструирование интеллектуальных систем поддержки принятия решений реального времени // Интеллектуальное управление: новые интеллектуальные технологии в задачах управления (ИСТ'99). Труды междунар. конф., Переславль-Залесский, 6-9 декабря 1999. — М.: Наука. Физматлит, 1999.
- [Варшавский и др., 2006] Варшавский П.Р., Еремеев А.П. Методы правдоподобных рассуждений на основе аналогий и прецедентов для интеллектуальных систем поддержки принятия решений // Новости искусственного интеллекта. — 2006. — №3. — С.39-62.
- [Варшавский и др., 2009] Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений. Искусственный интеллект и принятие решений, №2, 2009, с. 45-57.
- [Eremeev at all, 2013] Eremeev A.P., Kurilenko I.E. Методы моделирования временных зависимостей в интеллектуальных системах с использованием темпоральных прецедентов // Information Models and Analyses Vol. 2, 2013, № 4, с. 324-335.
- [Варшавский и др., 2012] Варшавский П.Р., Еремеев А.П., Куриленко И.Е. Моделирование временных зависимостей в интеллектуальных системах поддержки принятия решений на основе прецедентов // Information technologies and knowledge Vol. 6, 2012, № 3, с. 279-294.
- [Montani at all, 2013] Stefania Montani, Lakhmi C. Jain. Successful Case-based Reasoning Applications-2. — Springer, 2013.
- [Куриленко, 2009] Куриленко И.Е. Применение временной логики при построении интеллектуальной системы управления крупными парковочными комплексами // Сб. док. научно-практ. конф. ИММВИ 2009 в 2 т. — Т.2 — М.:ФизМатЛит, 2009. С. 171 - 180.
- [Борисов и др., 2005] Борисов А.В., Куриленко И.Е. О современных подходах к построению систем учета автотранспорта. Программные и аппаратные средства. // Информационные технологии моделирования и управления. 2005. №5. С. 786-794.
- [Еремеев и др., 2007] Еремеев А.П., Куриленко И.Е. Реализация механизма временных рассуждений в современных интеллектуальных системах // Изв. РАН. Теория и системы управления, 2007, № 2, с. 120–136.
- [Куриленко, 2009] Куриленко И.Е. Применение временной логики при построении интеллектуальной системы управления крупными парковочными комплексами // Сб. док. научно-практ. конф. ИММВИ 2009 в 2 т. — Т.2 — М.:ФизМатЛит, 2009. С. 171 - 180.
- [Eremeev at all, 2009] Alexander Eremeev, Ivan Kurilenko, Pavel Varshavskiy. Application of Temporal Reasoning and Case-based Reasoning int Intelligent Decision Support Systems // International Book Series «Information science & computing», Number 10, Supplement to International Journal «Information technologies & knowledge» Volume 3/2009 – 2009. – Pp. 9–16.
- [Еремеев и др., 2005] Еремеев А.П., Куриленко И.Е. Реализация временных рассуждений для интеллектуальных систем поддержки принятия решений реального времени // Программные продукты и системы. – 2005. – №2. – С.8-16.
- [Еремеев и др., 2010] Еремеев А.П., Куриленко И.Е. Применение темпоральных моделей в интеллектуальных системах / Интеллектуальные системы. Коллективная монография. Выпуск четвертый. / Под. Ред. В.М. Курейчика. — М.: Физматлит, 2010, 300 с., 222-252.
- [Еремеев и др., 2009] Еремеев А.П., Куриленко И.Е. Компонента временных рассуждений для интеллектуальных систем поддержки принятия решений реального времени // Искусственный интеллект и принятие решений, 2009, № 1, с. 31–45.

[Куриленко, 2009] Куриленко И.Е. Подход к решению метрических задач согласования временных ограничений в подклассе простых темпоральных задач (Simple Temporal Problems (STP)) // Труды XVII междунар. науч.–техн. конф. «Информационные средства и технологии».–М.: Издательство МЭИ, 2009.–Т.1.–С.142–149.

[Еремеев и др., 2008] Еремеев А.П., Куриленко И.Е. «Моделирование временных рассуждений в интеллектуальных системах реального времени» // Вестник МЭИ, 2008. №1. С. 114-123.

[Куриленко, 2009] Куриленко И.Е. Система временного вывода для интеллектуальных систем поддержки принятия решений реального времени // Сб. док. междунар. научно-практ. конф. Интегрированные модели и мягкие вычисления в искусственном интеллекте в 2 т. – Т.1. – М.:ФизМатЛит, 2009. С 241 – 252.

[Куриленко и др., 2014] Куриленко И.Е., Шорникова Д.А. Реализация подсистемы временного вывода на базе точечно-интервальной временной логики для систем поддержки принятия решений реального времени // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2014 (24-27 сентября 2014 г., г. Казань, Россия): Труды конференции. Т.1. – Казань: Изд-во РИЦ «Школа», 2014. – С.50-58.

[Zadeh et al, 1978] Zadeh, Lotfi, «Fuzzy Sets as the Basis for a Theory of Possibility», *Fuzzy Sets and Systems* 1:3-28, 1978. (Reprinted in *Fuzzy Sets and Systems* 100 (Supplement): 9-34, 1999.)

ON TEMPORAL MODELS FORMALISATION FOR INTELLIGENT SYSTEMS

Kurilenko I.E.

*National Research University «Moscow Power
Engineering Institute», Moscow, Russia*

ivan@gmail.com

The paper describes the temporal reasoning extension of the case-based reasoning. This work was financially supported by RFBR.

Keywords: intelligent decision support system, case-based reasoning, temporal logic, temporal case-based reasoning.



УДК 519.816

СПЕЦИФИКАЦИЯ ПОТОКА ЗАДАЧ В РАСШИРЕННОЙ ИНТЕРВАЛЬНОЙ ЛОГИКЕ АЛЛЕНА

Плесневич Г.С., Нгуен Тхи Минь Ву

Национальный исследовательский университет МЭИ, г. Москва, Россия

salve777@mail.ru

mimhvu.357@gmail.com

В работе введена расширенная интервальная логика Аллена и показано как эта логика может быть использована для спецификации и анализа потоков работ (workflows). Предложениями расширенной логики служат булевы комбинации предложений логики Аллена содержащие, возможно, метрическую информацию. В частности, этот анализ включает: (1) распознавание несовместности (противоречивости) схемы потока работ; (2) вычисление ответа на запросы к непротиворечивому потоку работ.

Ключевые слова: потоки работ, спецификация потоков работ, темпоральные логики, интервальная логика Аллена

отгрузка товара и оформление платежки.

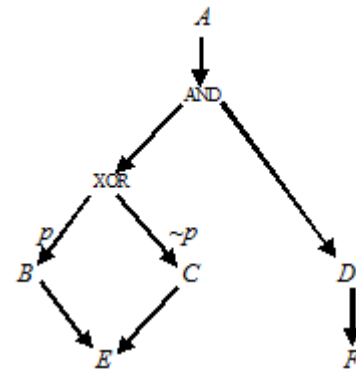
Введение

Поток работ (workflow) – это система скоординированных и взаимодействующих друг с другом активностей, предназначенная для выполнения сложного процесса, например, бизнес-процесса на предприятии [van der Aalst et al., 2002]

Активность (activity) в потоке работ может совершаться человеком, устройством или программой. Активность рассматривается как минимальная часть процесса, т.е. она соответствует одному шагу осуществления процесса. Каждая активность ассоциирована с *задачей (task)*. В потоке работ задачи выполняются в порядке, который определяется временными зависимостями между ними. Таким образом, поток работ содержит информацию об этом порядке в виде некоторой схемы задач.

Рассмотрим очень простой пример схемы потока работ (см. Рис.1).

Пример 1. Поток работ представляет бизнес-процесс, целью которого является перевозка товара. Эта перевозка должна осуществляться одной из двух фирм F1 или F2. Фирма F1 используется для перевозки тогда и только тогда, когда выполняется некоторое условие p , выражающее предпочтение для выбора этой фирмы. Бизнес-процесс начинается с оформления заказа (задача A). Затем оформляется накладная (задача D) и осуществляется перевозка товара (задача F1 или F2). Наконец, осуществляется



- A: обработка заказа
- B: перевозка фирмой F1
- C: перевозка фирмой F2
- D: оформление накладной
- E: отгрузка товара
- F: оформление платежа
- p : условие, определяющее выбор фирмы F1

Рисунок 1 -- Пример схемы потока работ

Выполнение каждой задачи из этой схемы требует некоторого времени, т.е. с задачами ассоциированы темпоральные интервалы, которые мы обозначим теми же буквами, что и соответствующие задачи. Ясно, что для интервалов имеют место следующие временные ограничения: (i) если p , то A раньше B; (ii) если $\sim p$, то A раньше C; (iii) D раньше F; (iv) B раньше E; (v) C раньше E.

Мы можем также предполагать, что оформление платежки выполняется во время отгрузки, но раньше ее окончания. Для любого интервала X обозначим X^- и X^+ его начало и конец. Поэтому, если для любого интервала X обозначить через X^- и X^+ его начало и конец, то для интервалов E и F будем иметь утверждение (vi) $E^- = F^-$ и $E^+ < F^+$ или $E^- < F^-$ и $E^+ < F^+$.

Утверждения (i), (ii), ..., (vii) можно представить следующими формулами расширенной логики Аллена: (i) $p \rightarrow A \ b \ B$; (ii) $\sim p \rightarrow A \ b \ C$; (iii) $A \ b \ D$; (iv) $B \ b \ E$; (v) $C \ b \ E$; (vi) $E \ f \ F \vee E \ d \ F$. Последняя формула в логике Аллена записывается короче как $E \ f d \ F$. (Расширенная логика Аллена излагается далее в разделе 2.)

Метрическая информация в схема потока задач выражается равенствами и неравенствами вида $X - Y = r$, $X - Y \leq r$, $X - Y < r$, $X - Y \geq r$, $X - Y > r$, где X и Y – концы интервалов, участвующих в схеме, а r – число, измеряющее время. Мы будем предполагать, что время дискретно, и что его моментами служат натуральные числа $0, 1, 2, \dots$. Например, в схеме потока работ на Рис.1 мы можем предполагать, что $E^+ - E^- \leq 20$ и $E^+ - E^- \geq 15$, т.е. что отгрузка длится от 15 до 20 единиц времени. В расширенной логике Аллена эту метрическую информацию можно записать как $15 \leq E^+ - E^- \leq 20$.

В настоящей статье мы сформулируем расширенную интервальную логику Аллена, изложим основанный на аналитических таблицах метод дедукции для этой логики и покажем, как этот метод можно использовать для анализа потоков задач. Этот анализ включает: (1) распознавание несовместности (противоречивости) схемы потока работ; (2) вычисление ответа на запросы к непротиворечивому потоку работ.

1) Для анализа потоков работ использовались темпоральные логики. Например, в работе [Wang et al., 2004], рассмотрено применение разработанной Лампортом логики действий [Lamport, 1991]. Повидимому, однако, логика Лампорта достаточно сложна, и не имеет эффективных методов дедукции.

Замечания. 1) Для анализа потоков работ использовались темпоральные логики. Например, в работе [Wang et al., 2004], рассмотрено применение разработанной Лампортом логики действий [Lamport, 1991]. Повидимому, однако, логика Лампорта достаточно сложна, и не имеет эффективных методов дедукции.

2) Сфера применения потоков работ не ограничивается бизнес-процессами. Вообще, поток работ можно рассматривать как формальное средство для представления любых сложных распределенных процессов, состоящих из элементарных подпроцессов (активностей), для координации этих подпроцессов и для управления ими. Например, современные естественно-научные исследования являются сложными процессами такого рода. В связи с этим появилось понятие

научного потока работ (scientific workflow) [Gil et al., 2007]. Существенные приложения техники потоков работ появились в области биоинформатики (см., например, [Matschiner et al., 2005])

1. Расширенная интервальная логика Аллена

В интервальной логике Аллена, обозначаемой **LA**, рассматриваются 7 базовых интервальных отношений с их инверсиями [Allen, 1983]. Эти отношения обозначаются так: **b** (before – раньше), **m** (meets – встречается), **o** (overlaps – перекрывает), **f** (finishes – заканчивает), **s** (starts – начинает), **d** (during – в течение), **e** (equates – равняется). Инверсия отношения обозначается звёздочкой «*» (таким образом, например, $A \ o^* \ B$ имеет место тогда и только тогда, когда верно $B \ o \ A$). Интерпретация этих отношений показана в табл.1, где A, B означают временные интервалы, а A^-, A^+, B^-, B^+ – их концы. Мы предполагаем, что интервалы не вырождаются в точки, т.е. всегда $A^- < A^+$ для любого интервала A .

Предложениями логики Аллена **LA** являются выражения вида $A \ \omega \ B$, где ω – любое подмножество множества $\Omega = \{b, b^*, m, m^*, o, o^*, f, f^*, s, s^*, d, d^*, e\}$. (Таким образом, при фиксированных A и B имеется ровно 2^{13} различных предложений логики Аллена.) Атомарное предложение логики **LA** – это предложение of the form $A \ \omega \ B$ с одноэлементным подмножеством ω , т.е. оно имеет вид $A \ \{a\} \ B$, $a \in \Omega$ (но обычно фигурные скобки не ставятся).

Таблица 1 -- Интерпретация базовых отношений

Базовые отношения Аллена	Иллюстрация	Неравенства и равенства для концов интервалов
$A \ b \ B$	==A== ==B==	$A^+ < B^-$
$A \ m \ B$	==A== ==B==	$A^+ = B^-$
$A \ o \ B$	===A=== ==B==	$A^- < B^-, B^- < A^+, A^+ < B^+$
$A \ d \ B$	==A== =====B=====	$B^- < A^-, A^+ < B^+$
$A \ s \ B$ $B \ s^* \ A$	===A=== =====B=====	$A^- = B^-, A^+ < B^+$
$A \ f \ B$ $B \ f^* \ A$	===A=== =====B=====	$B^- < A^-, A^+ = B^+$
$A \ e \ B$	=====A===== =====B=====	$A^- = B^-, A^+ = B^+$

Предложение $A \ \omega \ B$ интерпретируется как дизъюнкция $\vee \{A \ a \ B \mid a \in \omega\}$. Поэтому, например, имеют место эквивалентности: $A \ o^* \ d \ s \ B \approx A \ o^* \ B \vee A \ d \ B \vee A \ s \ B \approx B \ o \ A \vee A \ d \ B \vee A \ s \ B$. Здесь $o^* \ d \ s$ – сокращенная запись для подмножества $\{o^*, d, s\}$.

Как и во всякой логике, в логике Аллена имеется отношение \models логического следования. Пусть O – онтология в языке LA , т.е. конечное множество предложений LA : $O = \{A_1 \omega_1 B_1, A_2 \omega_2 B_2, \dots, A_m \omega_m B_m\}$, и $A \omega B$ – произвольное предложение LA , где $A, B \in \{A_1, B_1, A_2, B_2, \dots, A_m, B_m\}$. По определению $A \omega B$ есть логическое следствие из онтологии O (в записи: $O \models A \omega B$), если нет такого назначения символом A_i ($1 \leq i \leq m$) конкретных темпоральных интервалов, при котором все предложения $A_i \omega_i B_i$ ($1 \leq i \leq m$) истинны, а предложение $A \omega B$ ложно.

Выразительность логики LA можно увеличить, если добавить к атомарным предложениям метрическую информацию. Возьмем, например, атомарное предложение $A o B$. Оно характеризуется неравенствами $A^- < B^-$, $B^- < A^+$, $A^+ < B^+$ (см. третий столбец Табл.1) и $A^- < A^+$, $B^- < B^+$, которые можно интерпретировать как то, что отрезки $[A^-, B^-]$, $[B^-, A^+]$, $[A^+, B^+]$ и $[B^+, A^+]$ имеют положительные длины. Естественное обобщение состоит в задании длин этих отрезков или в задании оценок для них. Например, если нам известно, что интервал A имеет длину 6, интервал B – длину 5, а длина перекрытия интервалов A и B не меньше 2 и не больше 3, то мы имеем следующие равенства и неравенства: $A^+ - A^- = 6$, $B^+ - B^- = 5$, и $2 \leq A^+ - B^- \leq 3$. Эту информацию мы представляем выражением $A o(A^+ - A^- = 6, B^+ - B^- = 5, 2 \leq A^+ - B^- \leq 3) B$, которое будем считать предложением с метрической информацией в языке, являющимся расширением языка LA . Этот язык обозначим μLA . Синтаксис языка μLA определяется так:

- *атомарное ограничение* – это равенство или неравенства вида $X - Y = r$, $X - Y > r$, $X - Y \geq r$, $X - Y < r$ и $X - Y \leq r$, где X, Y – концы интервалов, $X \neq Y$ и r – натуральное число;
- *произвольное ограничение* – это конъюнкция атомарных ограничений, причем точка с запятой используется как знак для конъюнкции;
- *связка Аллена с метрическими ограничениями* – это выражение вида $\alpha(\lambda)$, где $\alpha \in \Omega$ и λ – ограничение. При этом ограничение λ должно быть согласовано со связкой α в следующем смысле: (а) если $\alpha = b$, то λ использует только разности $A^+ - A^-$, $B^+ - B^-$, $B^- - A^+$; (б) если $\alpha = f$, то λ использует только разности $A^+ - A^-$, $B^+ - B^-$, $A^+ - B^+$, $A^- - B^-$; (в) и т.д.;
- *предложение логики μLA* – это выражение вида $A \alpha_1 \alpha_2 \dots \alpha_k B$, где α_k – связка Аллена или связка Аллена с метрическими ограничениями.

Наконец, введем расширенную интервальную логику Аллена $\mu LA+$, предложения которого получаются из предложений языка μLA применением булевых связок \sim , \wedge и \vee . Язык $\mu LA+$ имеет следующий синтаксис:

- пропозициональные переменные p, q, p_1, p_2, \dots есть предложения $\mu LA+$;
- атомарные ограничения есть предложения $\mu LA+$;
- предложения μLA есть предложения $\mu LA+$;

• если ϕ и ψ – предложения μLA , то $\sim\phi$, $\phi \wedge \psi$ и $\phi \vee \psi$ – предложения $\mu LA+$. Выражение $\phi \rightarrow \psi$

Таблица 2 -- Правила вывода для пропозициональных связок

$\frac{+ \sim \phi}{- \phi}$	$\frac{- \sim \phi}{+ \phi}$	$\frac{+ \phi \wedge \psi}{+ \phi}$ $\frac{+ \phi \wedge \psi}{+ \psi}$	$\frac{- \phi \wedge \psi}{- \phi - \psi}$
$\frac{+ \phi \vee \psi}{+ \phi + \psi}$	$\frac{- \phi \vee \psi}{- \phi}$ $\frac{- \phi \vee \psi}{- \psi}$	$\frac{+ \phi \rightarrow \psi}{- \phi + \psi}$	$\frac{- \phi \rightarrow \psi}{+ \phi}$ $\frac{- \phi \rightarrow \psi}{ - \psi}$

Таблица 3 -- Правила вывода для связок Аллена

$\frac{+A b B}{B^- - A^+ \geq 1}$	$\frac{-A b B}{A^+ - B^- \geq 0}$
$\frac{+A m B}{A^+ = B^-}$	$\frac{-A m B}{B^- - A^+ \geq 1 A^- - B^+ \geq 1}$
$\frac{+A o B}{B^- - A^- \geq 1}$ $\frac{+A o B}{A^+ - B^- \geq 1}$ $\frac{+A o B}{B^+ - A^+ \geq 1}$	$\frac{-A o B}{A^- - B^- \geq 0 B^- - A^+ \geq 0 A^+ - B^+ \geq 0}$
$\frac{+A f B}{A^- - B^- \geq 1}$ $\frac{+A f B}{A^+ = B^+}$	$\frac{-A f B}{B^- - A^- \geq 0 B^+ - A^+ \geq 1 A^+ - B^+ \geq 1}$
$\frac{+A s B}{A^+ = B^-}$ $\frac{+A s B}{B^+ - A^+ \geq 1}$	$\frac{-A s B}{B^- - A^- \geq 1 A^- - B^- \geq 1 A^+ - B^+ \geq 0}$
$\frac{+A d B}{A^- - B^- \geq 1}$ $\frac{+A d B}{B^+ - A^+ \geq 1}$	$\frac{-A d B}{B^- - A^- \geq 0 A^+ - B^+ \geq 0}$
$\frac{+A e B}{A^- = B^-}$ $\frac{+A e B}{A^+ = B^+}$	$\frac{-A e B}{B^- - A^- \geq 1 A^- - B^- \geq 1 B^+ - A^+ \geq 1 A^+ - B^+ \geq 1}$
$\frac{+A \alpha(V; \lambda) B}{V}$ $\frac{+A \alpha(\lambda) B}{+A \alpha(\lambda) B}$	$\frac{-A \alpha(V; \lambda) B}{\sim V -A \alpha(\lambda) B}$
$\frac{\sim X - Y \geq r}{Y - X \geq 1 - r}$	$\frac{\sim X = Y}{X - Y \geq 1 Y - X \geq 1}$
V – атомарное ограничение, λ – произвольное ограничение, X и Y – концы интервалов, $\alpha \in \Omega$.	

также относим к $\mu LA+$, считая его сокращением предложения $\sim\phi \vee \psi$.

Дедукцию в логике $\mu LA+$ определим по методу аналитических таблиц с помощью правил вывода, которые показаны в Табл.1 и Табл.2. Правила из Табл.1 – это обычные правила вывода для пропозициональной логики, соответствующие семантике булевых связок. Правила Табл.3

соответствуют семантике связок Аллена. Рассмотрим, например, правило, показанное в Табл.2 в второй строке и втором столбце. Из Табл.1 видно, что $A m B$ эквивалентно равенству $A^+ = B^-$. Отрицание этого равенства эквивалентно дизъюнкции $A^+ < B^- \vee A^+ > B^-$, которую можно переписать как $B^- - A^+ > 0 \vee A^+ - B^- > 0$ или как $B^- - A^+ \geq 1 \vee A^+ - B^- \geq 0$. Эквивалентность последних двух дизъюнкций следует из принятого нами предположения о дискретности времени для рассматриваемых интерпретаций. (При обычном порядке в множестве натуральных числах неравенства $x > y$ и $x \geq y+1$ эквивалентны.) При применении указанного правила к формуле $A b B$ к каждой ветви дерева, проходящей через эту формулу, присоединяется «вилка» с неравенствами $B^- - A^+ \geq 1$ и $A^+ - B^- \geq 1$ на ее концах.

Рассмотрим пример вывода в логике μLA^+ .

Пример 2. На Рис.2 изображено дерево вывода, построенное для доказательства того, что из формул

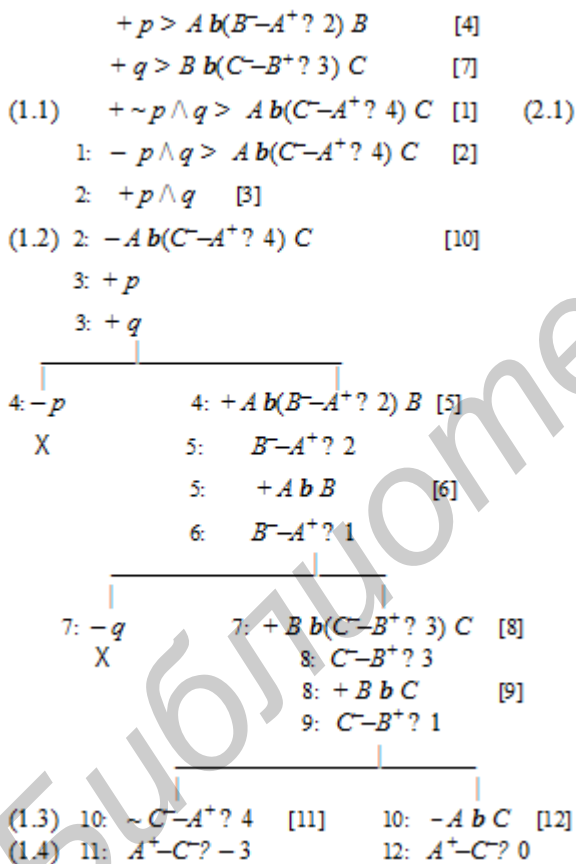


Рисунок 2 -- Дерево вывода в Примера 2

$p \rightarrow A b(B^- - A^+ \geq 2) B$ и $q \rightarrow B b(C^- - B^+ \geq 3) C$ логически следует формула $p \wedge q \rightarrow A b(C^- - A^+ \geq 4) C$. Это логическое следствие имеет место тогда и только тогда, когда невыполнимым является множество

$$\begin{aligned}
 O = \{ & p \rightarrow A b(B^- - A^+ \geq 2) B, q \rightarrow B b(C^- - B^+ \geq 3) C, \\
 & \sim p \wedge q \rightarrow A b(C^- - A^+ \geq 4) C \}.
 \end{aligned}$$

Построение дерева вывода начинается с ветви из

трех означенных формул $+p \rightarrow A b(B^- - A^+ \geq 2) B$, $+q \rightarrow B b(C^- - B^+ \geq 3) C$, $+\sim p \wedge q \rightarrow A b(C^- - A^+ \geq 4) C$, что означает предположение о выполнимости (непротиворечивости) множества O . Так как на самом деле это множество невыполнимо, то в результате должно получиться дерево, замкнутое в том смысле, что каждая ее ветвь замкнута, т.е. множество ее формул противоречиво (во всякой интерпретации ложна хотя бы одна из формул ветви).

На каждом шаге вывода к «нераскрытой» формуле (т.е. к которой не применялось правило вывода) может быть применено соответствующее правило вывода из Табл.1 и Табл.2. Например, к формуле $+p \rightarrow A b(B^- - A^+ \geq 2) B$ применимо правило, стоящее во второй строке и третьем столбце, которое обозначим T1.23. Результатом применения этого правила будет присоединение к начальной ветви вилки из двух формул $-p$ и $+A b(B^- - A^+ \geq 2) B$. Раскрытие формул $+q \rightarrow B b(C^- - B^+ \geq 3) C$ также дает вилку из формул $+q$ и $-B b(C^- - B^+ \geq 3) C$. же к формуле $+\sim p \wedge q \rightarrow A b(C^- - A^+ \geq 4) C$ применить T1.12 (стоящее в первой строке и втором столбце Табл.2), то к начальной ветви будет присоединена формула $-p \wedge q \rightarrow A b(C^- - A^+ \geq 4) C$.

На первом шаге вывода применяем к формуле $+\sim p \wedge q \rightarrow A b(C^- - A^+ \geq 4) C$ правило T1.12, что на Рис.2 указано помещением справа от этой формулы метки [1], обозначающей номер шага вывода. К полученной на шаге 1 формуле слева метка 1: , также обозначающая номер шага вывода формулы $-p \wedge q \rightarrow A b(C^- - A^+ \geq 4) C$. На шаге 2 к формуле $-p \wedge q \rightarrow A b(C^- - A^+ \geq 4) C$ применяем правило T1.24, в результате чего к текущей ветви (состоящей уже из четырех формул) будут последовательно приписаны две формулы $+p \wedge q$ и $-A b(C^- - A^+ \geq 4) C$. На шаге 3 к текущей ветви (состоящей из шести формул) приписываются две формулы $+p$ и $+q$. На шаге 4 к текущей ветви приписывается вилка из двух формул $-p$ и $+A b(B^- - A^+ \geq 2) B$. Таким образом, полученное после шаге 4 дерево содержит две ветви. Ясно, что левая ветвь дерева замкнута, так как она содержит контрарную пару формул $+p$ и $-p$. (Замкнутость ветви обозначена символом X.)

При построении дерева вывода мы каждый раз выбираем для раскрытия формулу, к которой применимо правило вывода с наименьшим ветвлением (т.е. с наименьшим числом альтернатив). Другими словами, дается следующим приоритет для выбора правил вывода:

(T1.11, T1.12, T1.13, T1.22, T1.24, T2.11, T2.12, T2.21, T2.31, T2.41, T2.31, T2.41, T2.51, T2.61, T2.71, T2.81, T2.91) раньше (T1.14, T1.21, T1.23, T2.14, T2.21, T2.26, T2.82, T2.92) раньше (T2.32, T2.42, T2.43) раньше T2.72.

Из нераскрытых формул одинакового приоритета выбирается самая верхняя и самая левая.

Результирующее дерево вывода содержит 4 ветви. Докажем, что дерево замкнуто. Выпишем из третьей ветви все неравенства: $B^-A^+ \geq 2$, $B^-A^+ \geq 1$, $C^-B^+ \geq 3$, $C^-B^+ \geq 1$, $A^+C^- \geq -3$. Ясно, что первое неравенство поглощает второе и третье поглощает четвертое; значит, второе и четвертое неравенства можно исключить. Возьмем еще неравенства $A^-A^+ \geq 1$, $B^-B^+ \geq 1$, $C^-C^+ \geq 1$ (которые справедливы, так как интервалы предполагаются невырожденными и время дискретно) и рассмотрим множество неравенств

$$S_1 = \{B^-A^+ \geq 2, C^-B^+ \geq 3, A^+C^- \geq -3, A^-A^+ \geq 1, B^-B^+ \geq 1, C^-C^+ \geq 1\}.$$

Множество неравенств S_1 противоречиво (несовместно). В самом деле, складывая эти неравенства, получаем

$$(B^-A^+) + (C^-B^+) + (A^+C^-) + (A^-A^+) + (B^-B^+) + (C^-C^+) \geq 2 + 3 + (-3) + 1 + 1 + 1.$$

Отсюда получаем противоречие $0 \geq 5$. Таким образом, третья ветвь замкнута. Аналогично, для четвертой ветви имеем множество неравенств

$$S_2 = \{B^-A^+ \geq 2, C^-B^+ \geq 3, A^+C^- \geq 0, A^-A^+ \geq 1, B^-B^+ \geq 1, C^-C^+ \geq 1\},$$

которое также противоречиво.

С системами неравенств S вида $X - Y \geq r$ (где X, Y – интервалы и r – натуральное число) свяжем граф $\Gamma(S)$, считая интервалы его вершинами и соединяя X с Y дугой с меткой r , если неравенство $X - Y \geq r$ принадлежит S . На Рис.3 показаны графы $\Gamma(S_1)$ и $\Gamma(S_2)$. (Здесь дуги без обозначенных меток по умолчанию имеют метку 1.) Мы видим, что эти графы имеют циклы положительной длины.

Легко понять, что (в общем случае) система неравенств S тогда и только тогда граф имеет цикл положительной длины. Поэтому эффективные алгоритмы для обнаружения у графа положительного цикла могут быть использованы для доказательства замкнутости ветвей дерева вывода.

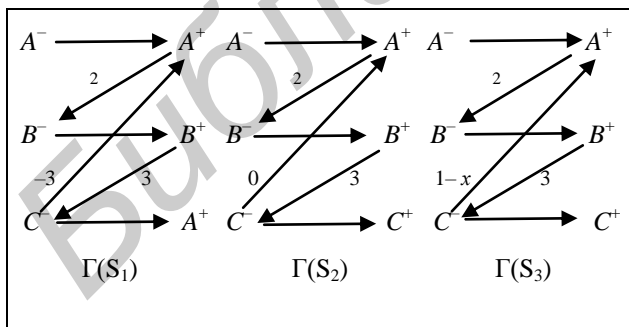


Рисунок 3 -- Графы систем неравенств

Метод аналитических таблиц можно применить для вычисления запросов к онтологиям, записанных в расширенной интервальной логике Аллена. Рассмотрим это на примере.

Пример 3. Возьмем онтологию

$$O = \{p \rightarrow A b(B^-A^+ \geq 2) B, q \rightarrow B b(C^-B^+ \geq 3) C\}$$

и запрос $? \max x - p \wedge q \rightarrow A b(C^-B^+ \geq x) C$ к O , который можно прочитать так: «Найти наибольшее x такое, что $O \models p \wedge q \rightarrow A b(C^-B^+ \geq x) C$ ».

Для получения ответа на этот запрос нужно построить дерево вывода для множества формул $O \cup \{\sim p \wedge q \rightarrow A b(C^-B^+ \geq x)\}$. Это дерево совпадает с показанным на Рис.3 за исключением того, что в формулах (1.1), (1.2) и (1.3) число 4 заменено на x , а в формуле (1.4) число -3 заменено на $1-x$.

Третья ветвь этого дерева вывода определяет следующее множество неравенств:

$$S_3 = \{C^-B^+ \geq 3, A^+C^- \geq 1-x, A^-A^+ \geq 1, B^-B^+ \geq 1, C^-C^+ \geq 1\}.$$

Граф $\Gamma(S_3)$ этого множества неравенств показан на Рис.3. Цикл A^+, B^-, B^+, C^-, C^+ в этом графе имеет длину $1 + 1 + 3 + (1-x)$, и эта длина положительна тогда и только тогда, когда $x \leq 6$. Следовательно, число 6 является наибольшим значением x и ответом на указанный запрос.

2. Спецификация потока работ в расширенной интервальной логике Аллена

В Примере 1 мы рассмотрели простейший пример потока работ для бизнес-процесса, целью которого является перевозка товара (см. схему этого потока работ на Рис.1). Спецификацией этого потока работ в расширенной интервальной логике Аллена дадим в форме следующей онтологии:

$$O = \{p \rightarrow A b(B^-A^+ = 2) B, \sim p \rightarrow A b(C^-A^+ = 3) C, A b(D^-A^+ \geq 1) D, B b(E^-B^+ \geq 22; B^+ - E^+ \geq 20) E, C b(E^-B^+ \geq 26; B^+ - E^+ \geq 22) E, E f d F\}.$$

Здесь мы предположили следующее: (1) обработка заказа выполняется 2 единицы времени; (2) оформление накладной выполняется 3 единицы времени; (3) перевозка товара фирмой F1 выполняется от 20 до 22 единиц времени; (4) перевозка товара фирмой F1 выполняется от 22 до 26 единиц времени;

Из неформального смысла данного потока работ очевидно, что схема потока работ непротиворечива. Но, чтобы формально это установить, нужно построить дерево вывода и убедиться, что оно имеет незамкнутые ветви. В данном случае в дереве вывода для онтологии O все ветви являются открытыми (незамкнутыми).

Рассмотрим пример запроса к онтологии O и вычисления ответа на запрос.

Пример 4. Сформулируем запрос так: «Оценить снизу время перевозки товара». Формально этот запрос записывается следующим образом:

$$? \max x - E^+ - A^- \geq x.$$

Ответ на этот запрос получается изложенным в предыдущем разделе способом (см. Пример 3). Для этого нужно построить дерево вывода для

множества формул $O \cup \{\sim E^+ - A^- \geq x\}$ и замкнуть ветвь, содержащую формулу $A^- - E^+ \geq 1 - x$, выбирая число x максимальным.

Закключение

Мы определили некоторое расширение известной интервальной логики Аллена. Предложениями расширенной логики являются булевы комбинации логики Аллена с включенной метрической информацией. Для расширенной логики был дан метод дедукции, основанный на аналитических таблицах. Показано, как этот метод дедукции можно применить к задачам спецификации и анализа потоков работ и как можно вычислять запросы, обращенные к онтологиям, специфицирующим потоки работ.

Работа выполнена при финансовой поддержки Российского Фонда Фундаментальных Исследований (проект 14-07-0387) и Министерства Образования и Науки Казахстана (проект 0115 РК 00532).

Библиографический список

- [Allen, 1987] Allen J.A. Maintaining knowledge about temporal intervals // Communications of the ACM, 20(11), 1983. – P.832-843.
- [van der Aalst et al, 2002] van der Aalst W., van Hee K. Workflow Management Models, Methods, and Systems. The MIT Press, Cambridge, Massachusetts London, 2002. – P. 53.
- [Gil Y. et al., 2007] Gil Y., Deelman E., Ellisman M., Fahringer T., Fox J., Gannon J., Goble C., Livny M., Moreau L., Myers J. Examining the challenges of scientific workflows // IEEE Computer, vol. 40, no. 12, 2007.
- [Lampert, 1991] Lampert L. The temporal logic of action. DEC Systems Research Center. Techn. Report 79, 1991. – P. 71.
- [Matschiner et al., 2005] Matschiner M., Saltzburger W. TANDEM: Integrating automated allele binning into genetics and genomic workflows // Bioinformatics 25 (15), 2005. – 1983-1985.
- [Wang et al., 2004] Wang Y., Fan Y. Using Temporal Logics for Modeling and Analysis of Workflows. Intern. Conf. on E-commerce for Dynamic E-Business. – IEEE Xplore, 2004.– P. 101-110.
- [WFMC, 1999] Workflow Management Coalition, Terminology & Glossary. Document Number WFMC-TC-1011. – <http://www.wfmc.org>.

SPECIFICATION OF WORKFLOWS IN THE EXTENDED INTERVAL ALLEN'S LOGIC

Plesniewicz G.S., Nguyen Thi Minh Vu

National Research University «MPEI», Moscow, Russia

salve777@mail.ru,

minhvu.357@gmail.com

It is shown, in the paper, that the extended interval Allen's logic can be used for specifying and analyzing workflows. The sentences of this logic are Boolean combination of Allen's logic sentences with, possibly, metric information.

Introduction

A workflow is a collection of cooperated, coordinated activities designed to carry out a complex process. Although the first and the main applications of workflows were in the field of business informatics, new their applications lies in organizing scientific investigations (scientific workflows, workflows for biological processes, genomic workflows et al.)

We propose to use the extended interval Allen's logic for specifying workflows.

Main Part

In the interval Allen's logic **LA**, there are 7 basic relations between temporal intervals: *b* (before), *m* (meets), *o* (overlaps), *f* (finishes), *s* (starts), *d* (during), *e* (equates). Let Ω be the set of these basic relations and their inversions. A **LA** sentence has the form $A \omega B$ where A, B are temporal intervals and ω is subset of Ω . The sentence $A \omega B$ is equivalent to disjunction of all $A \alpha B$ where $\alpha \in \Omega$.

The interval logic with metric information $\mu\mathbf{LA}$ has sentences that are obtained from **LA** sentences by incorporating metric information in the form of conjunctions inequalities between ends of intervals. For example, $A \ o(A^+ - A^- = 2, B^+ - B^- \leq 4) \ m(B^- - A^+ \geq 1) \ B$ is a $\mu\mathbf{LA}$ sentence.

The extended interval Allen's logic $\mu\mathbf{LA}+$ has sentences that are obtained from **LA** sentences by applying Boolean connectives.

It is shown how to apply the logic $\mu\mathbf{LA}+$ can be used for specifying workflows, analyzing workflows and query answering over workflows.

Conclusion

In this paper, we have shown how the extended interval Allen's logic can be used for specifying workflows. For this logic we define the deduction method based on analytical tableaux. This method is applied to the problems of analyzing workflows and query answering over workflows.



УДК 004.822:514

АЛГОРИТМ УНИФИКАЦИИ ВРЕМЕННЫХ РЯДОВ

Сапунков А.А., Афанасьева Т.В.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская федерация*

sapaslks@gmail.com

tv.afanasjeva@gmail.com

В работе приводится методика унификации временных рядов, то есть приведение временных рядов разной длины к временным рядам одинаковой длины для последующей обработки или анализа. Описывается наиболее популярный метод для решения данной задачи, метод F-преобразований, и приводится новая разработанная методика унификации временных рядов. В конце работы сравниваются обе методики, указываются преимущества и недостатки.

Ключевые слова: временной ряд, унификация временного ряда по длине, F-преобразование, обработка временного ряда.

Введение

Актуальность. Одной из важных проблем при анализе больших баз данных временных рядов (ВР) является обнаружение схожих по поведению ВР и их лингвистическое описание для упрощения последующего процесса моделирования. Данная проблема тесно связана с феноменами “Big Data” и “Internet of Thing”, при помощи которых предоставляются большие объемы данных с датчиков высокой частоты. Полученные ВР трудно оценить эксперту из-за их объема и количества в текстовом или графическом представлении. А для автоматизированных систем анализа, сравнения и оценки ВР необходимо приводить все ВР к единому виду. Специализированных методов для унификации ВР, т.е. для приведения ВР к общей длине и общему диапазону значений, на момент написания данной работы обнаружено не было. Обычно для выполнения данной задачи использовался метод F-преобразования [Перфильева 2003], поэтому с ним и будет сравниваться приведенный в данной статье алгоритм

Цель. Главной целью данной работы было разработать методику приведения ВР к единому формату для последующей обработки (анализа, сравнения, прогнозирования) ВР в автоматизированных системах. Под единым форматом подразумевается единый для всех ВР диапазон значений ВР и его длина.

Задача. Разработать алгоритм унификации ВР, т.е. алгоритм приведения ВР разных длин к ВР указанной длины путем сжатия или расширения ВР.

В первом разделе описывается применение алгоритма F-преобразований для сжатия или расширения ВР. В следующем разделе определяется новый метод унификации ВР, приводится пошаговый алгоритм преобразования заданного ВР в унифицированный и приводятся примеры работы алгоритма. В третьем разделе приводятся результаты обработки базы временных рядов при помощи F-преобразования и нового предложенного алгоритма, а также оценка эффективности нового алгоритма.

1. Описание F-преобразования

Методика F-преобразования успешно применяется для анализа ВР на основе моделей мягкого вычисления, к примеру [Novak 2010], [Perfilieva 2010]. Цель F-преобразования – предобработка ВР. Ниже приведены важные для F-преобразования свойства:

- удаление высокочастотных колебаний,
- сглаживание и разложение.

Далее мы коротко напомним основные положения F-преобразования, которые были представлены в [Перфильева 2003]. Данный метод уже был описан в ряде работ, поэтому детали опускаются.

Предположим, задан ВР X из n значений, где $x_t \in X, t=1,2,\dots,n$.

F-преобразование проходит в 2 этапа:

- Прямое F-преобразование.
- Обратное F-преобразование.

На первом этапе F-преобразования происходит разбиение заданного ВР X на $[1, n]$ отрезков с помощью m : $2 \leq m \leq n - 2$ нечетких множеств A_1, \dots, A_m : $[1, n] \rightarrow [0, 1]$, которые называются прямыми F-преобразованиями. Значение m получается путем деления длины ВР n на коэффициент преобразования c , который указывается при преобразовании, и округляется до целого. Далее компоненты вектора $Fm[X] = [F_1, \dots, F_m]$ рассчитываются следующим образом:

$$F_k = \frac{\sum_{t=1}^n x_t * A_k(t)}{\sum_{t=1}^n A_k(t)} \quad (1.1)$$

где $k = 1, 2, \dots, m$; A_1, \dots, A_m – это треугольные функции, определенные на отрезке $[1, n]$. согласно [19], для F-преобразования ВР необходимо определить значение m равноудаленных точек основной функции и рассчитать компоненты вектора $Fm[X] = [F_1, \dots, F_m]$ при помощи взвешенных средних.

Второй этап F-преобразования – обратный этап. На этом этапе аппроксимированное представление некой функции, определенной поведением временного ряда X предстает в новом виде:

$$f_t = \sum_{k=1}^m F_k * A_k(t) \quad (1.2)$$

Таким образом F-преобразование можно использовать для решения поставленной задачи: прямое F-преобразование для сжатия ВР, обратное – для расширения.

2. Приведение ВР к унифицированному виду

В данной методике приводится понятие унифицированного ВР. Предположим, задан ВР X из n значений, где $x_t, x_t \in X, t=1, 2, \dots, n$, тогда унифицированным ВР будем считать ВР Z из m значений где $z_p, z_p \in Z, z_p \in [0..K] p=1, 2, \dots, m$. Таким образом получаем, что привести ВР к унифицированному виду значит преобразовать все значения заданного ВР в диапазон от 0 до K и расширить/сократить сам ВР до m значений.

Получим ВР Y из n значений, где $y_t, y_t \in Y, y_t \in [0..K] t=1, 2, \dots, n$. Для этого преобразуем все значения ВР X по формуле (2.1):

$$y_t = \frac{x_t - x_{\min}}{x_{\max} - x_{\min}} * K, \quad (2.1)$$

где x_{\min} – минимальное значения ВР X , x_{\max} – максимальное значение ВР X .

Далее возможно две ситуации при приведении ВР к унифицированному ВР. Первая – это при n и m не равны, тогда необходимо провести сжатие или расширение ВР Y . Вторая ситуация при $n=m$, в таком случае считаем, что ВР Y приведен к унифицированному виду.

Рассмотрим два ВР Y и Z различных длин. Для расчета значений унифицированного ВР z_p необходимо определить между какими значениями ВР Y y_t и y_{t+1} оно находится, поэтому необходимо ввести соответствие между координатами ВР t и p :

$$t = \frac{p * n}{m} \text{ DIV } 1, \quad (2.2)$$

где $a \text{ DIV } b$ – операция целочисленного деления числа a на b .

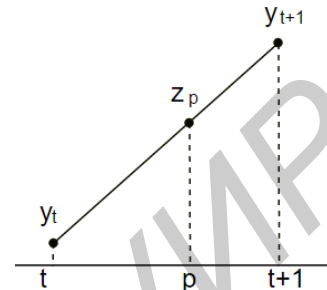


Рисунок 1 – определение значения z_p

В данном случае (2.2) применяется для получения целой части значения a . Далее определяем значение z_p , опираясь на подобие треугольников с общей гипотенузой (y_t, y_{t+1}) и противолежащим углом в 90 градусов (рис. 1). Так как точки p и t относятся к разным координатам, то нам необходимо найти p' – положение точки p в координатах ВР Y . Этом можно сделать по следующей формуле:

$$p' = \frac{p * n}{m}. \quad (2.3)$$

Таким образом взяв дробную часть от p' мы получим расстояние от точки t до p' или же коэффициент отношения треугольников. Теперь мы можем получить значения z_p :

$$z_p = (y_{t+1} - y_t) * (p' \text{ MOD } 1) + y_t \quad (2.4)$$

Где $a \text{ MOD } b$ – операция получения остатка деления числа a на b . Таким образом по формуле (2.4) можно получить ВР Z вне зависимости от того, сжимаем мы ВР или расширяем.

3. Пример обработки ВР методикой F-преобразований и алгоритмом унификации ВР

Рассмотрим ВР ts11 (ВР изображен на рисунке 2) длиной в 909 значений из базы данных, предложенной в рамках конкурса International Time Series Forecasting Competition [ITSFC]. В рамках эксперимента будем сживать данный ВР до 100 значений и расширять до 2000 методом F-преобразований и алгоритмом унификации ВР, чтобы можно было их сравнить.

Для того чтобы сократить ВР ts11 при помощи прямого F-преобразования с коэффициентом сжатия $c=9$. В результате преобразования получаем ВР из 101 значения, поэтому для получения ВР нужной нам длины мы отсекаем лишнее последнее значение

(результат изображен на рисунке 3). Для расширения ВР необходимо применить обратное F-преобразование. Однако мы сталкиваемся с проблемой выбора коэффициента, если $c=2$, то ВР получится длиной 1818 (результат изображен на рисунке 4) т.е. нам будет не хватать 182 значений, если $c=3$, то получится ВР длиной 2727 значений, а отсекать 727 значений, которые составляют больше четверти временного ряда нельзя.

Далее преобразуем заданный ВР ts11 при помощи алгоритма унификации с параметрами $k=50$, $m=100$. В результате мы получаем, как и при применении F-преобразования, сглаженный ВР без всесоставных колебаний, однако его длина составляет 100 значений, поэтому нет необходимости отсекать какие-либо значения. Результат отображен на рисунке 5.

При преобразовании ВР ts11 при помощи алгоритма унификации с параметрами $k=50$, $m=2000$ у нас получился ВР длиной в 2000 значений изображенный на рисунке 6. Данный ВР получился не сглаженный и сохранил все высокочастотные колебания.

Выводы. При использовании метода F-преобразований для сжатия и расширения следует учитывать длину полученного ВР и требуемую длину, чтобы количество отсекаемых значений не влияло на основную тенденцию ВР.

Использование предложенного алгоритма не имеет недостатка с отсечением значений, подобно методу F-преобразований. При расширении ВР данный алгоритм может иметь небольшую погрешность, которая не повлияет ни на общую тенденцию ВР ни на высокочастотные шумы.

Если длина требуемого временного ряда делится нацело на длину заданного временного ряда, то результаты расширения ВР по обоим методикам будут идентичны.

При сжатии ВР результаты методик сильно отличаются друг от друга из-за того, что метод F-преобразований предназначен для сглаживания ВР, а предложенный метод берет соответствующие значения ВР без сглаживания.

4. Заключение

Выводы. При использовании метода F-преобразований для сжатия и расширения следует учитывать длину полученного ВР и требуемую длину, чтобы количество отсекаемых значений не влияло на основную тенденцию ВР.

Использование предложенного алгоритма не имеет недостатка с отсечением значений, подобно методу F-преобразований. При расширении ВР данный алгоритм может иметь небольшую погрешность, которая не повлияет ни на общую тенденцию ВР ни на высокочастотные шумы.

Если длина требуемого временного ряда делится нацело на длину заданного временного ряда, то

результаты расширения ВР по обоим методикам будут идентичны.

При сжатии ВР результаты методик сильно отличаются друг от друга из-за того, что метод F-преобразований предназначен для сглаживания ВР, а предложенный метод берет соответствующие значения ВР без сглаживания.

По результатам эксперимента можно сделать вывод, что новый алгоритм унификации лучше выполняет поставленную задачу, по сравнению с методом F-преобразований. Главное его преимущество заключается в том, что он может преобразовывать заданный ВР во ВР любой указанной длины.

Библиографический список

[Перфильева 2003] Перфильева И. Нечеткое преобразование: применительно к проблеме роста рифов. В кн: Демикко Р. Клир GJ и др. редакторы. Нечеткая логика в геологии. Амстердам: Академическая пресса: 2003, 275-300.

[Novak 2010] Novak, V., Stepnicka, M. Dvorak, A., Perfilieva, I. and Pavliska, V. Analysis of seasonal time series using fuzzy approach, International Journal of General Systems, 2010 39, pp.305-328.

[Perfilieva 2010] Perfilieva I., Yarushkina N., Afanaseva T. Relaxed Discrete F-Transform and its Application to the Time Series Analysis // Da Ruanetal (Eds.): Computational Intelligence. Foundations and Applications (Proc. of the 9th Int. FLINS Conf.). P. 249 -255, World Scientific, Emei, Chengdu, China, 2-4 August, 2010.

[ITSFC] International Time Series Forecasting Competition [Электронный ресурс]. URL: <http://irafim.osu.cz/cif/main.php> (дата обращения: 16.03.2015).

TIME SERIES REDUCTION ALGORITHM TO THE SPECIFIED FORMAT

Sapunkov A.A. Afanasjeva T.V.

Ulyanovsk State Technical University

sapaslks@gmail.com

tv.afanasjeva@gmail.com

The paper describes a method of unification of the time series, in other words, bringing the different length of time series to a time series of the same length for later processing or analysis. It describes the most popular method for solving this problem, the method of F-transformation, and describes the new developed method. At the end of the two methods are compared, indicate advantages and disadvantages.

Key words: time series, the unification of the length of time series, F-transform, processing time series.

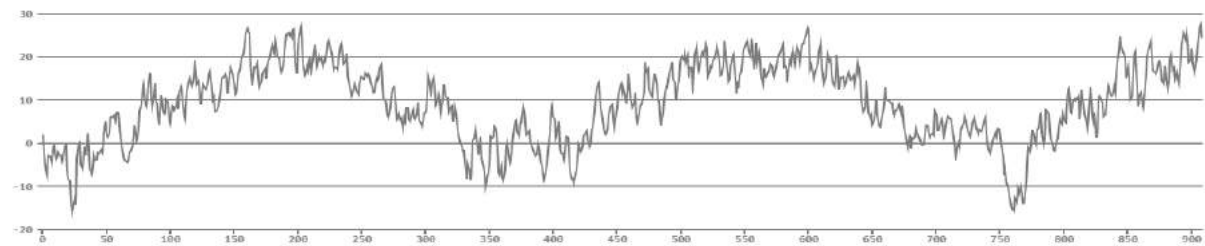


Рисунок 2 – ВР ts11

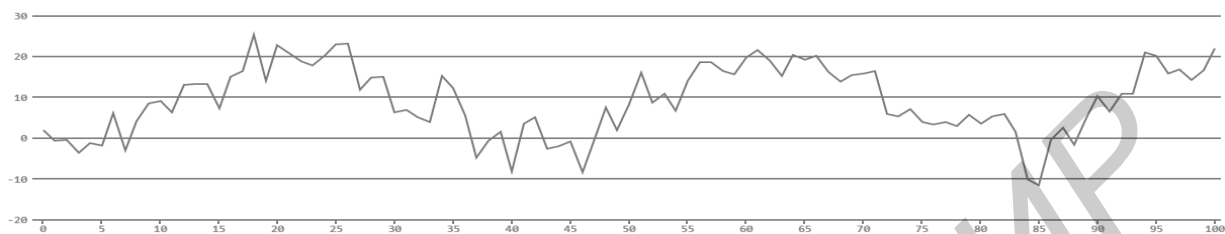


Рисунок 3 – ВР ts11 сжатый методом F-преобразований

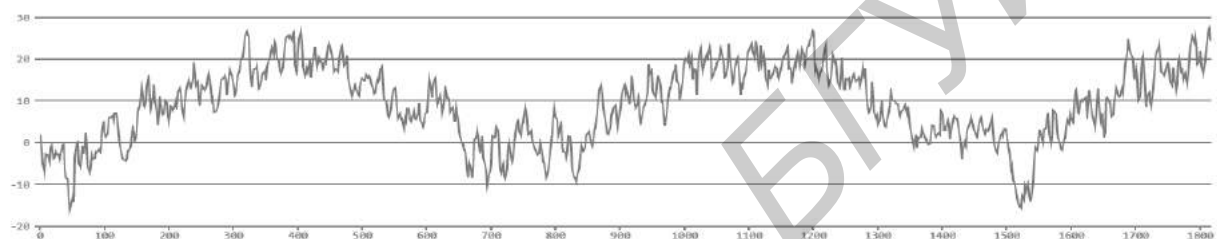


Рисунок 4 – ВР ts11 расширенный методом F-преобразований

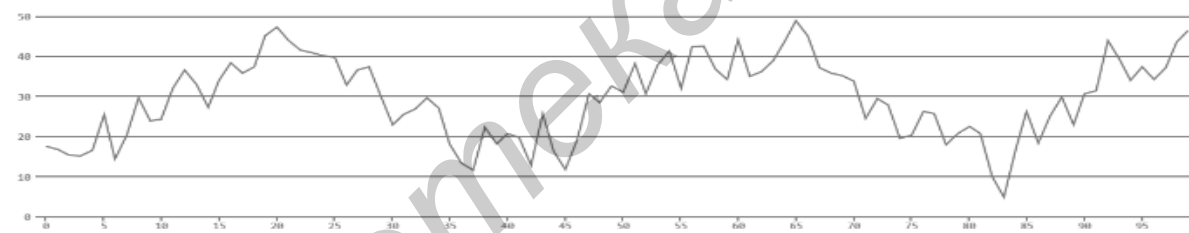


Рисунок 5 – ВР ts11 сжатый алгоритмом унификации ВР

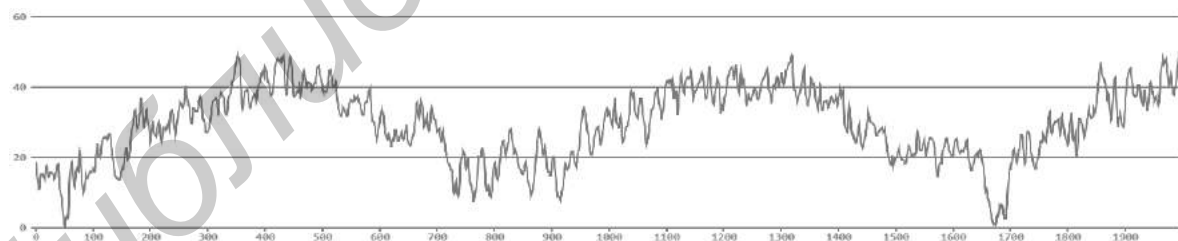


Рисунок 6 – ВР ts11 расширенный алгоритмом унификации ВР



УДК 004.89

ОПИСАНИЕ ПОДХОДА К ИЗВЛЕЧЕНИЮ ЗНАНИЙ ОБ АНОМАЛИЯХ ЛОКАЛЬНЫХ ТЕНДЕНЦИЙ ВРЕМЕННОГО РЯДА

Афанасьева Т.В., Заварзин Д.В.

Ульяновский государственный технический университет,

г. Ульяновск, Россия

tv.afanasjeva@gmail.com

d.zavarzin91@gmail.com

В работе предлагается новый подход для извлечения знаний об аномалиях в данных которые представлены в виде локальных нечетких тенденций временных рядов (ВР). Работа основана на определениях гранулярного представления ВР, а также классификациях аномалий и свойств поведения нечетких тенденций ВР описанных в предыдущих работах.

Ключевые слова: извлечение знаний, поиск аномалий, временной ряд, нечеткая тенденция.

Введение

Целью процесса извлечения знаний из любого источника данных – является выявление в них нетривиальных зависимостей и закономерностей. Это необходимо для того, чтобы получить знания, пригодные для интерпретации человеком экспертом и для помощи в принятии решений человеку, не обладающими квалификацией эксперта в широком диапазоне сфер человеческой деятельности. Например, методы извлечения знаний типа могут применяться в следующих отраслях деятельности: торговле (анализ структуры клиентов и покупок, моделей поведения потребителей), в сферах защиты информации банков и страховых организаций (выявление случаев мошенничества, создание новых целевых программ и услуг), медицинских учреждений (выработка дорожной карты лечения, прогнозирование состояния здоровья).

Рассматривая примеры, приведенные выше, нетрудно заметить, что одной из наиболее востребованных задач в этих сферах является поиск отклонения от нормального поведения процесса, знания из данных которого мы намерены извлекать. Поведение процесса не свойственное картине нормального (типичного) протекания процесса характеризуется понятием – аномалия. Процедуру извлечения знаний о поведении анализируемого процесса необходимо автоматизировать и это связано в первую очередь, со сложностью систем мониторинга, сбора и накопления гетерогенных данных с применением традиционных средств анализа (визуальное наблюдение, экспертное

оценивание). Каждый экземпляр данных в этом случае может быть зависим от множества других, а также внешних воздействий.

Процесс извлечения знаний всегда сопровождается работой с данными внушительных объемов, распределенных по временным интервалам разной размерности – т.е. темпоральными данными. В современных аналитических системах данные представляются в виде временных рядов, как наиболее удобного средства представления темпоральных данных. Такая форма представления является удобной для решения задач экспертного и традиционного методов анализа процессов в настоящее время.

Анализом временных рядов, а также задачей извлечения знаний об аномалиях в наборах данных занимается специальное научное направление – теория временных рядов. В частности, извлечение знаний об аномалиях является важным для решения задач прогнозирования данных – идентификация и устранение аномалий, например, способствует повышению точности и глубины прогноза [ссылка]. В настоящей работе предлагается рассмотреть задачу извлечения знаний об аномалиях с точки зрения частного метода анализа временных рядов – метода нечетких тенденций.

В последнее время популярность получило относительно новое направление теории анализа временных рядов – теория нечетких временных рядов. Она рассматривает временные ряды не с точки зрения привычных числовых моделей, а опираясь на математический аппарат нечеткой логики [Афанасьева и др., 2009]. Наибольший

интерес в этой области представляет извлечение знаний о наличии редко встречающихся локальных тенденций ВР. Однако, задача представления знаний об аномалиях в локальных тенденциях не получила достаточного рассмотрения в исследованиях отечественных и зарубежных авторов.

В настоящей работе будет предложен новый подход к извлечению знаний об аномалиях в локальных тенденциях на основе их нечеткого представления и особенностей поведения ВР. В первой главе будет дано понятие временного ряда, нечеткой тенденции и аномалии, рассмотрены существующие методы извлечения знания об аномалиях в локальных тенденциях ВР, а также какие недостатки они содержат. Во второй главе будет описан новый подход к извлечению знаний об аномалиях локальных тенденций ВР на основе анализа поведения ВР.

1. Существующие методы и подходы к извлечению знаний об аномалиях локальных тенденций

Временной ряд – это последовательность дискретных упорядоченных в неслучайные равноотстоящие моменты времени измерений (показателей наблюдений) $X = \{x_t\}$, где $t = \overline{1, n}$, характеризующих уровни состояния изучаемого процесса. Практически любые наблюдения, полученные в результате мониторинга процессов в различных сферах человеческой деятельности (безопасность, медицина, экология и т.д.), могут быть представлены в виде некоторой последовательности, зависимой от времени – то есть в форме ВР. В теории нечетких систем временной ряд представляется в форме нечеткого временного ряда (НВР), где в равноотстоящие дискретные временные отсчеты $\tilde{X} = \{\tilde{x}_t\}$, где $t = \overline{1, n}$, фиксируются последовательности нечетких меток. Для НВР было формализовано понятие нечеткой тенденции, выражающей качественное изменение на временном интервале НВР.

Нечеткая тенденция (НТ) ВР $\tau \in \mathfrak{Z}$ НВР $\tilde{X} = \{\tilde{x}_t\}$, $t = \overline{1, n}$, может быть представлена структурно-лингвистической моделью в виде отношения, построенного на декартовом произведении свойств нечеткой тенденции $\tilde{V} \times \tilde{A} \times \Delta \tilde{T} \rightarrow \mathfrak{Z}$ [Афанасьева и др., 2011]:

$$\tau = \langle \tilde{v}, \tilde{a}, \Delta \tilde{t}, \mu \rangle \quad (1)$$

где \tilde{v} – функция принадлежности нечеткого множества типа НТ $\tilde{v} \in \tilde{V}$, выражающая направление изменения (например, $\tilde{v}_1 =$ «рост», $\tilde{v}_2 =$ «стабильность», $\tilde{v}_3 =$ «падение» и т.д.); \tilde{a} – функция принадлежности нечеткого множества интенсивности НТ, $\tilde{a} \in \tilde{A}$, выраженное лингвистическим термом (например, $\tilde{a}_1 =$

«большое», $\tilde{a}_2 =$ «среднее», $\tilde{a}_3 =$ «малое» и т.д.); $\Delta \tilde{t}$ – длительность НТ, где $\Delta \tilde{t} \in \Delta \tilde{T}$; μ – функция принадлежности участка НВР, ограниченного интервалом $\Delta \tilde{t}$ НТ.

Под аномалиями объектов ВР мы можем понимать следующие: четкие и нечеткие значения, локальные, элементарные и основные тенденции, а также те типы, которые могут быть найдены в зависимостях объектов ВР одного типа (правила следования) или разных типов (комбинированные) (рис. 1).

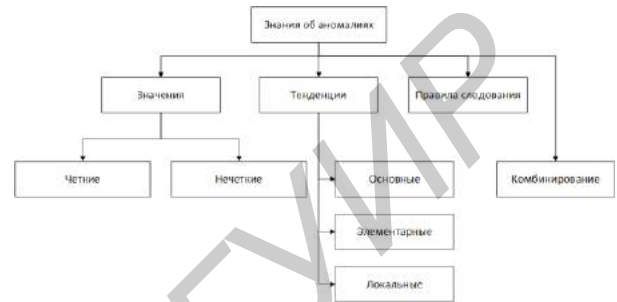


Рисунок 1 – Схема типологии знаний об аномалиях ВР

Опираясь на формулу (1) для представления знаний об аномалиях в локальных тенденциях в работе [Афанасьева и др., 2014] использовалась следующая классификация темпоральных процессов, которые могут быть описаны при помощи локальных нечетких тенденций ВР:

- процессы I рода – класс процессов, содержащих аномалии;
- процессы II рода – класс процессов, не содержащих аномалии.

Пусть $\mathfrak{Z} = \{\tilde{\tau}_t\}$, где $t = \overline{1, m}$ – дискретный ВР, состоящий из m – локальных нечетких тенденций. Тогда, согласно формуле (1) условие обнаружения аномального поведения нечеткого ВР по компонентам нечетких локальных тенденций может быть выражены следующим образом:

$$\tilde{v}_t = \begin{cases} \tilde{v}_t^{valid}, \tilde{v}_t \notin \tilde{V}' \\ \tilde{v}_t^{invalid}, \tilde{v}_t \in \tilde{V}' \end{cases} \quad (2)$$

$$\tilde{a}_t = \begin{cases} \tilde{a}_t^{valid}, \tilde{a}_t \notin \tilde{A}' \\ \tilde{a}_t^{invalid}, \tilde{a}_t \in \tilde{A}' \end{cases} \quad (3)$$

$$\Delta \tilde{t}_t = \begin{cases} \Delta \tilde{t}_t^{valid}, \Delta \tilde{t}_t \notin \Delta \tilde{T}' \\ \Delta \tilde{t}_t^{invalid}, \Delta \tilde{t}_t \in \Delta \tilde{T}' \end{cases} \quad (4)$$

где \tilde{v}_t^{valid} – допустимое значение типа НТ \tilde{v}_t , $\tilde{v}_t^{invalid}$ – недопустимое значение типа НТ \tilde{v}_t , $\tilde{V}' = \{\tilde{v}'_1, \tilde{v}'_2, \dots, \tilde{v}'_n\}$ – множество аномальных типов НТ на ВР X;

\tilde{a}_t^{valid} – допустимое значение интенсивности НТ

$\tilde{a}_i, \tilde{a}_i^{invalid}$ – недопустимое значение интенсивности

НТ $\tilde{a}_i, \tilde{A}' = \{\tilde{a}'_1, \tilde{a}'_2, \dots, \tilde{a}'_n\}$ – множество аномальных интенсивностей НТ на ВР X;

$\Delta\tilde{t}_i^{valid}$ – допустимое значение продолжительности НТ $\Delta\tilde{t}_i, \Delta\tilde{t}_i^{invalid}$ – недопустимое значение продолжительности НТ $\Delta\tilde{t}_i, \Delta\tilde{T}' = \{\Delta\tilde{t}'_1, \Delta\tilde{t}'_2, \dots, \Delta\tilde{t}'_n\}$ – множество аномальных значений продолжительности НТ на ВР X.

В работе [Заварзин и др., 2015] был приведен пример классификации знаний об аномалиях с точки зрения анализа поведения ВР.

2. Описание алгоритма извлечения знаний об аномалии локальной тенденции ВР

В данной работе предлагается подход к извлечению знаний об аномалиях анализируя поведение локальных тенденции ВР. В отличие от других работ (например, [Заварзин и др., 2015]) в этой статье описан подход для поиска аномалий в условиях, когда множество возможных аномалий неизвестно. Идея нового подхода заключается в сравнении поведения анализируемой НЛТ относительно поведения основной НТ выраженной через свойство типа тенденции. Локальная тенденция тип которой противоречит поведению основной тенденции ВР – возможно является аномальной.

Для достижения своей цели подход использует ACL – шкалу для интерпретации локальной тенденции и её составляющих – типа, интенсивности и длительности в виде лингвистических термов. Более подробно метод разбиения при помощи ACL-шкалы описан в работе [Афанасьева, 2008].

Далее осуществляется анализ НЛТ относительно основной тенденции ВР (табл. 1).

Таблица 1 – Пример извлечения знаний об аномалиях из локальных тенденций ВР

Поведение ВР, представленное в виде типа основной тенденции		Пример	Возможный тип аномальной локальной тенденции
Тип = «стабильность»		Рис. 2	рост, падение
Тип = «рост»	ступенчатый	Рис. 3	падение
	регулярный	Рис. 4	падение
Тип «падение»	ступенчатое	Рис. 6	рост
	регулярное	Рис. 7	рост

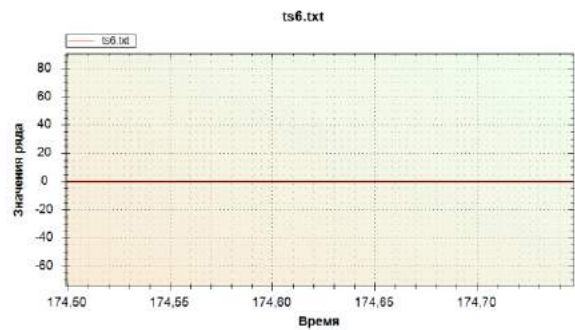


Рисунок 2 – График временного ряда, с типом основной тенденции – стабильность

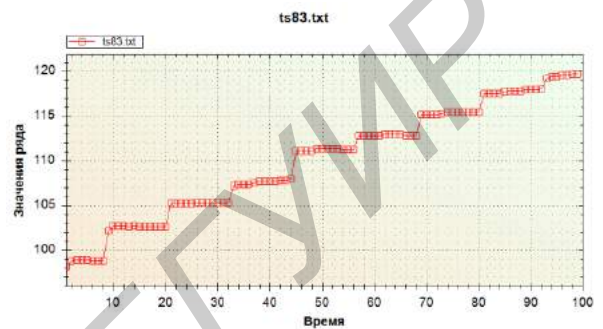


Рисунок 3 – График временного ряда, с типом основной тенденции – ступенчатый рост

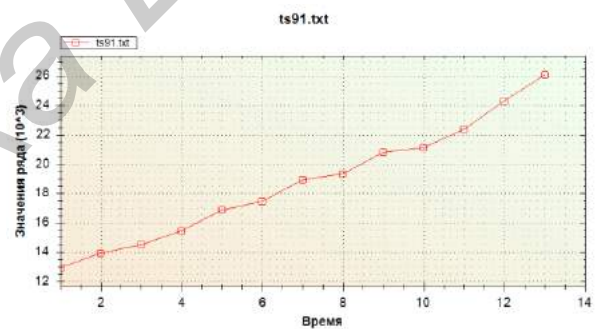


Рисунок 4 – График временного ряда, с типом основной тенденции – регулярный рост

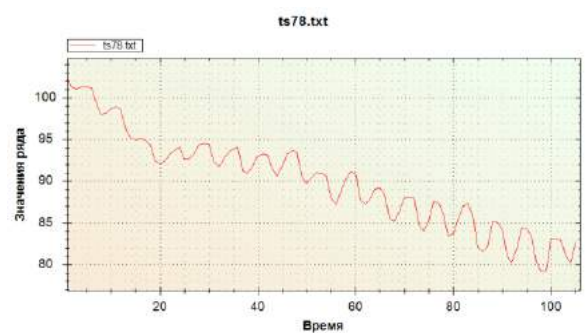


Рисунок 5 – График временного ряда, с типом основной тенденции – ступенчатое падение

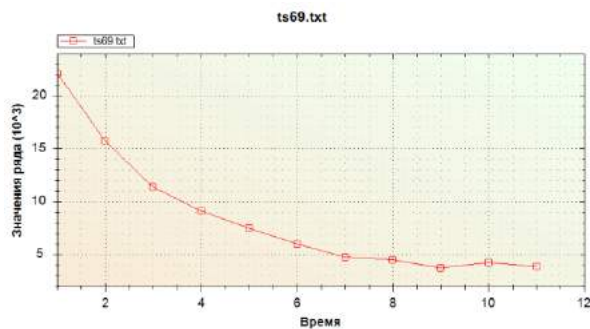


Рисунок 6 – График временного ряда, с типом основной тенденции – регулярное падение

Как видно в таблице 1, мы вполне однозначно можем утверждать об аномальности типа тенденции ВР, если она противоречит целевому типу основной тенденции. Однако, в случае с преобладающими типами «роста» и «падения» мы не можем дать утвердительной ответ на аномальность анализируемой локальной тенденции ВР. Одним из вариантов решения этой проблемы может быть более глубокая детализация целевой ЛТ, на более узкие интервалы ЛТ или элементарные тенденции.

Представление знания о локальной нечеткой тенденции может быть выражено следующим образом – отношения, задаваемые между характеристикой поведения локальными тенденциями и основной тенденцией ВР можно формализовать при помощи модели нечеткого логического вывода в виде множества нечетких правил «Если-То»:

Правило 1. Если ОТ = «стабильность» И НЛТ = «падение» ИЛИ НЛТ = «рост», ТО АЛТ = «истина»

Правило 2. Если ОТ = «ступенчатый рост» И НЛТ = «падение», ТО АЛТ = «истина».

Правило 3. Если ОТ = «регулярный рост» И НЛТ = «падение», ТО АЛТ = «истина».

Правило 4. Если ОТ = «ступенчатое падение» И НЛТ = «рост», ТО АЛТ = «истина».

Правило 5. Если ОТ = «регулярное падение» И НЛТ = «рост», ТО АЛТ = «истина».

где АЛТ – аномалия локальной тенденции ВР, ОТ – основная тенденция ВР, НЛТ – локальная нечеткая тенденция.

Используя базу нечетких правил, можно извлечь знание об аномальности анализируемого ВР.

Заключение

В настоящей работе предложен новый подход для извлечения знаний об аномалиях нечетких локальных тенденций ВР. Данный подход предполагает акцентирование внимания на анализе текущего поведения ВР, а не на сравнении с существующей моделью нормального поведения.

Данный подход, однако, в силу отсутствия экспериментального подтверждения и работы с объектами – локальными тенденциями, которые не раскрывают закономерностей поведения ВР в

полной мере может быть расширен и дополнен с учетом полной классификации, представленной в работе [Заварзин и др., 2015]. Кроме того, данный подход может быть применен при решении задачи определения адекватности прогноза исходного ВР, по анализу его локальной тенденции.

Таким образом следующий этап развития этого подхода заключается в его алгоритмизации, реализации и проведении эксперимента, идентификация трудоемкости его исполнения и быстрей действия.

Библиографический список

[Афанасьева и др., 2009] Афанасьева Т.В., Ярушкина Н.Г. Нечеткое моделирование временных рядов и анализ нечетких тенденций. – Ульяновск: УлГТУ, 2009. – 299 с.

[Афанасьева и др., 2011] Афанасьева Т.В., Ярушкина Н. Г. Нечеткий динамический процесс с нечеткими тенденциями в анализе временных рядов // Вестник Ростовского государственного университета путей сообщения, -2011, -№3. -С. 7-16.

[Афанасьева и др., 2014] Афанасьева Т.В., Заварзин Д.В. Модель локальных тенденций в задачах анализа аномалий для BIG DATA // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. А. Н. Афанасьева. - Ульяновск: УлГТУ, 2014. - 228 с.

[Заварзин и др., 2015] Заварзин Д.В., Курлов А.П. Классификация аномалий временного ряда на основе анализа его тенденций // Интеллектуальные системы и технологии: современное состояние и перспективы. Сборник научных трудов III-ей Международной летней школы-семинара по искусственному интеллекту для студентов, аспирантов и молодых ученых (Тверь – Протасово, 1-5 июля 2015 г.) – Тверь: Изд-во Тверского государственного технического университета, 2015. – 234 с.

[Афанасьева, 2008] Афанасьева Т.В. Модель ACL-шкалы для генерации лингвистических оценок в принятии решений // Вопросы современной науки и практики. – Изд. Университет им. В.И. Вернадского. – №4(14). – 2008. – т. 2

DESCRIPTION APPROACH TO EXTRACTING KNOWLEDGE ABOUT GLOBAL TRENDS ANOMALIES TIME SERIES

Afansieva T.V., Zavarzin D.V.

Ulyanovsk State Technical University,
Ulyanovsk, Russia Federation

tv.afanasjeva@gmail.com

d.zavarzin91@gmail.com

The paper proposes a new approach for extracting knowledge of the anomalies in the data presented in the form of local fuzzy time-series trends. The work is based on the definitions of granular representation of time series, as well as classifications of anomalies and behavior properties of fuzzy BP trends described in previous papers.



УДК 004.81

КОГНИТИВНЫЙ ФРАКТАЛЬНЫЙ АНАЛИЗ ВРЕМЕННЫХ РЯДОВ: R/S АНАЛИЗ И V-СТАТИСТИКА

Романко О.Р.

*Национальный технический университет Украины «Киевский политехнический институт»
г. Киев, Украина*

romanko1993@gmail.com

Фрактальный анализ временных рядов – новая методология, использующая инструментарий фрактальной математики для анализа и прогнозирования финансовых и экономических процессов. Представлены описания и алгоритмы нового подхода к анализу временных рядов, позволяющие привнести некий семантический смысл в исследуемые данные - методология R/S анализа, позволяющая идентифицировать процессы с длинной и короткой памятью, и V-статистика, для нахождения длины неперiodических циклов.

Ключевые слова: когнитивный фрактальный анализ; R/S анализ; V-статистика.

Введение

Современный подход к ведению бизнеса в любой сфере человеческой деятельности состоит, прежде всего, в возможности предугадать или предсказать ближайшее развитие событий. Для этого, в первую очередь, нужно «понять» данные, провести так называемый технический и фундаментальный анализ процесса. Последний подразумевает под собой соединение всех известных знаний о нем в единую взаимосвязанную сеть, соединенную причинно-следственными связями. Большинство стараний исследователя направлено на получение более семантически значимого представления о процессе или явлении.

В данной работе изложены следующие модели и инструменты анализа, позволяющие, по мнению автора, получить достаточно значимые знания относительно исследуемых процессов, являющиеся определенным маячком при дальнейшем их исследовании:

- Метод R/S – анализа, который позволяет исследователю идентифицировать процесс с длинной памятью и классифицировать по его особенностям (персистентный или антиперсистентный).
- V-статистика, позволяющая исследовать среднюю длину неперiodических циклов, возникающих в процессах вследствие сезонности определенных явлений.
- Альтернативный метод оценки памяти – с помощью модели дробного броуновского движение

Данная методология позволяет дать количественную оценку такого явления, как память процесса. Этой оценкой является параметр Хёрста.

Параметр Хёрста $H \in (0,1)$ – показывает характер изменения процесса. При $H \in (0.5,1]$ процесс является персистентным, он сохраняет имеющуюся тенденцию, то есть возрастание в прошлом более вероятно приводит к возрастанию в дальнейшем, и наоборот. Биржевые маклеры описывают такое поведение возникновением «трендов» и «пробоев». При $H \in [0,0.5)$ процесс называется антиперсистентным, любая тенденция стремится смениться противоположной, формируется так называемый «коридор», в котором процесс осуществляет колебания. При $H = 0.5$ процесс является полностью случайным, явной тенденции не выражено. Стоит отметить, что природные процессы имеют параметр Хёрста $H = 0.72 - 0.73$

1.Предпосылки возникновения нового фрактального подхода к анализу временных рядов

Появление нового фрактального подхода, связано с обнаружением некоторых ошибочных представлений о природе и поведении многих финансовых процессов.

Во-первых, это широко распространенное применение основной теоремы статистики – центральной предельной теоремы. Эта теорема гласит о том, что сумма большого числа независимых одинаково распределенных величин

имеет нормальное (гауссово) распределение. Но почти всегда временные показатели финансового процесса не являются независимыми. Поэтому центральная теорема может дать сбой.

Явления «тяжелых» хвостов и высоких пиков являются весьма распространенными, для объяснения «тяжелых» хвостов были разработаны отдельные вероятностные распределения (Коши, Парето, Леви).

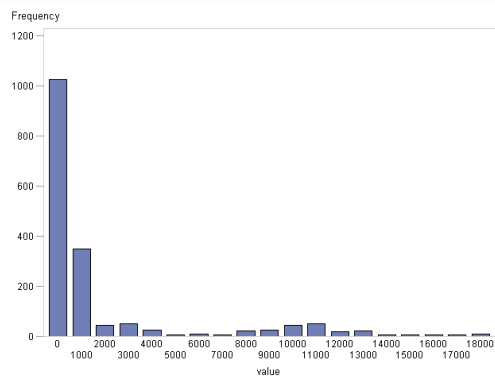


Рисунок 1 -- Пример данных с «тяжелыми» хвостами – 20-тидневный индекс Доу–Джонса

Для проведения качественного исследования, необходимо понять природу процесса, по возможности провести анализ политических, экономических, социальных составляющих, которые повлияли на поведение исследуемого процесса [Schwager,2012].

2. Применение фрактального анализа – анализ финансовых данных предприятия

Для наглядного примера возьмем данные ежедневных финансовых потоков предприятия «А» за 2014 год и проведем фрактальный анализ (данные и программу на языке SAS можно найти на сайте https://sites.google.com/site/sas4ua2015/romanko_alex)

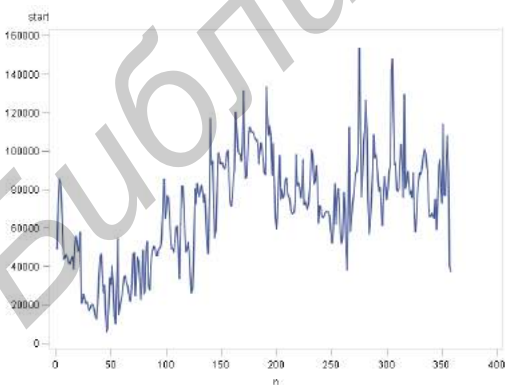


Рисунок 2 -- Данные о входящих денежных потоках (income) предприятия за 358 дней

3. Метод поиска параметра Хёрста. R/S анализ и V-статистика

Как упоминалось выше, некоторые временные

ряды недостаточно хорошо описываются стандартной гауссовой статистикой, поскольку системы изначально не являются независимыми и идентично распределенными (independent and identically-distributed, IID).

Х. Е. Хёрстом была открыта непараметрическая методология, позволяющая изучать системы без предварительных предположений о форме распределения вероятностей. Хёрст включил в свое исследование многие естественные системы и дал нам новую статистическую методологию для различения случайных и неслучайных систем, постоянства трендов и продолжительности циклов если таковые имеются [Mandelbrot and others,2008]. Идея метода заключается в том, что рассчитывается так называемый «нормированный размах», который показывает расстояние S , на которое перемещается система за дискретное время n . Известно, что случайное блуждание (например случайное броуновское движение молекул) за время n проходит расстояние $S = c * \sqrt{n}$ (где c – произвольная константа). Хёрст обобщил это уравнение для систем, не являющихся «IID»: $S = c * n^H$, где $H \in [0,1]$. Следовательно, мы можем выполнить простую линейную регрессию на $\log(n)$

как независимой переменной и $\log(r/s)n$ как зависимой переменной. Наклон уравнения является оценкой показателя Хёрста H .

Алгоритм R/S анализа следующий:

1. Начните с временного ряда длины M . Преобразуйте его во временной ряд длины $N = M - 1$ из логарифмических отношений

$$N_i = \log\left(\frac{M_{i+1}}{M_i}\right), i = 1, 2, 3, \dots, (M - 1). \quad (1)$$

Или используя AR(1) разности - построив регрессию M_{i+1} как зависимой переменной против M_i как независимой переменной.

$$N_i = M_{i+1} - (c + a \cdot M_i), i = 1, 2, 3, \dots, (M - 1). \quad (2)$$

2. Разделите этот период времени на A смежных подпериодов длины n , так что $A * n = N$. Пометьте каждый подпериод I_a , с учетом того, что $a=1, 2, 3, \dots, A$. Каждый элемент в I_a помечен N_k при этом $k = 1, 2, 3, \dots, n$. Для каждого I_a длины n среднее значение определяется как:

$$E_a = (1/n) \sum_{k=1}^n N_{k,a} \quad (3)$$

где E_a - среднее значение N_k содержащегося в подпериоде I_a длины n .

3. Временной ряд накопленных отклонений ($X_{k,a}$) от среднего значения для каждого подпериода I_a определяется как:

$$X_{k,a} = \sum_{i=1}^k (N_{i,a} - e_a), k = 1, 2, \dots, n. \quad (4)$$

4. Диапазон определяется как максимальное значение за вычетом минимального значения $X_{k,a}$ в пределах каждого подпериода I_a :

$$R_{k,s} = \max(X_{k,a}) - \min(X_{k,a}), k = 1, 2, 3, \dots, n. \quad (5)$$

5. Выборочное стандартное отклонение, рассчитываемое для каждого подпериода I_a :

$$S_{ia} = \left(\frac{1}{n} \cdot \sum_{k=1}^n (N_{k,a} - e_a^2)^{0.5} \right). \quad (6)$$

6. Каждый диапазон R_{ia} теперь нормализуется путем деления на соответствующий S_{ia} . Поэтому повторно нормированный размах в течении каждого I_a подпериода равен R_{ia} / S_{ia} . В шаге 2 выше мы получили смежные подпериоды длины n . Следовательно, среднее значение R/S для длины n определяется как:

$$(R/S)_n = \frac{1}{A} \cdot \sum_{a=i}^a (R_i / S_{i,a}). \quad (7)$$

7. Длина n увеличивается до следующего более высокого значения, а $(M-1)/n$ является целочисленным значением. Мы используем значения n , включающие начальные и конечные точки временного ряда, и шаги 1-6 повторяются до $n = (M - 1)/2$. Теперь мы можем выполнить простую регрессию методом наименьших квадратов на $\log(n)$ как независимой переменной и $\log(r/s)_n$ как зависимой переменной. Отрезок, отсекаемый на координатной оси, является оценкой $\log(c)$, константой. Наклон уравнения является оценкой показателя Хёрста H .

Ожидаемое значение нормированного размаха с поправкой Энниса и Ллойда [Peters, 1994]:

$$E(R/S_n) = \frac{n-0.5}{n} \left(n \cdot \frac{\pi}{2} \right)^{-0.5} \sum_{r=1}^{n-1} \sqrt{\frac{n-r}{r}}. \quad (8)$$

Ожидаемое значение соответствует размаху полностью случайного независимого процесса (то есть с параметром Хёрста, равным 0.5).

Variable	DF	Estimate
Intercept	1	0.1947
Hurst_par	1	0.3285

Рисунок 3 -- Параметр Хёрста, определенный с помощью авторегрессии-0.3285

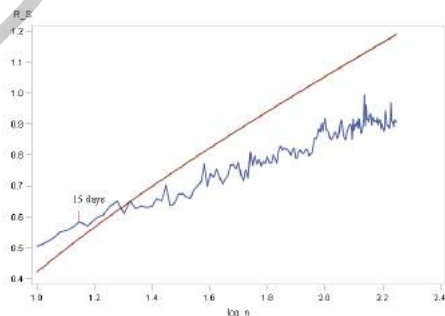


Рисунок 4 -- График результатов R/S – анализа

Горизонтальной оси графика соответствует $\lg(n)$, вертикальной - $\lg(R/S_n)$. На графике видно что реальные значения размаха (синяя ломанная) лежат ниже ожидаемых значений (красная линия). Это значит, наш процесс имеет длинную память (поскольку его размах намного отклоняется от ожидаемого) и антиперсистентен (поскольку синяя ломанная лежит ниже красной линии).

Теперь воспользуемся V-статистикой. Как было сказано ранее, она позволяет находить среднюю длину неперiodических циклов [Peters, 1994]. Простейшим примером естественных неперiodических циклов являются смены периодов года (ведь мы не знаем наверняка, сколько дней будет длиться осень, а сколько – зима, но знаем среднюю продолжительность времен года).

Общая идея V-статистики – когда цикл какого-либо процесса начинает закончился, нормированный размах (R/S_n) перестает расти (поскольку значения опять начинают повторяться, следовательно не вносят изменений в диапазон значений). Если вычертить R/S_n против $\lg(n)$, то при окончании цикла, график начнет стремительно падать вниз.

Конечная формула: $V_n = \frac{R/S_n}{\sqrt{n}}$. Это отношение приведет к горизонтальной линии, если процесс случаен. Если он персистентен, линия на графике будет наклонена вверх. График будет иметь наклон вниз, если процесс антиперсистентен.

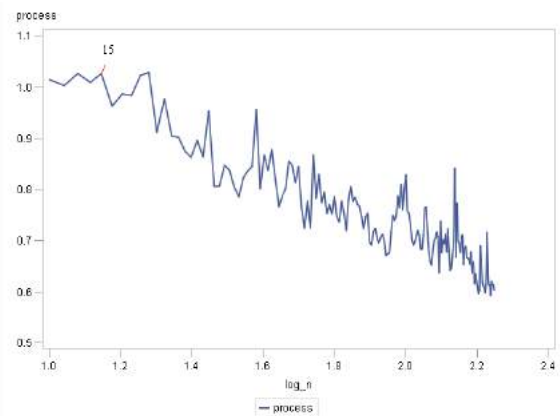


Рисунок 5 График V-статистики процесса

Как мы видим на рис 7, первый «разрыв» на графике (резкое его снижение), соответствует $n=15$. Действительно, если посмотреть на график приростов y_k , можно увидеть похожие друг на друга паттерны, длина которых составляет от 13 до 17 дней.

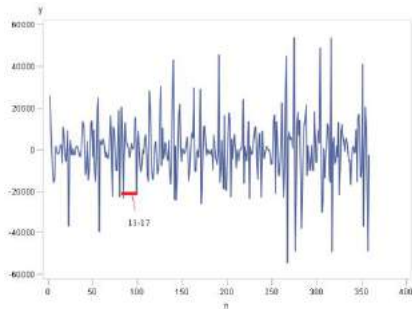


Рисунок 6 -- График приростов u_k

Заключение

В данной работе описан принципиально новый подход к анализу временных рядов, для которых корреляционная функция постепенно уменьшается со временем («длинная память»).

Использование новых инструментов фрактального анализа, таких как R/S – анализ и V-статистика дает более глубокое понимание природы временного ряда и его особенностей, возможность исследовать периодические и непериодические циклы. При развитии и дополнении фрактальных инструментов, они открывают новые возможности анализа и прогнозирования временных рядов различной природы: всевозможных финансовых индексов, финансовых потоков, различных природных явлений.

Библиографический список

[Mandelbrot and others,2008] Mandelbrot, Benoit B., and Richard L. Hudson. Misbehaviour of Markets. Profile Books, 2008.

[Peters,1994] Peters, Edgar E. Fractal market analysis: applying chaos theory to investment and economics. Vol.

[Schwager,2012] Schwager, Jack D. Market wizards: Interviews with top traders. John Wiley & Sons, 2012.

COGNITIVE FRACTAL ANALYSIS OF TIME SERIES: R / S ANALYSIS AND V-STATISTICS

Romanko A.R.

National University of Ukraine «Kiev Politechnic Institute»

romanko1993@gmail.com

Fractal Time Series Analysis - a new methodology using modern vision of the financial market and fractal mathematics tools for analyzing financial and economic processes. There are descriptions and algorithms of a new approach to the analysis of time series - methodology of R/S analysis and V-statistic.

Introduction

The emergence of a new fractal approach is due to the discovery of some of the misconceptions about the nature and behavior of many financial processes.

Firstly, it is the widespread misuse of the fundamental theorem of statistics - the central limit theorem. But many processes are not independent and

identically distributed - so the central theorem can not be used.

Secondly, the use of fractal approach is due to the emergence of a new concept of the behavior of the global market, and financial markets in particular. Classical theory is that markets are "fair", where participants are guided only by the principle of maximizing profits therefore fluctuations in the market are mostly random, aimed at stabilizing processes.

Practical approach

The modern approach to business in any sphere of human activity is, above all, the ability to anticipate or predict the near future developments. The quality and reliability of doing business depends on the most accurate answer to the question "what will happen tomorrow?".

To answer this question correctly, we must do some fundamental and technical research, "understand" data. The goal is to aggregate all knowledge about some process into one logical whole

Following analysis tools, that help us to gain some considerable knowledge on how to use the information available, are outlined:

- The method of R/S analysis allows the researcher to identify a process with a long memory, and classify according to its characteristics (persistent or antipersistent) and find the Hurst exponent.
- V-statistic allows to investigate the average length of non-periodic cycles that occur in the process due to the seasonality of certain phenomena.

The Hurst exponent $H \in (0,1)$ – shows the nature of change in the process. When $H \in (0.5,1]$ the process is persistent, it maintains the existing tendency, i.e. the increase in the past are more likely to lead to an increase in the future, and vice versa. Stock brokers describe this behavior by the emergence of "trends" and "structural breaks."

When $H \in [0,0.5)$ the process is called antipersistent, any tendency tends to be replaced by its counterpart. If $H=0.5$ it is a completely random process, a clear trend is not expressed. It is worth noting that natural processes have $H = 0.72-0.73$

Conclusion

This paper describes a new approach to the analysis of time series, for which the correlation function decreases gradually with time ("long memory").

Using the new fractal analysis, such as R / S - analysis and V-statistics gives a better understanding of the nature of the time series and its features, the ability to investigate non-recurring and periodic cycles.

With the development and addition of fractal tools, they open up new possibilities of analysis and forecasting of time series of different nature: all kinds of financial indices, financial flows of various natural phenomena.



УДК 004.89

СЕМАНТИЧЕСКОЕ КОДИРОВАНИЕ НА ОСНОВЕ ГЛУБОКИХ АВТОАССОЦИАТИВНЫХ НЕЙРОННЫХ СЕТЕЙ

Головки В.А., Крощенко А.А., Хацкевич М.В.

*Брестский государственный технический университет,
г. Брест, Республика Беларусь*

gva@gmail.com

kroschenko@gmail.com

В работе предлагается подход к реализации семантического кодирования (хеширования) с помощью глубоких автоассоциативных нейронных сетей. Построенная на ее основе поисковая система отличается простотой и скоростью осуществления поиска релевантных изображений.

Ключевые слова: нейронные сети глубокого доверия; глубокие автоассоциативные сети; семантическое хеширование; анализ и распознавание изображений; ограниченная машина Больцмана

Введение

Вопросы семантического кодирования имеют первостепенное значение при реализации поиска по образцу. При осуществлении поиска среди изображений особое значение приобретает проблема скорости выполнения такой операции.

Теория нейронных сетей предоставляет для решения данной задачи модели, называемые автоассоциативными нейронными сетями. Применение данных моделей позволяет сжать исходные изображения и выделить наиболее информативную часть на среднем слое («бутылочном горлышке»). Обученная сеть позволяет соотнести с каждым изображением код, позволяющий оценить близость данного изображения с уже оцененными. Применение подобных сетей для бинарного кодирования изображений имеет ряд преимуществ – прежде всего быстрый поиск, на который не влияет размер базы данных [Krizhevsky et al., 2011].

Благодаря работам [Hinton et al., 2006a], [Hinton, 2002], [Hinton et al., 2006b], [Hinton, 2010], стало возможным осуществлять обучение сетей, характеризующихся большим количеством слоев. Эти сети получили название нейронных сетей глубокого доверия (Deep Belief Networks – DBN). Глубокие автоассоциативные нейронные сети представляют собой тип сетей, включающий ключевые особенности и нейронных сетей глубокого доверия, и автоассоциативных сетей. Модели этого класса обладают чертами, наличие

которых сделало возможным использовать их в решении задач, характеризующихся большими (гигабайтными) объемами данных.

В данной работе рассматривается подход к реализации семантического кодирования на основе глубокой автоассоциативной нейронной сети. В качестве обучающей выборки выступает база изображений CIFAR-10.

1. Нейронные сети глубокого доверия

1.1. Общая структура. Обучение

Нейронная сеть глубокого доверия содержит множество скрытых слоев (рисунок 1) и осуществляет глубокое иерархическое преобразование входного пространства образов.

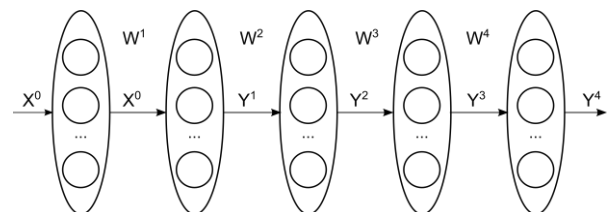


Рисунок 1 - Нейронная сеть глубокого доверия

Выходное значение j -го нейрона k -го слоя определяется следующим образом:

$$y_j^k = F(S_j^k), \quad (1)$$

$$S_j^k = \sum_{i=1}^n w_{ij}^k y_i^{k-1} + T_j^k, \quad (2)$$

где F – функция активации нейронного элемента, S_j^k – взвешенная сумма j -го нейрона k -слоя, w_{ij}^k – весовой коэффициент между i -ым нейроном $(k-1)$ -го слоя и j -м нейроном k -го слоя, T_j^k – пороговое значение j -го нейрона k -го слоя.

Для первого (распределительного) слоя

$$y_i^0 = x_i. \quad (3)$$

В матричном виде выходной вектор k -го слоя

$$Y^k = F(S^k) = F(W^k Y^{k-1} + T^k), \quad (4)$$

где W – матрица весовых коэффициентов, Y^{k-1} – выходной вектор $(k-1)$ -го слоя, T^k – вектор пороговых значений нейронов k -го слоя. Если нейронная сеть глубокого доверия используется для классификации образов, то выходные значения сети часто определяются на основе функции активации **softmax**:

$$y_j^F = \text{softmax}(S_j) = \frac{e^{s_j}}{\sum_l e^{s_l}} \quad (5)$$

Процесс обучения нейронных сетей глубокого доверия в общем случае состоит из двух этапов:

1. предобучение нейронной сети методом послыоного обучения, начиная с первого слоя (pre-training). Данное обучение осуществляется без учителя.
2. Настройка синаптических связей всей сети (fine-tuning) при помощи алгоритма обратного распространения ошибки или алгоритма «бодрствования и сна» [Hinton et al., 1995] (wake-sleep algorithm).

Ключевым этапом обучения нейронных сетей глубокого доверия, определяющим их преимущества перед поверхностными моделями, является предобучение. Один из основных подходов к предобучению – метод, базирующийся на представлении каждого слоя нейронной сети в виде ограниченной машины Больцмана (RBM), а всей сети – в виде стека таких машин.

1.2. Ограниченная машина Больцмана

Ограниченная машина Больцмана состоит из двух слоев стохастических бинарных нейронных элементов, которые соединены между собой двунаправленными симметричными связями (рисунок 2). Входной слой нейронных элементов называется видимым (слой X), а второй слой называется скрытым (слой Y). Нейронную сеть глубокого доверия можно представить как совокупность ограниченных машин Больцмана. Ограниченная машина Больцмана может генерировать (представить) любое дискретное распределение, если используется достаточное количество нейронов скрытого слоя [Bengio, 2009].

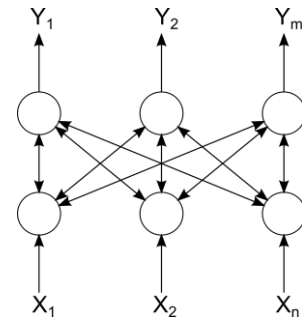


Рисунок 2 - Ограниченная машина Больцмана

Данная сеть является стохастической нейронной сетью, в которой состояния видимых и скрытых нейронов меняются в соответствии с вероятностной версией сигмоидной функции активации вида:

$$p(y_j = 1 | x) = \sigma\left(\sum_i w_{ij} x_i + T_j\right) \quad (6)$$

$$p(x_i = 1 | y) = \sigma\left(\sum_j w_{ij} y_j + T_i\right) \quad (7)$$

где $\sigma(x) = \frac{1}{1 + e^{-x}}$, w_{ij} – весовые коэффициенты, T_i, T_j – пороговые элементы.

RBM такой структуры получила название бинарной RBM.

Помимо представленной модели, ориентированной на использование с бинарными данными, существует модель линейно-бинарной RBM, в которой на видимом слое применяется линейная функция активации. Подобная RBM рекомендуется к использованию при работе с не бинарными (вещественными) данными [Hinton, 2010].

Для обучения линейно-бинарной RBM используются следующие правила:

$$p(y_j = 1 | x) = \sigma\left(\sum_i w_{ij} x_i + T_j\right) \quad (8)$$

$$x_i = N\left(\sum_j w_{ij} y_j + T_i, 1\right) \quad (9)$$

где $N(m, s)$ – нормально распределенная случайная величина со средним значением m и стандартным отклонением s .

Состояния видимых и скрытых нейронных элементов принимаются независимыми:

$$P(x | y) = \prod_{i=1}^n P(x_i | y)$$

$$P(y | x) = \prod_{j=1}^m P(y_j | x)$$

Таким образом, состояния всех нейронных

элементов ограниченной машины Больцмана определяются через распределение вероятностей. В RBM нейроны скрытого слоя являются детекторами признаков, которые сохраняют закономерности входных данных. Основная задача обучения состоит в воспроизведении распределения входных данных на основе состояний нейронов скрытого слоя как можно точнее. Это эквивалентно максимизации функции правдоподобия путем модификации синаптических связей нейронной сети [Головко, 2015a].

Получаемые в процессе вывода выражения для обучения RBM являются достаточно сложными в вычислительном плане, поэтому Хинтон предложил использовать аппроксимацию, которую он назвал контрастным расхождением (contrastive divergence (CD)) [Hinton et al., 2006a]. Такая аппроксимация основывается на дискретизаторе Гиббса (Gibbs sampling). В этом случае первые слагаемые в выражениях для градиента характеризуют распределение данных в момент времени $t = 0$, а вторые слагаемые характеризуют реконструированные или генерируемые моделью состояния в момент времени $t = k$. Исходя из этого, CD-k процедура может быть представлена следующим образом:

$$x(0) \rightarrow y(0) \rightarrow x(1) \rightarrow y(1) \rightarrow \dots \rightarrow x(k) \rightarrow y(k)$$

В результате можно получить следующие правила для обучения RBM-сети:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(k)y_j(k)) \\ T_i(t+1) &= T_i(t) + \alpha(x_i(0) - x_i(k)) \\ T_j(t+1) &= T_j(t) + \alpha(y_j(0) - y_j(k)). \end{aligned} \quad (10)$$

где $x_i(0)$, $y_j(0)$ характеризует распределение данных в момент времени $t=0$, $x_i(k)$, $y_j(k)$ - в момент времени $t=k$, k - параметр, определяющий количество стадий проходов «вперед-назад» для осуществления аппроксимации соответствующих градиентов.

В случае применения CD-1, $k = 1$ получим следующие правила обучения:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(1)y_j(1)) \\ T_i(t+1) &= T_i(t) + \alpha(x_i(0) - x_i(1)) \\ T_j(t+1) &= T_j(t) + \alpha(y_j(0) - y_j(1)). \end{aligned} \quad (11)$$

Из выражений (8), (9) видно, что правила обучения ограниченной машины Больцмана минимизируют разницу между оригинальными данными и данными генерируемыми моделью. Генерируемые моделью данные получаются при помощи процедуры сэмплирования Гиббса.

Для обучения RBM могут быть использованы и другие методы, например, [Golovko et al., 2014], [Головко, 2015b], [Головко и др., 2015c], [Golovko et al., 2015d].

1.3. Ограниченная машина Больцмана

Обучение нейронной сети глубокого доверия происходит на основе «жадного» алгоритма послойного обучения (greedy layer-wise algorithm). В соответствии с ним вначале обучается первый слой сети как RBM машина. Для этого входные данные поступают на видимый слой нейронных элементов и используя CD-k процедуру вычисляются состояния скрытых $p(y|x)$ и видимых нейронов $p(x|y)$. В процессе выполнения данной процедуры (не более 100 эпох) изменяются весовые коэффициенты и пороговые значения RBM сети, которые затем фиксируются. Затем берется второй слой нейронной сети и конструируется новая RBM машина. Входными данными для нее являются данные с предыдущего слоя. Происходит обучение и процесс продолжается для всех слоев нейронной сети [Hinton, 2009]. В результате такого обучения без учителя можно получить подходящую начальную инициализацию настраиваемых параметров сети глубокого доверия. На заключительном этапе осуществляется точная настройка параметров всей сети при помощи алгоритма обратного распространения ошибки или алгоритма «бодрствования и сна» (wake-sleep algorithm).

2. Глубокие автоассоциативные нейронные сети

Автоассоциативная сеть – это многослойная нейронная сеть прямого распространения сигнала, обученная для того, чтобы выдавать входные данные на выходе сети. При обучении автоассоциативной сети «учителем» для нее является сама входная информация. На первой половине сети происходит так называемое «прямое распространение», т.е. осуществляется сжатие входных данных. Далее на второй половине сети, которая является зеркальным отражением первой, происходит «обратное распространение», т.е. восстановление входного образа. Обычно сеть имеет скрытый слой меньшей размерности, который выделяет наиболее значимые признаки во входной информации. Пример такой сети представлен на рисунке 3.

Автоассоциативные сети с таким слоем оказываются полезны при решении задач визуализации и обработки данных высокой размерности, так как позволяют существенно сократить объем данных. В этой работе исследуются глубокие автоассоциативные сети.

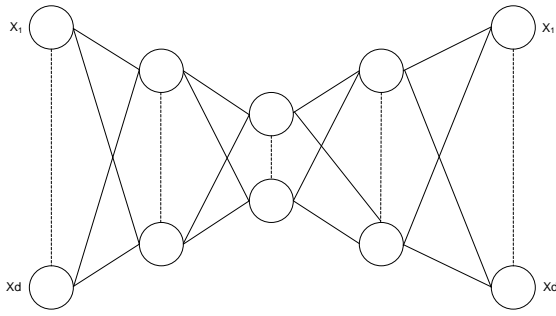


Рисунок 3 - Пример автоассоциативной сети

2.1. Обучение

В обучении глубоких автоассоциативных сетей также можно выделить два этапа:

1. Предобучение. На этом этапе выполняются действия, аналогичные производимым при предобучении обычных глубоких сетей с той лишь разницей, что предобучаются лишь те слои, которые производят кодирование информации.

2. Обучение (fine-tuning). На этом этапе предобученный кодировщик «разворачивается», формируя полную автоассоциативную сеть (рисунок 4). При этом в качестве матриц весовых коэффициентов для декодирующих слоев сети берутся транспонированные матрицы для соответствующих кодирующих слоев, а в качестве пороговых элементов выступают векторы порогов соответствующих видимых слоев RBM. После этого производится обучение получившейся автоассоциативной сети методом обратного распространения ошибки.

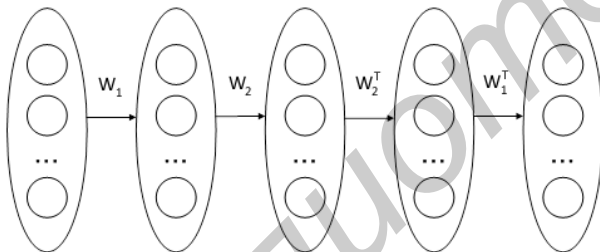


Рисунок 4 – Структура глубокой автоассоциативной сети

3. Применение глубоких автоассоциативных сетей для решения задачи семантического кодирования

3.1. Постановка задачи

В качестве обучающей выборки для решения задачи построения бинарных семантических кодов изображений нами была использована база CIFAR-10. Эта выборка включает в себя 50.000 изображений технических средств и живых существ, принадлежащий 10 различным классам (рис. 5). Каждое изображение имеет размер 32X32 пикселя.

Помимо этого, CIFAR-10 включает тестирующую выборку, состоящую из 10.000 изображений. Выборка предстает собой массив

байтовых значений. На сегодняшний момент задача классификации изображений данной выборки можно считать в сущности решенной [LeCun, 2014].

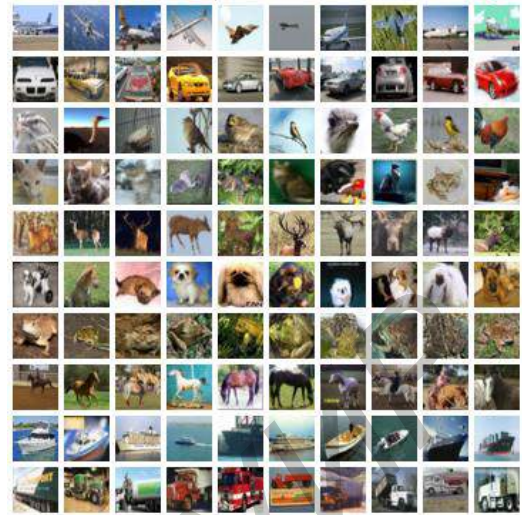


Рисунок 5 – Фрагмент базы изображений CIFAR-10

Семантическое кодирование (хеширование) относится к более сложному и не менее важному типу задач, позволяющему формировать бинарный код ограниченной длины, емко и однозначно описывающий изображение в редуцированном признаковом пространстве.

3.2. Структура сети и основные параметры обучения

Для решения задачи нами использовалась 10-слойная ассоциативная нейронная сеть с архитектурой 3072-4096-2048-1024-512-256-512-1024-2048-4096-3072. На среднем слое такой сети формируется вектор вещественных значений из отрезка $[0, 1]$, элементы которого затем округляются. Таким образом формируется бинарный код изображения. Легко видеть, что таким образом может быть закодировано 2^{256} изображений, что более чем достаточно для базы CIFAR-10.

Обучение проводилось в два этапа. На первом этапе предобучались соответствующие RBM, формирующие кодирующие слои сети. Исходные данные перед обучением были стандартизованы: из каждого компонента вектора изображения вычиталось его среднее значение по всем изображениям выборки и затем полученное значение делилось на стандартное отклонение по всем компонентам всех изображений. Данное преобразование определило тип первой обучаемой машины Больцмана – линейно-бинарная RBM. Остальные машины обучались как бинарная RBM.

Каждая RBM обучалась на протяжении 100 эпох мини-батчами по 100 элементов. Для линейно-бинарной RBM использовалась скорость обучения 0,001, для бинарной – 0,01. Помимо этого для ускорения процесса обучения использовался моментный параметр, равный 0,9.

После проведения предобучения выполнялось обучение развернутой автоассоциативной сети методом обратного распространения ошибки. Параметры обучения: скорость – $1e-6$, количество эпох обучения – 150, моментный параметр – 0.9.

После выполнения обучения, результат оценивался вычислением расстояния Хэмминга между бинарными кодами для тестового изображения и изображениями из базы (формула 10) После этого полученный ряд значений сортировался по возрастанию для выделения наиболее релевантных результатов.

$$H(v_1, v_2) = \sum_{i=1}^n |v_1^i - v_2^i| \quad (12)$$

Вычисления, производимые в процессе обучения глубокой автоассоциативной нейронной сети производились на видеокарте GTX 750 Ti и заняли около 90 минут.

3.3. Результаты обучения

Мы протестировали нашу модель двумя способами. Первый вариант предусматривал визуальное сравнение оригинального и восстановленного изображения. Некоторые из выполненных тестов представлены на рисунке 6.



Рисунок 6 – Оригинальные и восстановленные нейронной сетью изображения

Второй вариант тестирования заключался в подаче на обученную автоассоциативную сеть изображения, получения его бинарного кода с промежуточного слоя сети и вычисления расстояния Хэмминга для всех остальных изображений. Отсортировав получившуюся последовательность по возрастанию, можно изучить наиболее релевантные результаты поиска (рис. 7-8).

Исходя из представленных данных, можно отметить, что с увеличением расстояния Хэмминга количество изображений того же класса, что и

целевое изображение постепенно уменьшается.

	43	61	61
	64	64	65
	65	68	69

Рисунок 7 – Поискный шаблон и результаты поиска с расстояниями Хэмминга (тест №1)

	56	62	63
	66	70	72
	74	74	75

Рисунок 8 – Поискный шаблон и результаты поиска с расстояниями Хэмминга (тест №2)

Заключение

В последнее время в мире наблюдается активизация исследований в области нейронных сетей глубокого доверия. Эта безусловно перспективная область занимает достойное место в теории нейронных сетей. Из года в год увеличивается количество публикаций по этой тематике. Появляются целые книги, посвященные глубокому обучению (например, [Masters, 2015]). При этом особое значение приобретают скорость обучения таких сетей. Гетерогенные вычисления

уже давно стали стандартом де-факто при обучении таких типов сетей. При реализации практической части этой работы ключевое значение имело использование видеокарты для выполнения вычислительной работы. Это позволило существенно ускорить процесс обучения и провести большое количество экспериментов за приемлемое время.

В данной работе предлагается подход к семантическому кодированию изображений. На основе данного подхода может быть реализована полноценная поисковая система изображений. На текущем этапе реализованная простейшая система поиска позволяет находить релевантные результаты по поисковому запросу, сформированному в виде целевого изображения. Хочется отметить, что уровень pertinентности системы пока невысок. Его повышение является дальнейшим направлением работы.

Библиографический список

- [Krizhevsky et al., 2011] Krizhevsky, A. Using Very Deep Autoencoders for Content-Based Image Retrieval / A. Krizhevsky, G. Hinton // European Symposium on Artificial Neural Networks, Bruges, Belgium
- [Hinton et al., 2006a] Hinton, G. A fast learning algorithm for deep belief nets / G. Hinton, S. Osindero, Y. Teh // Neural Computation. – 2006. – Vol. 18. – P. 1527–1554.
- [Hinton, 2002] Hinton, G. Training products of experts by minimizing contrastive divergence // Neural Computation. – 2002. – Vol. 14. – P. 1771–1800.
- [Hinton et al., 2006b] Hinton, G. Reducing the dimensionality of data with neural networks / G. Hinton, R. Salakhutdinov // Science, 313 (5786). – 2006. – P. 504–507.
- [Hinton, 2010] Hinton, G. A practical guide to training restricted Boltzmann machines // Tech. Rep. 2010-000. – Toronto: Machine Learning Group, University of Toronto, 2010.
- [Hinton et al., 1995] Hinton, G. The "wake-sleep" algorithm for unsupervised neural networks / G. Hinton, P. Dayan, B. Frey, R. Neal // Science. – 1995. – Vol. 268. – P. 1158–1161.
- [Bengio, 2009] Bengio, Y. Learning deep architectures for AI // Foundations and Trends in Machine Learning. – 2009. – Vol. 2(1). – P. 1–127
- [Головко, 2015a] Головко В.А. Применение нейронных сетей глубокого доверия для выделения семантически значимых признаков / В.А. Головко, А.А. Крощенко // Материалы V Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (Open Semantic Technologies for Intelligent Systems). – Минск : БГУИР, 2015. – с. 481–486.
- [Golovko et al., 2014] Golovko, V. A Learning Technique for Deep Belief Neural Networks / V. Golovko, A. Kroshchanka, U. Rubanau, S. Jankowski // in book Neural Networks and Artificial Intelligence. – Springer, 2014. – Vol. 440. Communication in Computer and Information Science. – P. 136–146.
- [Головко, 2015b] Головко, В.А. От многослойных перцептронов к нейронным сетям глубокого доверия: парадигмы обучения и применение / В.А.Головко // XVII Всероссийская научно-техническая конференция «Нейроинформатика-2015»: Лекции по Нейроинформатике. – М.: НИЯУ МИФИ, 2015. – С. 47–84.
- [Головко и др., 2015c] Головко, В.А.Метод обучения нейронной сети глубокого доверия и применение для визуализации данных / В. Головко, А. Крощенко// Комп'ютерно інтегровані технології: освіта, наука, виробництво.– 2015. – № 19. – С. 6–12.
- [Golovko et al., 2015d] Golovko, Vladimir. A New Technique for Restricted Boltzmann Machine Learning / Aliaksandr Kroshchanka, Volodymyr Turchenko, Stanislaw Jankowski, Douglas Treadwell // Proceedings of the 8th IEEE International Conference

IDAACS-2015, Warsaw 24-26 September 2015. – Warsaw, 2015 – P.182–186.

[Hinton, 2009] Hinton, G. Greedy layer-wise algorithm // Journal of Machine Learning Research 1. – 2009. – P. 1–40.

[LeCun, 2014] Convolutional Nets and CIFAR-10: An Interview with Yann LeCun [Электронный ресурс]. – Режим доступа: <http://blog.kaggle.com/2014/12/22/convolutional-nets-and-cifar-10-an-interview-with-yan-lecun/>. – Дата доступа 01.11.2015.

[Masters, 2015] Masters, T. Deep Belief Nets in C++ and CUDA C: Volume 1: Restricted Boltzmann Machines and Supervised Feedforward Networks. – 1 edition. – CreateSpace Independent Publishing Platform, 2015. – 244 p.

SEMANTIC CODING BASED ON DEEP AUTOASSOCIATIVE NEURAL NETWORK

Golovko V.A., Kroshchanka A.A.,
Hackevich M.V.

Brest State Technical University

gva@bstu.by

kroschenko@gmail.com

An approach to implementation of semantic coding (hashing) using deep autoassociative neural network is proposed. The system based on the deep neural network is simple and fast concerning search relevant images.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

НЕЙРОННЫЕ СЕТИ В СЕМАНТИЧЕСКОМ АНАЛИЗЕ

Аверкин А.Н. *, Ярушев С.А. **

**Вычислительный центр им. А.А. Дородницына
Российской академии наук
Федерального исследовательского центра
«Информатика и управление»
Российской академии наук,
г. Москва, Россия
averkin2003@inbox.ru*

***Международный университет природы, общества и человека «Дубна»,
г.Дубна, Московская область, Россия
Sergey.Yarushev@icloud.com*

В работе проводится исследование по возможностям применения искусственных нейронных сетей в семантическом анализе. Рассматривается современное состояние дел в данной отрасли, перспективы использования искусственного интеллекта в области семантического анализа, направления и тенденции развития науки в данном направлении. Приводится обзор некоторых работ в области семантического анализа с применением технологий искусственного интеллекта и нейронных сетей.

Ключевые слова: семантический анализ, нейронные сети, искусственный интеллект, прогнозирование.

Введение

Обработка информации на естественном языке, анализ взаимосвязей между коллекцией документов и терминами, представленными в данном документе, понимание и определение направления и тематики текста – все это задачи семантического анализа. Латентно-семантическим анализом пользуются поисковые гиганты, чтобы находить тексты одной тематики. Множество работ ведется в области построения семантических моделей для обработки качества текста, понимания логических взаимосвязей, оптимизации баз знаний, а также широчайшего спектра задач.

Как дети учат язык? В частности, как они связывают структуру предложения с его значением? Данный вопрос непременно связан с более глобальным вопросом – каким образом мозг связывает последовательность символов для построения символических и суб-символических представлений? Многие ученые проводят исследования, чтобы получить ответы на данные вопросы. Одним из самых значимых аспектов в задаче обработки естественного языка является скорость, с которой происходит эта обработка. Наиболее четко эта проблема проявляется в

исследованиях потенциальных способностей мозга. Наиболее подходящим подходом к моделированию данных процессов является использование нейронных сетей. Подтверждением этого может быть большое количество работ в данной тематике.

В данной работе рассматривается ряд задач в области семантического анализа с применением искусственных нейронных сетей.

1. Обзор исследований по использованию нейронных сетей в задачах классификации текста

Китайские ученые [BoYoetal., 2007] в своем исследовании представили новую модель классификации текста с использованием нейронной сети и её обучением методом обратного распространения ошибки, а также с модифицированным методом. Используется эффективный метод выбора характеристик для уменьшения размерности выборки, что за собой несет повышение производительности системы. Стандартный алгоритм обучения по методу обратного распространения ошибки довольно медленно обучается, поэтому исследователи модифицировали данный алгоритм для увеличения

скорости обучения. Традиционные слово-сопоставления на основе классификации текста используют модель векторного пространства. Тем не менее, в данном подходе не учитываются семантические отношения между терминами, которые могут приводить к ухудшению точности классификации. Латентно-семантический анализ может преодолеть проблемы, вызванные использованием статистически полученных концептуальных индексов, а не только отдельных слов. Он создает концептуальные векторные пространства, в которых каждый термин или документ представляется в виде вектора в пространстве. Это не только значительно уменьшает размерность, но также позволяет обнаруживать важные ассоциативные отношения между терминами. Исследователи протестировали свою модель на наборе из 20 новостных данных, экспериментальные результаты показали, что модели с модифицированным методом обратного распространения ошибки, предложенным в данной работе, превзошли традиционный метод обучения. А также применение латентно-семантического анализа для данной системы позволяет резко сократить размерность, что позволяет достичь хороших результатов классификации.

Как видно из исследования, использование модифицированного алгоритма обучения нейронной сети, вкуче с семантическим анализом дает хорошие результаты в задаче классификации текста.

Индийские ученые [Tekwanietal., 2014] провели сравнение производительности обыкновенной нейронной сети обратного распространения ошибки с комбинацией данной нейронной сети с методом латентно-семантического индексирования в задаче классификации текста. В традиционной нейронной сети с обратным распространением ошибки, процесс настройки весов блокируется в локальном минимуме, а также, скорость обучения данного типа сетей довольно низка, что влечет за собой снижение производительности. В связи с данными недостатками, ученые решили сделать комбинацию латентно-семантического индексирования и данной нейронной сети. Латентно-семантические представления в структуре данных в низко-мерном пространстве, в котором документы, термы и последовательности слов также сравнивались. Одномерная декомпозиционная техника используется в латентно-семантическом анализе, в котором многомерные матрицы термов разбиваются в набор K ортогональных факторов, в которых оригинальные текстовые данные изменены до меньшего семантического пространства. Новый вектор документов можно найти в K -мерном пространстве. Так же, находятся новые координаты запросов.

Производительность комбинации данных методов проверялась на основе методики классификации 20 новостных групп из разных

категорий, таких как спорт, медицина, бизнес, политика и др. В итоге, данный метод позволяет значительно снизить размерность и получить лучшие результаты классификации текста.

2. Нейронные сети в задачах обработки естественного языка

2.1. Естественные нейронные сети в обработке языка

Одна из ключевых фигур в исследованиях головного мозга человека является В. Маунткасл [Mountcastle V., 1997]. В данной работе, он обобщил свои многолетние исследования, он утверждает, что, несмотря на разнообразие своих функций, все разделы коры головного мозга устроены, в принципе, одинаково. Это означает, что обучение и распознавание образов в коре происходит единообразно, а разнообразие ее функций есть следствие разнообразия сигналов, обрабатываемых разными участками коры.

Согласно Маунткаслу, кора имеет двумерную ячеистую структуру. Базовым функциональным элементом коры является мини-колонка диаметром около 30 мкм, состоящая из примерно 100 нейронов. Такие мини-колонки связаны между собой положительными и отрицательными латеральными связями. Причем, последние включаются резко, но с неким запаздыванием относительно первых. В результате одновременно возбуждается целый пул соседних мини-колонок, невольно заставляя вспомнить самоорганизующиеся карты Т. Кохонена [KohonenT., 2001]. В итоге, повсюду в коре мы наблюдаем самоорганизующиеся карты признаков: детекторы схожих сигналов располагаются рядом друг с другом.

Эксперименты свидетельствуют, что площадь элементарных детекторов на этих картах порядка 0.1 мм², т.е. они содержат 102 мини-колонок или 104 нейронов. Такие функциональные единицы Маунткасл называет макро-колонками. Именно они определяют «разрешающую способность» коры и предельное число признаков, которые может запомнить человек (всего несколько миллионов). Зато надежность этой памяти гарантируется большим числом нейронов, составляющих макро-колонку. Так что мы сохраняем свою память на протяжении всей жизни даже при гибели существенной части нейронов.

Таким образом, карты Кохонена являются, по-видимому, наиболее подходящим инструментом для моделирования работы коры. Надо только научить их работе с динамическими паттернами, с которыми только и работает мозг, т.к. его основная задача – предвидение.

2.2. Исследования в задачах обработки языка и предложений

Как человек овладевают языком, а также двумя или более разными языками с одной нервной

системой, до сих пор остается открытым вопросом. Для решения данной проблемы, французские ученые, во главе с Питером Домни [X.Hinautetal., 2015] предложили построить модель, которая будет способна изучать любой язык с самого начала. В данной работе они предлагают нейросетевой подход, который обрабатывает предложения по слову, слово за словом без предварительного знания семантики слов. Предлагаемая модель не имеет «предварительно связанную» структуру, а только случайную и обученные соединения, модель основана на технологии ReservoirComputing. Ранее учеными была разработана модель для робототехнических платформ, благодаря которой, пользователи могут научить робота основам английского языка, чтобы в дальнейшем давать ему различные задания. В данной работе была добавлена способность обрабатывать редкие слова для того, чтобы можно было сохранить размер словаря довольно маленьким при обработке естественного языка. Более того, данный подход был распространен на Французский язык и показано, что нейронная сеть может изучать два языка одновременно. Даже при небольшом корпусе языка, модель способна обучаться и обобщать знания в условиях моноязычности или двуязычности. Данный подход может быть более практичной альтернативой для небольших корпусов различных языков чем другие обучающие методы, опирающиеся на наборы больших данных.

Множество исследований проводится в области обработки языка с помощью нейронных сетей [Miikkulainen R., 1996], а также в последнее время с использованием так называемых EchoStateNetworks[Frank, 2006].

Как человеческий мозг обрабатывает предложения, которые человек читает или слышит? Задача понимания того, как мозг это делает, занимает одно из центральных мест в исследованиях ученых из данной области. Обработка предложений происходит в режиме реального времени. Предыдущие слова в предложении могут влиять на время обработки в сотни миллисекунд. Последние нейрофизиологические исследования позволяют предположить, что именно лобная часть головного мозга играет решающую роль в этом процессе. XavierHinaut [X.Hinautetal., 2013] провел исследование, которое дает некоторое понимание того, как определенные аспекты в данной обработке предложений в реальном времени происходят, основываясь на динамике периодических корковых сетей и пластичности в кортико-полосатой системе. Они моделируют префронтальную область ВА47 при помощи рекуррентной нейронной сети, получая он-лайн вход категорий слов в процессе обработки предложений. Система обучается по парам предложений, в которых закодирован смысл как функция активации, соответствующая той роли, которую играют глаголы и существительные в предложениях. Модель изучает расширенный набор грамматических конструкций и демонстрирует

возможность для генерации новых конструкций. Это демонстрирует, насколько рано в предложении параллельный набор предикатов создает смысл. Модель демонстрирует, как он-лайн отклики на слова подвержены влиянию от предыдущих слов в предложении и предыдущие предложения в дискурсе, обеспечивающие новый взгляд на нейрофизиологию коры головного мозга для распознавания грамматической структуры. Исследование показало, что рекуррентные нейронные сети могут декодировать грамматическую структуру из предложений в реальном времени с целью получения представления о значении предложений. Это может обеспечить понимание основных механизмов кортико-полосатой функции головного мозга человека в обработке предложений.

Нейросетевая обработка естественного языка. Центральное внимание в когнитивной науке сегодня сконцентрировано на исследовании того, как нейронные сети в головном мозге используются для чтения и понимания текста. Данный вопрос исследуется огромным количеством ученых по нейрофизиологии на ряду с недавними исследованиями, которые призваны обследовать процессы головного мозга, вовлеченные в обучение неязыковых последовательностей или искусственного обучения грамматике. PeterFordDominey [Domineyetal., 2008] в своем исследовании предпринял попытку совместить данные с несколькими нейрофизиологическими моделями обработки предложений, через спецификации нейросетевой модели, архитектура которой основана на известной кортико-стриато-таламо-кортикальной (КСТК) нейроанатомии системы человеческого языка. Задача состоит в том, чтобы разработать имитационные модели, учитывающие ограничения и нейроанатомической связи, и функциональные данные изображений. В предлагаемой модели, структурные кии, закодированные в рекуррентной кортиковой нейронной сети в ВА47, активируют схему (КСТК) для модулирования потока лексической семантической информации в интегрированное представление смысла на уровне предложений. Результаты моделирования продемонстрированы в работе Caplan [Caplan D. etal.,1985].

Моделирование органа языка провел С. А. Шумский. В своей работе [Шумский, 2012], автор выдвигает три гипотезы: **первая гипотеза** состоит в том, что обработка временных рядов в коре осуществляется подобными модулями, распознающими типовые временные паттерны, каждый в своем входном потоке. Например, участок коры, ответственный за морфологический анализ слов, распознает порядка 10^5 слов и составляющих их морфем и слогов. Другой участок коры, определяющий структуру предложений, работает таким же образом, только с другим первичным алфавитом, каждый символ которого кодирует уже не букву, а целое слово. Этот участок запоминает характерные паттерны комбинирования слов в

грамматически правильные фразы. Согласно **второй гипотезе**, входом для следующего коркового модуля, ответственного за анализ временных структур более высокого порядка, служит сжатый таламусом выходной сигнал от предыдущего модуля. По **третьей гипотезе**, в «органе языка» существуют два взаимосвязанных канала «глубокого обучения»: грамматический и семантический. Аналогично дорсальному (анализ сцен) и вентральному (распознавание объектов) каналам анализа зрительной информации.

Для проверки данных гипотез, был создан программный комплекс «семантический процессор Голем», способный выявлять иерархии языковых паттернов при обучении на больших текстовых массивах. Обучение проводилось на текстовом массиве объемом 6 ГБ, состоящем из материалов русскоязычных интернет-СМИ. Чтобы приблизить условия эксперимента к обучению устной речи ребенком, все слова приводились к строчным буквам. Объем обучающей выборки примерно соответствует языковому опыту 20-летнего человека (при восприятии $\sim 10^5$ слов в день). Обучение заняло около двух месяцев работы современного ПК.

В результате, разработанный Шумским комплекс довольно уверенно может распознавать имена, фамилии, города, страны и некоторые другие понятия. Также, удалось добиться понимания и того, какие понятия в данном предложении соотносятся с какими, можно сказать, что Голем способен достаточно адекватно распознавать и индексировать смысловое содержание предложений.

Заключение

В данной работе проведен обзор современных работ в области исследования современного языка, исследования по изучению работы головного мозга и понимания им языка. Каждое исследование, содержит применение нейросетевых технологий в задачах семантического анализа и моделирования работы головного мозга.

Исходя из результатов, полученных в каждом исследовании, можно сделать вывод, что нейронные сети в задачах семантического анализа показывают высокую производительность и расширяют возможности в анализе текстовых данных, а также являются незаменимой технологией в задачах моделирование мозговой деятельности, в частности моделирование обучения новым языкам и применение данных технологий в построении роботов, способных самостоятельно изучать языки и понимать их смысл.

Работа выполнена при поддержке гранта: РФФИ №14-07-00603

Библиографический список

[Шумский, 2012] Шумский, С. А. Мозг и язык: Гипотеза о строении «Органа языка».

[Bo Yo et al., 2007] Yu B., Xu Z., Li C. Latent semantic analysis for text categorization using neural network // Knowledge-Based Systems. – 2008. – Т. 21. – №. 8. – С. 900-904.

[Dominey et al., 2008] Peter Ford Dominey, Toshio Inui, Michel Hoen. Neural network processing of natural language: II. Towards a unified model of corticostriatal function in learning sentence comprehension and non-linguistic sequencing // Brain and Language. – 2008 doi:10.1016/j.bandl.2008.08.002.

[Frank, 2006] Frank, S. L. (2006). Strong systematicity in sentence processing by an Echo State Network. In Proc. of ICANN 2006, pp. 505–514.

[X. Hinaut et al., 2015] Hinaut X. et al. A Recurrent Neural Network for Multiple Language Acquisition: Starting with English and French.

[X. Hinaut et al., 2013] Hinaut X., Dominey P. F. Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing // PloS one. – 2013. – Т. 8. – №. 2. – С. e52946.

[Kohonen T., 2001] Kohonen T. Self-Organizing Maps. Springer-Verlag. 2001.

[Miikkulainen R., 1996] Miikkulainen, R. (1996) Subsymbolic case-role analysis of sentences with embedded clauses. Cognitive Sci 20: 47–73.

[Mountcastle V., 1997] Mountcastle V. The columnar organization of neocortex // Brain. 1997. V. 120. P. 701–722

[Caplan D. et al., 1985] Caplan, D., Baker, C., & Dehaut, F. (1985). Syntactic determinants of sentence comprehension in aphasia. Cognition, 21, 117–175.

NEURAL NETWORKS IN SEMANTIC ANALYSIS

Averkin A.N. *, Yarushev S.Y. **

* *Institution of Russian Academy of Sciences
Dorodnicyn Computing Centre of RAS, Moscow,
Russia*

averkin2003@inbox.ru

** *Moscow region State Educational Institution for
higher professional education Dubna
International University for Nature, Society and
Man, Dubna, Russia*

Sergey.Yarushev@icloud.com

In this paper we study the possibilities of application of artificial neural networks in the semantic analysis. The current state of affairs in the industry, the prospects for the use of artificial intelligence in the field of semantic analysis, trends and tendencies of development of science in this direction. A review of some of the works in the field of semantic analysis with the use of artificial intelligence and neural networks.

Conclusion

Based on the results obtained in each study, we can conclude that the neural networks in the problems of semantic analysis shows high performance and extend the capabilities of the analysis of the text data. Also it is indispensable technology for modeling brain activity, in particular modeling of learning new languages and the use of Information technologies in building robots that can independently learn languages and understand their meaning.



УДК 519.711.74

НОВЫЙ ПОДХОД К МОДЕЛИРОВАНИЮ ФОРМАЛЬНЫХ НЕЙРОНОВ И ИХ ВЗАИМОДЕЙСТВИЙ

Жилякова Л.Ю.

*Институт проблем управления им. В.А. Трапезникова РАН,
г. Москва, Россия*

zhilyakova.ludmila@gmail.com

В работе предпринята попытка обогатить аппарат искусственных нейронных сетей за счет усложнения модели формального нейрона. Рассматриваются не только электрические, но и химические его свойства: чувствительность к различным нейротрансмиттерам и способность их производить. Сеть из этих нейронов – гетерогенная структура, связи в которой зависят от химической ситуации в межклеточном пространстве, которое также является частью модели. В такой сети транзмиттеры могут играть решающую роль в формировании паттернов активности.

Ключевые слова: неоднородная нейронная сеть; паттерн активности; гетерохимический ансамбль; модель формального нейрона.

Введение

Модель формального нейрона и нейронной сети была предложена У. Маккалоком и У. Питтсом [McCulloch, Pitts 1943]. Математически формальный нейрон – это пороговый элемент с единственным выходом, функция активации которого зависит от линейной комбинации всех входных сигналов. Изначально нейрон Маккалока–Питтса мог оперировать только бинарными сигналами: логическим нулем и логической единицей, однако в процессе развития нейронных сетей были предложены не только бинарные, но и непрерывные функции срабатывания. Нейронные сети успешно применяются в распознавании образов, классификации, кластеризации, прогнозировании, решении ряда вычислительных и оптимизационных задач.

В течение последнего десятилетия появились принципиально новые сетевые модели мозга. Их возникновение и большой всплеск количества исследований в этом направлении обусловлены сразу двумя факторами: появлением высокочувствительной регистрирующей техники, позволяющей получать большие качественные наборы данных, и появлением мощных компьютеров, способных эти данные обрабатывать. Оказалось, что сразу во многих биологических и социальных системах структура связей между их элементами описывается сложными сетями со сходными свойствами. Сети мозга не стали исключением. Теоретико-графовые исследования

сетей мозга получили названия «структурная и функциональная коннектомика» [Bullmore, Sporns 2009; Baronchelli et al 2013].

Искусственные нейронные сети и структурная коннектомика основываются на идее «проводочного мозга», в которой мозг представляется электрической сетью с жестко заданной топологией, которая образуется «проводами» (аксонами), соединяющими простые элементы (нейроны), причем, все нейроны одинаковы. Однако многие фундаментальные свойства, присущие живым нейронным сетям, в таких терминах описать невозможно. Современные исследования свидетельствуют о том, что нейроны не одинаковы – они являются транзмиттер-специфическими; и перестройка топологии сетей и изменение режимов активности нейронов происходят ad hoc – под действием нейротранзмиттеров [Bargmann, 2012; D'yakonova, 2014; Дьяконова, 2012; Сахаров, 2012].

Цель настоящей работы – описание формальной модели нейрона, обладающего не только электрическим зарядом, но и химическими входами и выходами. Сеть из таких нейронов будем называть гетерогенной нейронной сетью. При этом термин «сеть» не означает наличия лишь электрических связей – любая химическая связь в ней может быть отражена теми же математическими средствами.

1. Описание модели формального нейрона

1.1. Основные моделируемые свойства

В искусственных нейронных сетях формальный нейрон является пороговым суммирующим элементом, принимающим однородные сигналы и активирующимся в зависимости от линейной комбинации входов. Наша задача – обогатить эту модель некоторыми свойствами, присущими живым нейронам, и позволяющими реализовать новые функции, такие как выделение различных активных подсетей в зависимости от изменяющихся внешних условий и реализация на них паттернов циклической активности.

В данном разделе мы перечислим основные свойства естественных нейронов, которые станут объектами моделирования. Следует отметить, что все эти свойства претерпели ряд значительных упрощений в попытке выделить из них только информационно значимую составляющую. С этой оговоркой опишем процессы, происходящие при взаимодействии нейронов друг с другом и с окружающей средой.

Кроме электрических нейроны принимают химические сигналы, причем, их чувствительность к различным химическим веществам – нейротрансмиттерам (или нейромедиаторам) – зависит от наличия соответствующих рецепторов. Чем больше у нейрона рецепторов к определенному медиатору и чем выше их чувствительность, тем сильнее воздействие на него даже малых концентраций данного вещества.

В естественной нейронной сети нейроны находятся в среде, представляющей собой смесь нейротрансмиттеров, концентрации которых непрерывно изменяются: трансммиттеры выбрасываются в синаптическую щель при передаче импульса, при этом часть из них попадает во внеклеточное пространство и воздействует на другие нейроны; часть трансммиттеров утилизируется нейронами; часть приходит извне (см., напр., [Сукова, 2004]). Изменение соотношения концентраций веществ во внеклеточном пространстве воздействует на связи между нейронами, на очередность и скорость их активации, амплитуду и частоту спайков. Однако континууму состояний внеклеточного пространства соответствует конечное (и малое) число устойчивых состояний нейронной сети.

Диапазон медиаторов, к которым чувствителен единственный нейрон, достаточно широк. Один нейрон может иметь до нескольких десятков типов рецепторов. При этом, когда в нейроне происходит генерация спайка, он выпускает во внеклеточное пространство единственный трансммиттер или трансммиттерную смесь, состоящую из малого числа трансммиттеров (в модели предполагается, что число типов в выбрасываемой смеси не более пяти). Объем этого выброса может изменяться в зависимости от силы спайка, но состав остается

всегда одним и тем же; кроме этого, если типов трансммиттеров больше одного, их соотношение в смеси тоже постоянно.

В зависимости от химического состава внеклеточной среды изменяется чувствительность рецепторов нейронов. Вещество-антагонист некоторого трансммиттера может блокировать соответствующие рецепторы. Рецепторы имеют разное т.н. сродство: способность захватывать трансммиттер.

Рецепторы к одному и тому же трансммиттеру не однотипны. Известно, как минимум, три разных типа рецепторов: возбуждающие (деполяризующие мембрану клетки), тормозные (гиперполяризующие мембрану) и метаболитные (изменяющие состояние клетки и влияющие на мембранный потенциал отложено).

Как правило, нейроны имеют рецепторы к тому трансммиттеру, который сами выделяют. Таким образом, они управляют своей активностью за счет положительных или отрицательных обратных связей.

За счет комбинации в сети нейронов с различными типами рецепторов к одному и тому же трансммиттеру достигается большое разнообразие паттернов активности. Переключение между ними происходит не только из-за воздействия новых веществ. При разных концентрациях одного и того же вещества оно может действовать на один и тот же нейрон как возбуждающе, так и тормозящее. Это зависит от количества активных рецепторов, которое может изменяться с изменением концентрации.

Таким образом, состояние нейрона с информационной точки зрения описывается значением мембранного потенциала, наличием рецепторов и их чувствительностью.

1.2. Определения и обозначения

Опишем нейроны как сложные узлы, способные находиться в разных состояниях.

В модели задано множество типов (цветов) фишек $C = \{\bullet, \bullet, \dots, \bullet\}$, $|C| = m$. Цвета соответствуют типам трансммиттеров. Узлы некоторое количество рецепторов для фишек [Жилиякова, 2015]. Рецепторы чувствительны к фишкам «своего» цвета. Узлы могут обладать разными наборами рецепторов.

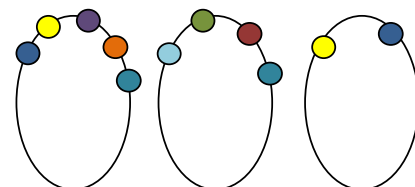


Рисунок 1 – Узлы с наборами рецепторов

Узлы могут принимать фишки в подходящие для них рецепторы.

При этом для каждого типа фишек существует три вида рецепторов:

- плюс-рецепторы (возбуждающие),
- минус-рецепторы (тормозные),
- ~-рецепторы (метаботропные) – на данном этапе построения модели они пока просто нейтральны.

Все рецепторы одного типа собраны в один слот. Их количество и чувствительность задаются весом слота (рисунок 2).

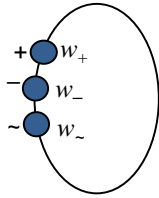


Рисунок 2 – Три вида слотов

Как правило, из трех весов слотов на рис.2 только один ненулевой. На каждый узел фишки определенного цвета имеют воздействие одного вида.

В сети вводится дискретное время t . На каждом такте слоты могут принимать фишки, после чего узел переходит в новое состояние и пересчитывает мембранный потенциал $U(t)$. $U(t)$ – агрегированная величина, зависящая от состояния узла на такте $t-1$ и от полученных на такте t фишек. Узлу приписывается пороговая величина мембранного потенциала U_T . Ее значение также может изменяться во времени: $U_T = U_T(t)$, но эти изменения относительно медленны. Если мембранный потенциал превысил пороговое значение, узел выстреливает фишками, что соответствует единичному спайку.

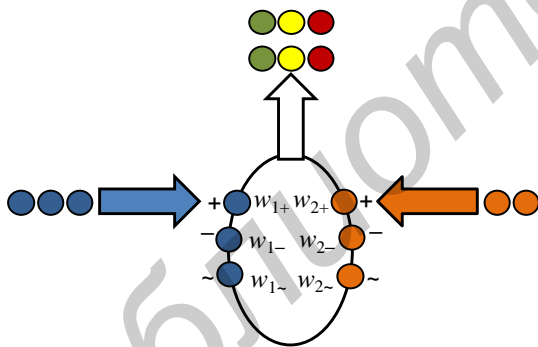


Рисунок 3 – Выброс фишек трех видов

Узел может выстреливать фишками с разной силой и периодичностью, но доля каждого вида фишек в смеси остается неизменной (рис. 4).

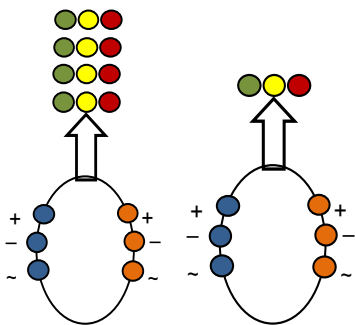


Рисунок 4 – Разные величины выбросов фишек

2. Организация нейронов в гетерохимические ансамбли

2.1. Внеклеточное пространство

Множество узлов, активируясь, выпускает наборы фишек, которые попадают во внеклеточное пространство (ВКП). В модели за ВКП отвечает выделенный (виртуальный) узел, который хранит выпущенные фишки. Этот узел связан со всеми остальными узлами (рис. 5).

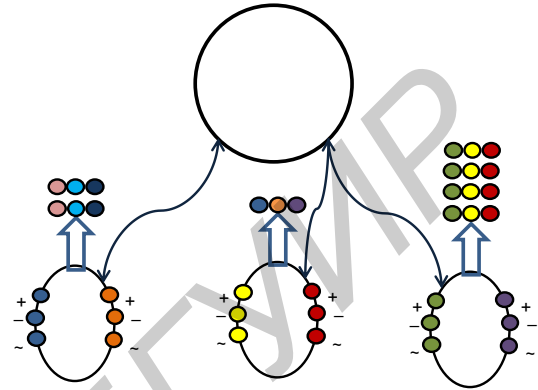


Рисунок 5 – Формальные нейроны и ВКП

Состав ВКП динамически изменяется. После того, как узлы выпустили фишки, они начинают их принимать. Если фишка принята, она исчезает из ВКП. Кроме того, в ВКП могут поступать фишки извне (рис. 6). В естественных сетях существуют дополнительные свойства утилизации фишек, однако на данном этапе они не реализованы в модели. При исследовании динамических процессов и моделировании изменения паттернов активности с большой вероятностью потребуются дополнительные механизмы отражения изменений в составе ВКП.

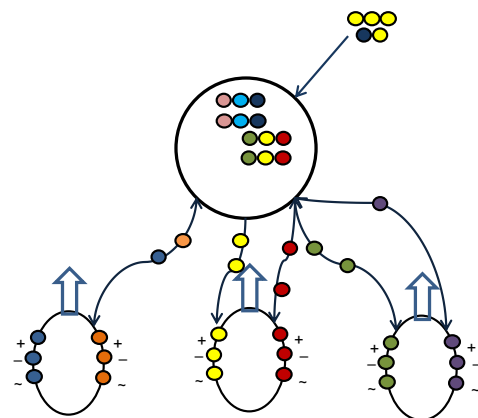


Рисунок 6 – Поступление фишек в слоты и изменение ВКП

2.2. Влияние состава ВКП на активность узлов

Преобладание в ВКП фишек некоторых типов включает одни наборы узлов сети и выключает другие. Таким образом, динамически формируются ансамбли из узлов. При этом между двумя узлами возникает ориентированная связь, если первый выстреливает определенными фишками, а второй имеет слот такого же типа. Эти раскрашенные

виртуальные связи делятся на усиливающие и тормозные. Сеть, построенная на таких принципах, обладает большой пластичностью и может по-разному реагировать на различные составы ВКП. Каждый информационно значимый в этой сети тип фишек (тип, к которому имеются слоты) может выделить свою подсеть (рис. 7).

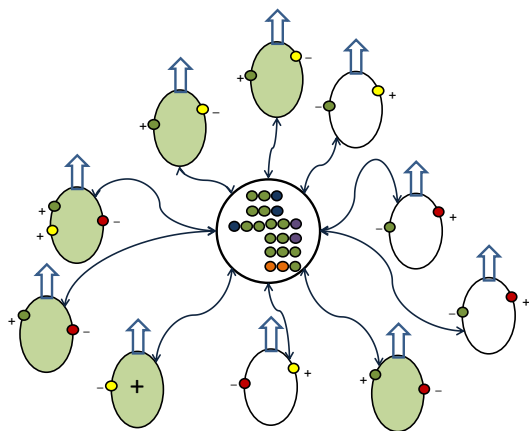


Рисунок 7 – Активация подсети отдельным типом фишек

Единичные фишки и определенные комбинации двух и более фишек могут задавать различные паттерны ритмической активности в такой сети.

Первоочередная задача, стоящая перед нами – обращение цикла трехфазного генератора с помощью различного состава ВКП. Схематически эта задача представлена на рис. 8.

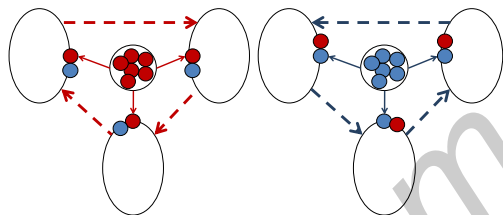


Рисунок 8 – Обращение обхода узлов в цикле

Заключение

В настоящей работе предложена модель формального нейрона, способного поддерживать не только электрические, но и химические взаимодействия. Описан аппарат объединения таких нейронов в разноцветные сети. Работа находится на начальной стадии, и разработка формального аппарата для реализации описанных в первой части доклада желаемых свойств – предмет будущих исследований. Однако хотелось бы отметить, что эти исследования ведутся достаточно интенсивно и мы надеемся, что эти результаты не заставят себя ждать.

Работа выполнена при частичной поддержке РФФИ (проекты № 14-01-00422а, 15-07-02488а).

Библиографический список

- [Bargmann, 2012] Bargmann, C.I. Beyond the connectome: how neuromodulators shape neural circuits. *Bioessays* 34, 458–465 (2012).
 [Baronchelli et al, 2013] Baronchelli, A., Ferrer-i-Cancho, R., Pastor-Satorras, R., Chater, N., Christiansen, M.H. Networks in

Cognitive Science // Trends in Cognitive Sciences. July 2013, Vol. 17, No. 7

[Bullmore, Sporns, 2009] Bullmore, E. Sporns, O. Complex brain networks: graph theoretical analysis of structural and functional systems // *Nature Reviews Neuroscience* 10, 186-198 (March 2009).

[D'yakonova, 2014] D'yakonova, V.E. Neurotransmitter mechanisms of context-dependent behavior. *Neurosci. Behav. Physiol.* 44, 256–267 (2014).

[McCulloch, Pitts, 1943] McCulloch W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 1943, v.5, pp.115-133. (Рус. пер. Маккаллоу У.С., Питтс У. Логическое исчисление идей, относящихся к нервной активности. // В кн. Автоматы. Под ред. К.Э.Шеннона и Дж.Маккарти. М., ИЛ, 1956, с. 362-384).

[Sykova, 2004] Sykova E. Extrasynaptic volume transmission and diffusion parameters of the extracellular space // *Neuroscience*. 2004; 129(4). P. 861-76.

[Дьяконова, 2012] Дьяконова В.Е. Нейротрансмиттерные механизмы контекст-зависимого поведения // *Журнал высшей нервной деятельности*, 2012, том 62, № 6. С. 1–17.

[Жилиякова, 2015а] Жилиякова Л.Ю. Сетевая модель распространения активности в среде гетерогенных автоматов / *Материалы V международной научно-технической конференции "Открытые семантические технологии проектирования интеллектуальных систем OSTIS-2015"*. Минск: БГУИР, 2015. С. 303-308.

[Жилиякова, 2015б] Жилиякова Л.Ю. Сетевая модель распространения нескольких видов активности в среде сложных агентов и её приложения // *Онтология проектирования*. 2015. Том 5. №3(17). С. 278-296.

[Сахаров, 2012] Сахаров Д.А. Биологический субстрат генерации поведенческих актов // *Журнал общей биологии*. Том 73, № 5, Сентябрь-октябрь, 2012. С. 334–348.

A NEW APPROACH TO MODELING OF ARTIFICIAL NEURONS AND THEIR INTERACTION

Zhilyakova L. Yu.

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

zhilyakova.ludmila@gmail.com

An attempt to enrich the apparatus of artificial neural networks adding the extra properties to the formal neuron model is undertaken. We consider not only the electrical but also its chemical properties: sensitivity to different neurotransmitters and the ability to produce them. A network of complex neurons is a heterogeneous structure, in which connections depend on the chemical situation in the intercellular space, which is also part of the model. In such a network transmitters can play a crucial role in shaping patterns of activity.

Keywords

Heterogeneous neural network; pattern of activity; heterochemical ensemble; formal neuron model.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.92

РЕАЛИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВИРТУАЛЬНЫХ СРЕД ПО ДЕКЛАРАТИВНОЙ МОДЕЛИ

Грибова В.В., Федорищев Л.А.

*Институт автоматики и процессов управления Дальневосточного отделения
Российской академии наук, г. Владивосток, Россия*

gribova@iacp.dvo.ru

fleo1987@mail.ru

В работе приводится описание подхода к разработке пользовательских интерфейсов виртуальных сред по декларативной модели. Данный подход позволяет заменить кодирование в разработке и сопровождении не только самих виртуальных сред, но и интерфейсов их естественным декларативным описанием, что открывает дополнительные возможности по гибкой настройке и управлению пользовательских интерфейсов.

Ключевые слова: интерфейсы, виртуальные среды, онтологии, облачные технологии.

Введение

В настоящее время при создании программного обеспечения большое внимание уделяется его жизнеспособности или максимально возможному увеличению жизненного цикла. Известны данные [Ogheneovo E.E., 2013], что в процессе жизненного цикла программного средства около 30% всех усилий уходит на стадию разработки и более 70% на стадию сопровождения в связи с изменениями требований пользователей, условий эксплуатации, предметной области, ошибками и дефектами в программном обеспечении

Разработка профессиональных виртуальных сред (ПВС) также связана с указанными трудностями [Трухин, 2008]. Существующие средства ориентированы на программистов, а не на экспертов предметных областей. При этом основу виртуальных сред составляют предметные знания, которые используются и реализуются в рассмотренных средствах неэффективно (встроены в код программного средства).

В лаборатории интеллектуальных систем ИАПУ ДВО РАН реализован комплекс Интернет Разработки Виртуальных Интерактивных Сред – ИРВИС [Gribova et al., 2014], в котором авторами предлагается принципиально другой подход к созданию и сопровождению виртуальных сред: разработка декларативных моделей (с привлечением экспертов предметной области) вместо прямого кодирования. Кроме того, данный комплекс работает на облачной платформе IASaaS [Клещев

и др., 2011], что обеспечивает значительное упрощение коллективной разработки и сопровождения виртуальных сред, т.к. все сервисы и инструменты находятся «в облаках».

Одна из проблем, возникших при разработке комплекса ИРВИС, заключается в сложности формирования произвольных пользовательских интерфейсов для различных сервисов виртуальной реальности. Предметные области, для которых создаются виртуальные среды, имеют различные задачи, а, следовательно, и различающийся интерфейс. В качестве возможного решения этой проблемы можно было бы предусмотреть множество заранее настроенных компонентов интерфейса для различных задач. Однако такой подход в создании интерфейса является достаточно трудоемким, т.к., например, включение-выключение нужных компонентов пришлось бы выполнять каждому пользователю. Другая проблема, возникающая при работе с интерфейсами, заключается в том, что в некоторых задачах необходимо добавлять некоторую специфическую программную логику для компонентов интерфейса.

Целью работы является описание метода создания интерфейсов виртуальных сред на основе декларативной модели и реализация данного подхода.

1. Онтологический подход к созданию и управлению виртуальных сред

Основная идея создания интерфейсов в комплексе ИРВИС заключается в использовании той

же технологии, что и для создания самих виртуальных сред: технологии декларативных моделей на основе семантических представлений.

Основные принципы подхода к генерации виртуальных сред на основе семантических представлений были рассмотрены в [Грибова и др., 2012; Грибова и др. 2015 (OSTIS)]: Логическое представление ПВС является ключевым описанием, по которому, с одной стороны, формируется визуальное представление виртуальной среды, с другой стороны, оно определяет логику ее работы. Логическое представление включает описание объектов виртуальной среды и их свойств (атрибутов), возможные действия пользователя в виртуальной среде и результаты, к которым могут привести эти действия. Вся информация, составляющая логическое представление модели ПВС, является знаниями экспертов предметной области, соответственно эксперты должны не только формировать их, но также и сопровождать в процессе жизненного цикла ПВС.

2. Онтология интерфейса

Принципиально новым расширением в разработке и управлении ПВС стала возможность декларативного формирования WIMP-интерфейса. На рис. 1 показано место декларативной модели интерфейса в общей системе программного комплекса, построенного на основе онтологического подхода.

Анализ виртуальных сред показал, что независимо от предметной области и решаемых задач, в которых используются виртуальные среды, их все можно разделить на несколько классов, каждый из которых имеет фиксированный основной функционал, и, возможно, некоторый

дополнительный. Набор фиксированных функций определен в интерпретаторе, поэтому не требует дополнительного программирования и обработки. Однако, в случае включения дополнительных функций, программист может их добавить и, если необходимо, описать соответствующий WIMP-интерфейс.

Структура описаний стандартных и расширенных задач в целом совпадают, принципиальная разница заключается в том, что решение стандартных задач поддерживаются интерпретатором модели ПВС; для включения в ПВС дополнительных или расширенных задач необходимо описать структуру интерфейса этой задачи и добавить агент, который обеспечивает ее решение (обработку).

Каждая задача может быть разделена на подзадачи. Подзадачи предназначены для группировки элементов интерфейса в семантически связанные группы, которые имеют один агент обработки для расширенных задач, либо связаны с решением стандартной подзадачи, реализованной в интерпретаторе. С каждой задачей может быть связано произвольное множество возможных фрагментов интерфейса (различных представлений), из которых разработчик может выбрать представление, удобное для конкретной реализации, либо описать это представление самостоятельно.

Каждый тип ПВС имеет свой обязательный или номинальный набор функций, необходимый для формирования данного типа ПВС. Онтология WIMP-интерфейса ПВС описывает декларативное определение компонентов разных типов ПВС, которые могут быть с успехом применены к различным виртуальным средам одновременно.

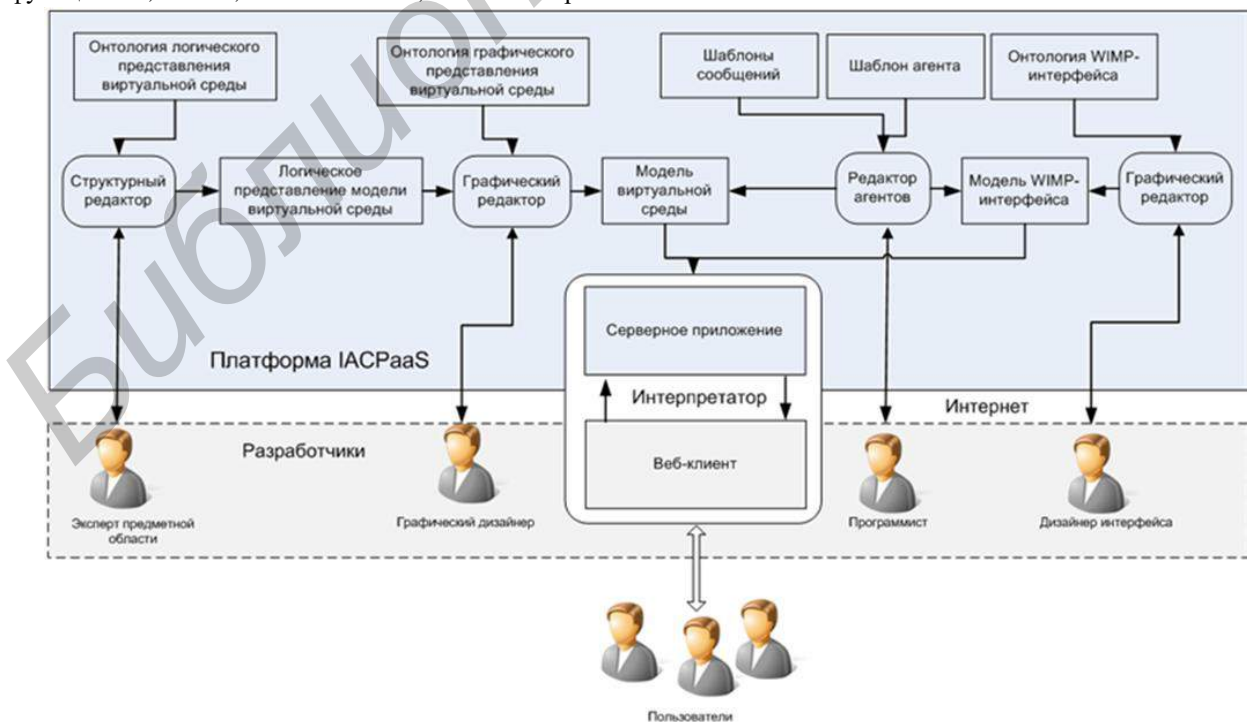


Рисунок 1 -- Модель программного комплекса

Онтология WIMP-интерфейса имеет следующий вид (см. рис. 2):

WIMP-интерфейс = *<Настройки, Стандартные компоненты, Пользовательские компоненты>*

Общие настройки = *<Настройки сервиса, Настройки виртуальной среды>*.

Настройки сервиса ∈ *<Ширина сервиса, Высота сервиса>*.

Настройки виртуальной среды = *<Ширина окна виртуальной среды, Высота окна виртуальной среды>*.

Стандартные компоненты = { *Объект VR_p* ∈ *Стандартный элемент* }ⁿ, n = количество стандартных элементов;

Пользовательские компоненты = { *Компонент_p* ∈ *Пользовательский компонент* }ⁿ, n = количество пользовательских компонентов;

Стандартный элемент ∈ { *Меню, Управление объектами среды, Библиотека, Свойства объекта среды, Координаты объекта, Тест* }.

Пользовательский компонент ∈ { *Текст, Кнопка, Список, Выпадающий список* }.

Кнопка = *<Текст, Координаты, Внешнее действие, Внутреннее действие, Ответное внутреннее действие>*.

Внешнее действие – указывает на внешний агент и описанную в нем функцию реализации необходимого действия, которое должно выполняться в результате нажатия на кнопку.

Внутреннее действие ∈ { *Функция АПИ* }_i где i < количество функций виртуальной среды, доступной для использования в интерфейсе пользователя.

Ответное внутреннее действие отличается от обычного внутреннего действия тем, что выполняется не после нажатия на интерфейсный элемент (кнопку), а после завершения внешнего действия и возвращения соответствующего результата от внешнего агента на клиент виртуальной среды.

Ответное внутреннее действие ∈ { *Функция АПИ* }_i где i < количество функций виртуальной среды, доступной для использования в интерфейсе пользователя

Список = *<Текст, Координаты, Внешнее действие, Внутреннее действие>*

Для описания интерфейса дизайнер интерфейса, используя графический редактор, формирует необходимый интерфейс. В основном проектирование интерфейса сводится к выбору представления интерфейса из множества предлагаемых вариантов. При этом библиотека представлений интерфейса является расширяемой.

Если разработчику необходимо свое уникальное представление, то, задавая тип задачи, решаемой с помощью виртуальной среды, он может спроектировать свое уникальное представление, выбрав интерфейсные элементы и определив их атрибуты.



Рисунок 2 - Онтология интерфейса

3. Метод реализации

Реализация программного средства разработки интерфейсов виртуальных сред с использованием онтологического подхода выполняется в виде отдельного программного сервиса-редактора на платформе IACPaaS. Данный редактор имеет графический WIMP-интерфейс с визуальным представлением всех элементов интерфейса, заданных в онтологии.

Реализовывается данный редактор как клиент-серверное приложение (аналогично самим виртуальным средам и другим сервисам платформы IACPaaS) на основе мультиагентной технологии на серверной стороне и с веб-клиентом на клиентской стороне.

Использование мультиагентного подхода совместно с семантическим представлением информации на всей платформе IACPaaS позволяет легко формировать, менять, загружать, использовать любую декларативную информацию, хранимую в семантическом представлении, в том числе, декларативные модели интерфейсов ПВС. Это означает, что единовременно созданная декларативная модель интерфейса может с успехом использоваться, обрабатываться и меняться во множестве других сервисов платформы, включая сервисы самих виртуальных сред (редактор ПВС и интерпретатор ПВС).

На клиентской стороне (в веб-клиенте сервиса) интерфейс создается визуально по технологии WYSIWYG (what you see is what you get - что видишь, то и получишь) (см. рис. 3). При этом все данные об изменениях интерфейса отправляются на

серверный скрипт (агент-контроллер), который формирует декларативное представление модели данного графического интерфейса. Механизм Drag&Drop позволяет перетаскивать и размещать элементы в любое место интерфейса.

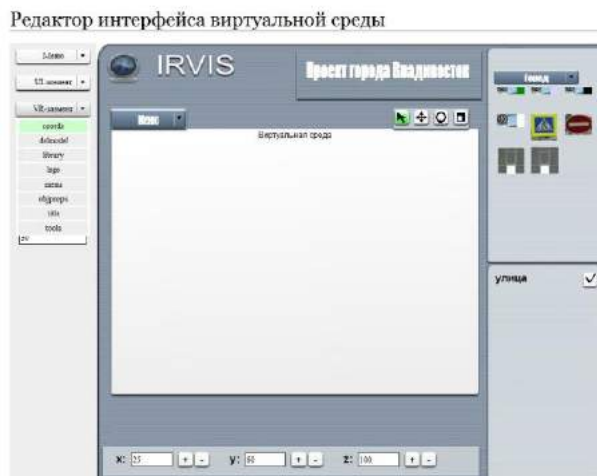


Рисунок 3 - Редактор интерфейса ПВС

4. Применение интерфейсов

На базе комплекса ИРВИС разработаны и находятся в разработке различные виртуальные среды [Грибова и др., 2014]: обучающий тренажер по классическим методам исследования зрения в офтальмологии, обучающий тренажер по неврологии, виртуальная химическая лаборатория, виртуальные среды по моделированию виртуального города, по созданию модели транспортных потоков. Приведенные примеры показывают разнообразные задачи, которые, соответственно, требуют разных интерфейсов: 1) в обучающих тренажерах требуется интерфейс тестирования и обучения, 2) в модели виртуального города требуются способы управления объектами города и изменения их с учетом информации, полученной из внешнего источника (по отношению к сервису ПВС), 3) в моделях транспортных потоков требуются интерфейсные средства управления визуализации потоков (например, в виде линий или графиков).

Заключение

В данной статье описаны принципы декларативного подхода для разработки интерфейсов такого класса интеллектуальных систем как виртуальные среды. Приводится онтология интерфейсных компонентов виртуальных сред. Описывается метод реализации пользовательских интерфейсов виртуальных сред по декларативным моделям.

Работа выполнена при финансовой поддержке РФФИ, проект № 15-07-03193 и ДВО РАН проект «Дальний Восток».

Библиографический список

[Gribova et al., 2014] Gribova V, Fedorischev L. Cloud service for development of virtual interactive environments // Proc. of 2014 2nd International Conference on Advanced ICT for Education (ICAICTE 2014), August 16-17, 2014 in Dalian, China. Atlantis Press, 2014. Vol. 1. Pp. 20-23

[Ogheneovo E.E., 2013] Ogheneovo E.E. Software Maintenance and Evolution: The Implication for Software Development // Journal of Industrial and Academic Research . – 2013. –Vol.7 No. 1

[Грибова и др., 2012] Грибова В.В., Федорищев Л.А. Обучающие виртуальные системы и средства их создания // Вестник информационных и компьютерных технологий. – 2012. – №3. – С. 48-51

[Грибова и др., 2014] Грибова В.В., Федорищев Л.А. Создание интеллектуальных сервисов с виртуальной реальностью // Материалы 14-й национальной конференции по искусственному интеллекту с международным участием (КИИ 2014). – 24-27 сентября 2014 Казань, Россия.

[Грибова и др., 2015] Разработка и генерация виртуальных сред на основе семантических представлений // Материалы V международной научно-технической конференции “Открытые семантические технологии проектирования интеллектуальных систем” (OSTIS-2015), Минск, 2015. С. 183-187

[Клещев и др., 2011] Клещев, А.С. и другие Облачная платформа для разработки и управления интеллектуальными системами // Труды конференции OSTIS-2011. С. 5-14.

[Трухин, 2008] Трухин, А.В. Анализ существующих в РФ тренажерно-обучающих систем // Открытое и дистанционное образование. - 2008. - №1. - С. 32-39

REALIZATION OF USER'S INTERFACES OF VIRTUAL ENVIRONMENTS BY DECLARATIVE MODEL

Gribova V.V. , Fedorischev L.A

Institute of Automation and Control Processes, the Far Eastern Branch of Russian Academy of Science Vladivostok, Russia

gribova@iacp.dvo.ru

fleo1987@mail.ru

The article presents an approach to development of user's interface for virtual environments by declarative model. The approach is to change coding of a program by declarative description of not only the program but it's interface too. It opens additional possibilities on management of users interfaces.

Introduction

At present virtual environments is perspective direction of development. However, development and maintenance of virtual interactive environments and it's interfaces are still a complicated process. We developed the new program complex IRVIS on the cloud platform IACPaaS. At this paper we want to show a new approach to generate user's interfaces for virtual environments that are developed by IRVIS.

Conclusion

Principles of applying of semantic nets to interfaces of virtual environments are described at the article. There are presented the method of development, formalized semantic nets and an example of interfaces of virtual environment.



УДК 004.8:165.1

ЛОГИКО-АЛГЕБРАИЧЕСКИЕ МЕТОДЫ ПОСТРОЕНИЯ КОГНИТИВНЫХ СЕНСОРОВ

Святкина М.Н., Тарасов В.Б.

*ФГБОУ ВПО «Московский государственный технический университет им. Н.Э.Баумана»
г. Москва, Россия*

maria.svyatkina@gmail.com

Vbulbov@yahoo.com

Основная идея настоящей статьи заключается в демонстрации возможностей и развитии открытых прагматических технологий проектирования интеллектуальных систем. Исследована комплексная проблема мониторинга состояния объектов железнодорожной инфраструктуры. Отмечено, что центральным компонентом интеллектуальной системы мониторинга является гибридная систем приобретения/ обнаружения знаний, включающая модули формирования знаний на основе измерений, извлечения знаний из эксперта и онтологического инжиниринга. Показана ключевая роль измерений в решении задач мониторинга. Рассмотрены особенности измерений как познавательного процесса, находящегося в тесном единстве с оценками и рассуждениями. Описана предложенная ранее авторами концепция когнитивных измерений как иерархического процесса грануляции информации, осуществляемого с помощью когнитивных сенсоров. Когнитивный сенсор представляет собой не только информационно-измерительное устройство, познающее объект мониторинга и обеспечивающее получение знаний на базе измерений, но и «понимающую» искусственную систему, способную формировать оценки результатов измерений и проводить рассуждения в интересах текущей диагностики объекта и прогнозирования его будущего состояния. Функционирование когнитивного сенсора опирается на логическую прагматику измерений. В этом контексте проведено обсуждение ряда определений, концепций и формальных моделей понимания. Предложен общий логико-алгебраический подход к построению моделей понимания на базе логических (истинностных) оценок и норм. В рамках этого подхода различные логические системы представляются в виде алгебраических структур, в частности, цепей, решеток, полурешеток и произведений решеток, причем эти структуры порождаются с помощью различных отношений порядка. Центральное место в работе занимает формальное описание и графическая иллюстрация понятия логического мира. Для этого введены наглядные представления прагматики логических значений, опирающиеся на «метафору цвета» и «метафору геометрического узора». Эти представления, реализующие подходы иллюстративной и когнитивной графики в многозначных логиках, лежат в основе конкретных когнитивных сенсоров – сенсора Васильева, сенсора Клини и сенсора Белнапа, а также когнитивных сенсорных сетей, основанных на бирешетках и мультирешетках. В результате разработана методика логико-алгебраического синтеза когнитивных сенсоров и представлена блок-схема соответствующего алгоритма.

Ключевые слова: искусственный интеллект, интеллектуальная система мониторинга, приобретение знаний, обнаружение знаний в сенсорных данных, многозначные логики, когнитивные измерения, прагматика, когнитивный сенсор, понимание, интерпретация, когнитивная графика.

Введение

Развитие современного высокоскоростного железнодорожного транспорта предъявляет новые, повышенные требования к мониторингу состояния железнодорожной инфраструктуры [Шабельников и др., 2010], в особенности, таких ее стратегических объектов как искусственные сооружения – мосты, тоннели, переезды и т.п. При этом проблема мониторинга является комплексной и включает в себя следующие задачи: измерение ключевых характеристик сложных технических объектов и их физико-технической среды, а также интерпретация

результатов измерений; анализ протекания процессов в наблюдаемых объектах и диагностика их текущих состояний; прогнозирование дальнейшего развития; принятие решения, связанного с управлением состоянием объекта железнодорожной инфраструктуры.

Здесь центральное место занимают различные измерения; так для мостов, это – измерения метеорологических показателей (в первую очередь, направление и скорость ветра), определение прогиба пролетных строений моста, измерение отклонений элементов конструкций моста от вертикальной оси под действием проходящих

составов и ветровых нагрузок, оценка колебаний конструкций моста под нагрузкой и пр.

Для эффективного решения последовательности задач мониторинга требуется создание передовых интеллектуальных систем, встроенных в реальную физико-техническую среду и имеющих гибридные подсистемы приобретения/обнаружения знаний.

Первое поколение систем приобретения знаний было связано с извлечением знаний из уникального эксперта (экстенсивная схема приобретения знаний) и наполнением «оболочек» экспертных систем. Второе поколение, предполагающее разработку системы онтологий и проведение онтологического инжиниринга, обеспечивает построение концептуальных моделей, разделяемых сообществом специалистов. Прежде всего, оно ориентировано на достижение взаимопонимания и осуществление совместной работы агентов в многоагентных системах. Наконец, в гибридных системах приобретения знаний 3-го поколения наряду с экспертными знаниями и онтологиями используются средства получения и анализа сенсорных данных [Святкина и др., 2014].

В целом, взаимосвязи между интеллектуальными технологиями и информационно-измерительными системами являются двусторонними. С одной стороны, сами измерения стали интеллектуальными, опирающимися на процессы и методы извлечения, представления, пополнения и использования метрологических знаний, создание как экспертных систем, так и специальных агентов измерений. С другой стороны, интеллектуальный анализ сенсорных данных и обнаружение знаний на базе измерений служит естественной основой для разработки искусственных когнитивных систем.

В настоящей работе рассмотрены теоретические основы *когнитивных измерений*, проводимых с помощью *когнитивных сенсоров*, и предложена логико-алгебраическая методика синтеза когнитивных сенсоров.

1. Когнитивные измерения и когнитивные сенсоры

1.1. Когнитивные процессы в интеллектуальных системах

Одна из главных особенностей современного развития прикладных интеллектуальных систем заключается в расширении базовой архитектуры, связанное с реализацией широкого спектра когнитивных процессов – восприятия, обучения, представления, мышления, понимания. До недавних пор в теории интеллектуальных систем исходили из того, что знания должны быть истинными, а их обработка – сохраняющей истинность. При этом процедуры корректировки и пополнения знаний были направлены на то, чтобы «общий уровень истинности» хранимых знаний монотонно возрастал.

Однако, в когнитивных системах знания могут быть основаны не на истине, а само понятие истинности претерпевает существенные изменения. По мнению О.П.Кузнецова, истинность – это лишь один из видов ценности знания, способствующий его устойчивости [Кузнецов, 2008]. Обыденное рассуждение основано скорее не на истине, как ее понимают в классической логике, а на хороших (в смысле гештальт-психологии) структурах.

Восходящие к Дж.Данну информационные трактовки истинностных значений как объектов, выражающих *размытость*, *неопределенность* или *противоречивость* понятий или как значений, передающих *информацию* о предложении; допускают их *градуальность*, *грануляцию* и *немонотонные преобразования*

Отличительные характеристики познавательных процессов, которые следует учитывать при разработке интеллектуальных систем мониторинга, таковы [Тарасов и др., 2012б]: 1) познание представляет собой открытую систему, которая базируется как на имеющемся знании, так и на оперативно получаемых сенсорных данных; 2) познание порождает гипотезы, а не «готовые выводы»; эти гипотезы нуждаются в подтверждении или опровержении; 3) познание неотделимо от грануляции данных – формирования «зерен» информации и знаний различного размера.

1.2. Измерение как познавательный процесс

Теоретические и практические проблемы измерений являются частью *метрологии* – науки об измерениях, методах и средствах обеспечения их единства и способах достижения требуемой точности. Под измерением в классическом смысле понимается нахождение значения измеряемой величины опытным путем с помощью специальных технических средств. Любое измерение предполагает определение измеряемой величины, метода измерения и соответствующей процедуры.

Данное определение многократно подвергалось обоснованной критике. В частности, оно не отражает связи измерения с моделью объекта измерения; поэтому открытой остается интерпретация результата измерения. Более того, измерение должно рассматриваться более широко как познавательный процесс, в результате которого получают описание объекта в количественных терминах [Розенберг, 1975].

Рассмотрение измерения как познавательного процесса имеет давние исторические корни. Одно из наиболее глубоких определений «измерения», дано русским философом и ученым П.А. Флоренски в «Технической энциклопедии»: «...*измерение* – основной познавательный процесс науки и техники, посредством которого неизвестная величина количественно сравнивается с другою, однородной с нею и считаемою известной».

На практике этот процесс находится в неразрывном единстве с другими познавательными процессами, прежде всего, с вычислениями и

рассуждениями. Например, в задаче мониторинга сложного объекта измерение не является самоцелью, а служит для диагностики текущего состояния этого объекта, прогнозирования вариантов ее развития и поддержки принятия решений.

В этом контексте особый интерес представляют варианты рассмотрения измерений как базовой когнитивной техники (гносеотехники по [Кнорринг, 1992]), основного источника знаний и рассуждений для интеллектуальных систем новых поколений [Тарасов и др., 2013].

В упомянутой выше книге [Розенберг, 1975], измерение определяется как эксперимент, имеющий целью формирование истинностных суждений об исследуемом объекте. Отсюда следует, что в процессе измерения важнейшее значение имеет этап интерпретации и анализа его результатов, на котором целесообразно использование логических подходов и оценок.

Таким образом, когнитивные аспекты измерений тесно связаны с операциями сопоставления, классификации, интерпретации, понимания. Когнитивные структуры измерений позволят реализовать автоматизированные переходы «данные-информация-знания-метазнание» [Тарасов, 2015], что обеспечит единство процессов получения исходных данных в первичных измерениях, выделения полезной информации и обнаружения знаний в сенсорных данных, формирования оценок, мнений и норм, проведения диагностических или прогностических рассуждений в интеллектуальных системах мониторинга.

1.3. Когнитивные измерения и прагматическая грануляция информации

Сам термин «когнитивные измерения» был предложен С.В. Прокопчиной в ее статье [Прокопчина, 2010], где речь идет об измерениях, в результате которых с помощью байесовских интеллектуальных технологий извлекаются метрологически аттестованные знания.

В наших работах [Тарасов и др., 2013; Святкина и др., 2014] концепция когнитивных измерений связана с грануляцией измерительной информации на основе принципа единства измерений, оценок и рассуждений, принципа формирования наглядной логико-алгебраической прагматики измерений и принципа синтеза теорий истины при интерпретации результатов измерений.

Согласно принципу *единства измерений, оценок и рассуждений*, измерение понимается как когнитивный процесс, связанный с построением гранулярной структуры информации в виде двухуровневой иерархии (рисунок 1), где на нижнем уровне с помощью набора датчиков, проводятся обычные измерения, результаты которых составляют *мелкозернистую* информацию, а на верхнем уровне полученные результаты отображаются на прагматическую шкалу

укрупненных оценок – аксиологических или деонтических – выражающих *крупнозернистую* информацию (полученные значения измеряемого параметра находятся «в норме», «почти в норме», «не в норме», и т.п., например, измеренное значение скорости ветра – «почти в норме»).

Итак, прагматическая шкала оценок содержит относительно небольшой набор аксиологических значений (типа «хорошо», «посредственно», «плохо» и др.) или деонтических значений («обязательно», «разрешено», «запрещено»), достаточных для диагностики, прогнозирования или принятия решений с допустимым уровнем точности. Главная цель когнитивных измерений в процессе мониторинга заключается не столько в получении точной числовой информации, сколько в выработке полезных практических рекомендаций с помощью рассуждений.

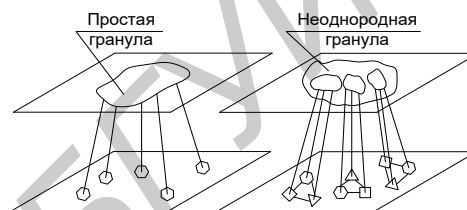


Рисунок 1 – К формированию простых (однородных) и составных (неоднородных) гранул

Следует отметить, что традиционные измерения, направленные только на работу с данными, явно носят машиноцентрический характер, тогда как когнитивные измерения, будучи ориентированными на преобразование данных в знания, являются антропоцентрическими.

Проект OSTIS [Голенков и др., 2013] нацелен на создание открытых семантических технологий проектирования интеллектуальных систем. В данной статье предлагается дополнить его содержание *открытыми прагматическими технологиями* проектирования интеллектуальных систем, восходящими к идеям «Отца Прагматизма» Ч.Пирса [Пирс, 2000]. В этом плане возникает необходимость не только систематизации ранних идей, но и получения новых результатов в таких областях как:

- теоретические основы формализации и представления прагматики (прагматические миры и пространства);
- технологии синтеза логических систем, в основе которых лежит формализация прагматики;
- формализованные модели мнений, оценок и других базовых прагматических категорий;
- архитектура и технология проектирования интеллектуальных систем, в основе которых лежат прагматические модели (в частности, и логические прагматики, и прагматические логики).

В семиотике семантика есть выражение отношений между сообщением и его автором (отправителем), а прагматика характеризует отношения между сообщением (символом) и его пользователем (получателем). С помощью правил

прагматики среди множества смысловых значений, приписываемых символу, выбирают определенное смысловое значение – основное в конкретный момент времени [Попов, 2004]. Прагматику часто рассматривают как *исследование значения* в некотором *контексте*. Иными словами, прагматика учитывает конкретного адресата (интерпретатора), а у семантики его нет.

Логическая семантика – это раздел металогики, в котором изучаются интерпретации логических исчислений. Она занимается изучением смысла и значений конструкций формализованного языка теории, способами понимания его логических связей и формул [Смирнова, 1986]. Современная семантика включает два основных направления: теория истины и теория значения. Основоположник логической семантики Г.Фреге даже определил логику как науку о наиболее общих законах бытия *истины* [Фреге, 2012], а Я.Лукасевиц, развивая это определение, указал, что логика есть наука об особом роде объектах, называемых *логическими значениями* [Лукасевиц, 1993]. Ниже главное внимание уделяется описанию и визуализации таких логических значений, как «истина», «ложь», «неопределенность», «противоречивость», и др.

Логическая прагматика измерений связана с пониманием и полезностью результатов измерений для решения задач диагностики, прогнозирования, и в целом, мониторинга. Ее можно определить как раздел металогики, изучающий, каким образом интерпретатор на основе информации, не заложенной в результатах измерений, выбирает из множества интерпретаций наиболее подходящую для данного случая. Базовыми *прагматическими структурами* являются *мнение*, выражающее отношение между агентом и суждением (по А.Черчу), и *оценка*.

Принцип формирования *наглядной логико-алгебраической прагматики* измерений связан с погружением результата измерения в некоторый логический мир, который задается с помощью алгебраической структуры (решетки, полурешетки, бирешетки, и пр.), визуализированной с помощью диаграммы Хассе. В реализации этого принципа должны быть задействованы как *гранулярные логические прагматики*, опирающиеся на обобщенные значения истинности, так и *прагматические логики*, т.е. логические системы, основанные на прагматических (ценностных) трактовках истины [Ajdukiewicz, 1974; Будбаева и др., 1974, Ивин, 2016].

Наконец, логико-алгебраическая обработка результатов измерений с помощью истинностных значений и деонтических оценок предполагает синтез различных теорий истины. В современной логике известны три основные теории истины [Финн, 2004]: а) *теория соответствия* Аристотеля-Тарского, по которой истина есть соответствие логического предложения фактам внешнего мира, причем первичными являются именно реальные факты; б) *теория когерентности*, согласно которой

значения истинности высказываний основаны не на соответствии реальности, а на имеющихся знаниях, (высказывание считается истинным, если оно согласуется с остальной частью знания, т.е. связность, непротиворечивость и устойчивость знаний принимаются как основные характеристики истины); в) *прагматическая теория* истины, где истинность отождествляется с *полезностью* знаний при решении конкретных задач. Здесь первичными являются наши знания о мире, воплощенные в виде формальных или неформальных систем предписаний и норм (примерами служат различные своды законов, от библейских заповедей до ГОСТ Р 54500.3-2011 – Неопределенность измерений. Руководство по выражению неопределенности измерений), а истина характеризует их полезность и применимость в жизни определенного сообщества.

Помимо этих классических теорий следует рассмотреть и ряд других, полезных для построения логических моделей понимания интерпретаций истины, предложенных в XX-м веке. Так А.Пуанкаре [Пуанкаре, 1990] трактует истину как результат соглашения, договоренности; его подход получил название *«конвенциональная теория истины»* и представляет особый интерес для логического моделирования процессов взаимопонимания.

В различных приложениях логических моделей могут появляться другие нестандартные трактовки истины. Так в логике аргументации В.К. Финна (см. [Финн, 2011]) введено понятие *«фактической истины»* как предельного случая ситуации аргументации, задаваемого непустым множеством аргументов и пустым множеством контраргументов. В логических моделях переговоров [Тарасов, 2009] определена *«согласованная истина»* как результат соглашения между агентами, причем ее отрицание есть неподвижная точка.

В контексте когнитивных измерений на первый план выходят классическая и прагматическая трактовки истины: первая приобретает вид соответствия между числовыми результатами измерений и их качественными оценками («измеренная истина»), а вторая связана с полезностью результата измерения для решения некоторой комплексной проблемы, например, задачи мониторинга или управления.

1.4. Когнитивные сенсоры и модели понимания

Инструментальными средствами реализации когнитивных измерений выступают когнитивные сенсоры – датчики, способные не только измерять значения некоторого параметра рассматриваемого объекта, но и «понимать» полученную информацию, а также проводить рассуждения в интересах формирования рекомендаций, способствующих решению задач мониторинга [Тарасов и Святкина, 2012]. *Когнитивный сенсор* представляет собой не только информационно-измерительное, но и интерпретирующее устройство, снабженное логико-лингвистической прагматикой, т.е. способное

выражать результат измерения в терминах ограниченного естественного языка. При этом происходит грануляция информации: исходная мелкозернистая информация сенсорного уровня превращается в обобщенную крупнозернистую информацию речемыслительного (логико-лингвистического) уровня [Тарасов, 2013]. В общем случае когнитивные сенсоры строятся по схемам, представленным на рисунке 2.

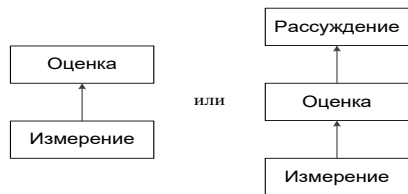


Рисунок 2 – Схемы построения когнитивных сенсоров

Ниже приведем примеры оценок и рассуждений, основанных на измерениях. Например, результат измерения скорости ветра на мосту с помощью анемометра равен 25-26 м/с. Тогда когнитивный анемометр диагностирует шторм и выдает деонтическую оценку «движение по мосту запрещено». Другой вариант: измеряются два параметра – скорость ветра анемометром и величина деформация конструкции моста тензометром. Результатом интерпретации полученных числовых значений являются нечеткие лингвистические оценки «Скорость ветра – довольно большая» и «Величина деформации» – «близкая к предельно допустимой». Тогда можно построить нечеткое продукционное правило типа «ЕСЛИ скорость ветра – довольно большая И величина деформации моста – близкая к предельно допустимой, ТО ввести ограничения на массу движущихся по мосту поездов в 3500 т».

Поскольку когнитивный сенсор трактуется здесь не только как инструментальное средство познания объекта мониторинга и получения знаний на основе измерений, но и как понимающая искусственная система, рассмотрим ниже основные определения, концепции и модели понимания.

До недавних пор весьма распространенной оставалась точка зрения, что пониматься может только текст, наделенный определенным смыслом: понять его – значит раскрыть смысл, вложенный в текст его автором. Однако объектами понимания могут быть не только тексты, но и сообщения, события, поведение, ситуации, результаты измерений и пр. Понимание может рассматриваться и как способность, и как операция, и как процесс, и как результат, и как средство (технология) усвоения знаний.

Сам термин «понимание» обычно раскрывается с помощью категорий «смысл», «значение», «истина», «ценность», «норма», например, понять мотивы поведения, понять предпосылки возникновения ситуации, установить ценность сведений, найти истину, и пр. Его часто уточняют с помощью дихотомий, например, «познание-понимание» (сама возможность познания во-многом определяется достигнутым уровнем понимания, но

понимание, в свою очередь, сильно зависит от исходных знаний, когнитивных способностей и структуры языка), «обучение-понимание» (цель обучения – понимание), «объяснение-понимание» (понимание – способность что-либо объяснить), «понимание-интерпретация» (прямой очевидный смысл связан с пониманием, тогда как интерпретация подразумевает выявление скрытого смысла или порождение нового смысла; таким образом, понимание является необходимым условием всякой интерпретации).

Формализации постановки и решения проблемы понимания уделяли значительное внимание еще классики искусственного интеллекта Т.Виноград, [Winograd et al., 1986], А.Ньюэлл, Г.Саймон, Э.В.Попов [2004], Д.А.Поспелов [1989], и др. Так в работе [Moore and Newell, 1973] было введено следующее определение понимания: система S понимает знание K , если она использует его всякий раз, когда K уместно. С одной стороны, здесь понимание относится к знаниям, а не к реальным предметам. С другой стороны, понимание связано с использованием этих знаний для решения задач; соответственно глубину понимания можно охарактеризовать тем классом задач, которые способна решать система. В этой связи Г.Саймон [Simon, 1977] предложил задачно-ориентированное определение понимания: система S понимает задачу T , если она имеет знания и процедуры K , необходимые для выполнения T . Таким образом, понимание выражается тернарным отношением, связывающим S , T и K .

Согласно Э.В.Попову, понимание собеседника означает определение тех целей (намерений), которые тот преследовал, формируя некоторый текст [Попов, 2004]. В этом плане «понять» смысл сообщения означает найти его взаимосвязь с целью общения. Если это взаимосвязь не установлена, то следует говорить о непонимании сообщения.

По [Карнап, 2007], основным средством прагматики, с помощью которого можно уточнить идею «понимания», являются «предложения о мнениях», т.е. высказывания вида $BEL(a, t, p)$ – агент a в момент времени t считает, что p , где p – предложение естественного языка. Следует отметить, что Карнап использовал двузначную логику при рассмотрении предложений о мнениях, игнорируя варианты неопределенности, неполноты информации, противоречия.

Воспользуемся аналогичной конструкцией для определения прагматики измерений. Пусть x – некоторый сенсор, $x \in X$, где X – множество сенсоров; res – результат измерения; sit – ситуация измерения, $sit \in SIT$, где SIT – множество ситуаций. Тогда имеем следующие предикаты: $UND(x, res)$ – сенсор x понимает результат измерения res . $UND(x, res, sit)$ – сенсор x понимает результат измерения res в ситуации sit . В общем случае, имеем градации понимания, т.е. предикаты понимания являются нечеткими.

Подробный анализ самой междисциплинарной проблемы понимания и основных концепций понимания, разработанных в различных научных дисциплинах, проведен в работе [Тарасов, 2015]. Далее воспользуемся *аксиологической* трактовкой понимания, восходящей к классику герменевтики и основоположнику «психологии понимания» В. Дильтею. В ее русле *понимание* – это универсальная операция мышления, которая является *оценкой объекта* с определенных позиций, на основе некоторого *образца, стандарта, нормы, принципа* и т.п. [Философия, 2004]. По сути, это определение отражает общую идею понимания как сравнения, сопоставления с образцом. Оно опирается на *теорию ценностей*, которая изучает вопросы, связанные с природой ценностей, их местом в реальной жизни, соотношениями между ними.

Оценка есть способ установления ценности (или вывод из принятых ценностей с использованием общих правил) [Ивин, 2016], а главными ценностями деятельности являются ее результаты. При этом именно *потребности* и *ценности* определяют мотивы любой деятельности и ее направленность на достижение определенных целей.

Частным случаем оценок являются *нормы*, которые можно рассматривать как общественно апробированные и закрепленные оценки. Нормы и стандарты являются основой для выработки *требований* и *предписаний*; они обычно формализуются в виде *деонтических модальностей* «обязательно», «разрешено», «запрещено». Также предписания выражаются с помощью глаголов «может» и «должен».

По М.М. Бахтину [Бахтин, 1979], безоценочное понимание невозможно, поскольку нельзя разделить понимание и оценку: они одновременны и составляют единый целостный акт. Двумя базовыми операциями, делающими понимание возможным, являются поиск и представление стандарта оценки (нормы, образца) и обоснование его приложения в конкретной ситуации.

Следуя Ч.С. Пирсу в представлении логики как *нормативной науки* и рассмотрении истины как «логического блага», получаем прагматическое выражение истины как ценности (полезности), связанной с достижением нужного результата [Пирс, 2000]. Таким образом, и истинностные значения, и модальные оценки можно использовать в качестве логических значений для интерпретации результатов измерений. При этом прагматизм, допуская сосуществование не одной, а множества разных логик, утверждает, что выбор между ними не произволен, а определяется различными факторами эффективности.

В целом, анализируемый механизм понимания имеет двойственную описательно-оценочную природу [Ивин, 2016]. В основе предпонимания измерений лежит укрупненное описание результата измерений с помощью логических (истинностных) значений. При этом пониматься может все, для чего

найден общий образец, а непонимание обычно обусловлено необоснованностью, неочевидностью или отсутствием такого образца.

В этом русле необходимо отметить, что образец как основа понимания принципиально отличается от примера. *Пример* говорит о том, что имеет место в действительности, а *образец* – о том, что должно быть. Примеры используются для поддержки описательных высказываний, а ссылки на образцы служат обоснованием предписаний и требований.

Для пояснения принципа работы когнитивного сенсора как «понимающего» датчика следует рассмотреть два мира – собственно мир измерений *W* и мир норм (стандартов) *M*. На рисунке 3 показано, что в когнитивном сенсоре, использующем логическую прагматику, должны быть воплощены, по крайней мере, две концепции истины: 1) истина как соответствие оценочных значений фактам, полученным в результате измерений; 2) истина как полезная норма (а ложь – как полезная антинорма), позволяющая определить выделенные (или антивыведенные) логические значения.

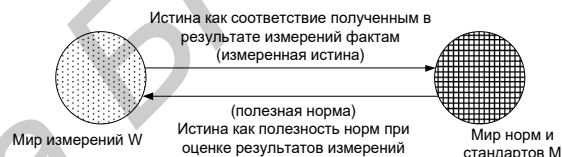


Рисунок 3 – Иллюстрации двух концепций истины для когнитивных сенсоров

В следующем разделе будет развит логико-алгебраический подход к построению моделей понимания на базе истинностных оценок и норм.

2. Многозначные логики и методы когнитивной графики в толковании результатов измерений

Многозначными логиками называются такие формальные логики, в которых допускается более двух значений истинности для высказываний. Здесь к обычным истинностным значениям «истина» и «ложь» добавляются и другие (промежуточные) значения. Примерами таких промежуточных значений являются «возможность» и «случайность» в трехзначной логике Я. Лукасевича, «половинчатая истина» в трехзначной логике А. Гейтинга, «бессмыслица» в трехзначной логике Д.А. Бочвара, «неопределенность» в трехзначной логике С.К. Клини, и пр. [Карпенко, 2010; Gottwald, 2001]. В свою очередь, нечеткие логики (в узком смысле) составляют подкласс многозначных логик с градуальными значениями истинности из интервала $[0, 1]$ и расширенными операциями конъюнкции и дизъюнкции, представленными в виде треугольных норм и конорм [Нажек, 1998; Новак и др., 2006].

В логике с давних пор применяют наглядные модели типа диаграмм. Здесь классическими примерами служат круги Эйлера и диаграммы Венна. Так круги Эйлера обеспечивают наглядное

толкование силлогизмов Аристотеля (например, знаменитого силлогизма *Modus Barbara*), а диаграммы Венна удобны для анализа логических высказываний. В алгебраической логике для наглядного представления семантических структур и реализации подхода «от семантики к исчислению» используют диаграммы Хассе. Еще одним характерным примером является графическое представление отношений логики Аллена.

Эти и подобные им методы имеют решающее значение на самых начальных этапах построения логических систем, когда речь идет об установлении некоторых главных принципов или аксиом, которые принимаются без доказательств и обосновываются лишь интуитивно.

Для иллюстрации перехода от классической логики к многозначным логикам очень полезной является «метафора цвета». Если классическая двузначная логика есть воплощение «черно-белого мира», то многозначная логика отражает столько «цветов радуги», сколько логических значений она содержит, а нечеткая логика демонстрирует непрерывный спектр цветов, включая все оттенки между зеленым и желтым, желтым и красным, пурпурным и розовым, розовым и голубым, голубым и синим.

В работе берется естественная «светофорная» интерпретация значений истинности для трехзначных и четырехзначных логических прагматик: 1) «истина» (значение параметра «в норме») – «зеленый цвет»; 2) «ложь» («не в норме») – «красный цвет»; 3) «противоречие» («и истина, и ложь одновременно» – промежуточная область между «в норме» и «не в норме») – «желтый цвет»; 4) «неопределенность («ни истина и ни ложь»)» – «синий цвет».

В свою очередь, все модальности можно классифицировать по двум критериям – знак и сила. Тогда основные модальные значения, в частности, деонтические и доксистические модальности, представлены следующими цветами: а) сильная положительная модальность («обязательно» или «уверенность») – «малахитовый (темно-зеленый) цвет»; б) слабая положительная модальность («разрешено» или «предположение») – «зеленый цвет»; в) слабая отрицательная модальность («необязательно», «безразлично» или «сомнение») – «розовый цвет»; г) сильная отрицательная модальность («запрещено» или «отвержение») – «красный цвет».

Здесь наглядно демонстрируются свойства операции отрицания в четырехзначной модальной системе, например, в доксистической системе $M_4 = \{O, P, B, Z\}$: так отрицание слабой положительной модальности есть сильная отрицательная модальность (например, $\neg P = Z$ – противоположность красного и зеленого цветов), а отрицание сильной положительной модальности – слабая отрицательная модальность (например, $\neg O = B$ – противопоставление темно-зеленого и розового цветов).

Другая удобная метафора для наглядного представления прагматики многозначных и модальных логик – это «метафора геометрического узора», построенного путём чередования в определенном порядке (ритме) точек, линий, повторяющихся элементов-мотивов. При узорчатой интерпретации логических значений будем использовать следующие приемы: 1) ритмический повтор косых (для значений истинности) или прямых (для модальностей) линий; 2) зеркальная симметрия и/или поворот (для логических значений, связанных между собой операцией отрицания, например для Т и F, Р и Z); 3) наложение нескольких элементов (создание сетки) для составных логических значений, т.е. подмножеств множества значений истинности; 4) отсутствие всякого узора для неопределенности, описываемой пустым множеством \emptyset значений истинности.

Графическое узорчатое представление значений истинности, в том числе, выделенных и антивыведенных значений, позволяет реализовать в логике идеи интерактивной (иллюстративной и когнитивной) графики А.А.Зенкина [Зенкин, 1991; Поспелов, 1992] и наглядно изобразить различные логические миры.

Общее изложение основ логико-алгебраического подхода к пониманию результатов измерений включает следующие этапы:

- анализ типов логических предложений (с акцентом на оценочные и нормативные предложения);
- выбор теории истины или описание варианта взаимодействия разных теорий истины;
- формирование множества логических (истинностных и ценностных) значений и представление их структуры с использованием когнитивной графики; спецификация выделенных значений;
- разработка системы логических миров и пространств;
- построение логических операций для выбранных логических миров;
- построение итоговой расширенной логической матрицы.

2.1. Типы логических предложений

Разработка логических моделей понимания требует определения и формального представления основных логических предложений – суждений, оценок, норм. Далее воспользуемся следующими основными определениями из [Ивин и др., 1998], [Ивин, 2016], [Борисов и др., 2006].

Суждением называется любое высказывание в форме повествовательного предложения, которое является истинным или ложным. Правдоподобные суждения могут связываться с промежуточными, градуальными значениями истинности. Они выражаются в виде $v(p)$, где $p = X \text{ is } A$ есть высказывание, характеризуемое значением истинности $v \in V$, например, $V = [0, 1]$ (X – переменная,

принимая свои значения в универсуме U , A – ограничение на X).

Согласно Л.Заде [Zadeh,1997], обобщенное ограничение можно определить в виде $p = X \text{ is } r \text{ A}$, где r – переменная связка, выражающая способ ограничения (например, равенство, неравенство, распределение вероятности, распределение возможности). В русле теории измерений чаще всего рассматривают вероятностные распределения.

Оценкой (оценочным суждением) называется высказывание, устанавливающее абсолютную или сравнительную ценность некоторого объекта. Оценки обычно включают следующие компоненты [Ивин, 2016]: 1) субъект (или агент) оценки, т.е. лицо или группа лиц, приписывающая ценность некоторому объекту; 2) предмет оценки, т.е. объекты, которым приписывают ценность или объекты, ценности которых сопоставляются; 3) характер оценки (например, абсолютная или сравнительная, положительная или отрицательная оценка); 4) основание оценки (явление или ситуация, в рамках которой производится оценивание). Таким образом, любую оценку можно представить в виде фрейма-прототипа: {Оценка, <Кто (субъект оценки),... >, <Что (Кого) (объект оценки),... >, <Какая (тип оценки),... >, <Почему (основание оценки),... >}

Оценки обычно выражаются с помощью *аксиологических* (оценочных) *модальностей*: как абсолютных («хорошо», «плохо», «безразлично»), так и относительных («лучше», «хуже», «равноценно»). По фон Вригту [Вригт фон, 1986], связь между абсолютными и относительными оценками может задаваться следующим образом: q – «хорошо», если q – «лучше, чем его отрицание $\neg q$ ».

Могут также привлекаться прагматические модальности: «полезно-вредно», «эффективно-неэффективно» и пр. Для правильного понимания оценок важную роль играет контекст, в котором они формулируются.

Средством, превращающим оценку в норму, является угроза наказания, т.е. стандартизация норм осуществляется с помощью санкций. Так еще К.Менгер установил прямую связь между предписанием $\Box p$ («обязательно p ») и «если не p , то наказание или ухудшение» (см. [Ивин,2016]).

Согласно принципу Юма, нельзя с помощью одной логики перейти от утверждений со связкой «есть» к утверждениям со связкой «должен». По этому принципу отвергается как выводимость оценок из чистых описаний, так и выводимость чистых описаний из оценок. Тем не менее, между описательными и оценочными высказываниями, несомненно, имеются связи. Так понимание как подведение под общую оценку (сильное понимание) есть дедуктивное умозаключение вида:

Всякое A должно быть B
 C есть A
Следовательно, C должно быть B

Например, всякий неисправный объект должен ремонтироваться.

Локомотив – неисправен.

Значит, локомотив должен ремонтироваться.

Здесь одной из посылок дедуктивного умозаключения является общая оценка («должно быть»), а другой – описательное утверждение о начальных условиях. В заключении общее предписание распространяется на частный случай, и тем самым достигается понимание, что надо делать с конкретным объектом.

Простые модальные оценки агентов имеют вид: $m(X \text{ есть } F)$, $m(X \text{ не есть } F)$, где F – ограничение на значение переменной X , $m \in M$, $M = \{\Box, \Diamond, \nabla, \neg\Diamond\}$ – множество модальных операторов, \Box – сильный положительный оператор, например оператор обязательства, \Diamond – слабый положительный оператор, например, оператор разрешения, $\neg\Diamond$ – сильный отрицательный оператор, к примеру, оператор запрещения, ∇ – слабый отрицательный оператор, к примеру, оператор безразличия. Схема вывода на основе простой модальной оценки: «плохо, что скорость ветра на мосту значительно выше нормы, значит, движение по мосту следует закрыть».

Оценки можно также выражать парами $\langle m, q \rangle$, где m – некоторая модальность, $m \in M$, а $q = a \text{ does } d$, $d \in D$ (агент a выполняет действие d из множества действий D). Например, «хорошо (поезд остановился вблизи неисправного переезда)».

Вариантами составных оценочных суждений с модальностями служат $m(X \text{ есть } F \text{ и } X \text{ не есть } F)$ (противоречивое суждение с модальностями), $m(X \text{ есть } F^+ \text{ или } X \text{ есть } F^-)$ (дизъюнктивное оценочное суждение с модальностями на оппозиционной шкале) и т.д.

Для логического описания сравнительных оценок служит исчисление предпочтений. В [фон Вригт, 1986], алфавит исчисления предпочтений образуют: 1) переменные x, y, z, \dots ; 2) логические связки $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$; 3) бинарное отношение предпочтения Pr ; с помощью этого отношения строятся как атомарные выражения вида $Pr(x, y)$ (читается « x лучше y »), так и составные выражения предпочтения (например, $Pr(x, y) \vee Pr(x, z)$). В общем случае, полагаем, что отношение нестрогого предпочтения Pr антисимметрично и рефлексивно. Его можно описать с помощью ориентированного графа. При этом обычно предпочтения не являются транзитивными.

Отношение равноценности (безразличия) Ind можно ввести как: $Ind(x, y) = \neg Pr(x, y) \wedge \neg Pr(y, x)$. Это отношение рефлексивно и симметрично.

Пусть заданы отношения предпочтения $Pr(x, y)$ и равноценности $Ind(x, y)$. Приведем аксиомы классической логики предпочтений [Ивин, 2010]:

1. $Pr(x, x) \geq Pr(x, y)$ (ничто не лучше самого себя);
2. $Pr(x, y) \rightarrow \neg Pr(y, x)$ (если первое лучше второго, то обратное невозможно). Иная формулировка:

$\lceil (Pr(x, y) \wedge Pr(y, x))$ (ничто не может быть и лучше, и хуже другого);

3. Если $Pr(x, y)$ и $Ind(y, z)$, то $Pr(x, z)$ (если первое лучше второго, а второе равноценно третьему, то первое лучше третьего);

4. $Pr(x, y) \vee Pr(y, x) \vee Ind(x, y)$ (принцип аксиологической полноты для сравнительных оценок: любые пары объектов таковы, что один из них или лучше другого, или хуже, или они равноценны).

Нечеткие отношения предпочтения для логического анализа сравнительных оценок описаны в [Тарасов и др., 2010].

2.2. Логические матрицы, логические миры и их наглядные модели

Предлагаемая ниже логико-алгебраическая методика интерпретации результатов измерений в интересах создания когнитивных сенсоров опирается как на представление истинностных значений в виде алгебраических структур, в частности, цепей, решеток, полурешеток и произведений решеток, так и на задание различных порядков в этих структурах.

Следуя А.С.Карпенко [Карпенко, 2010] и Я.В.Шрамко [Шрамко, 2009], будем называть *логическим миром* любую непустую совокупность логических (истинностных или ценностных) значений. Тремя основными принципами формирования любого логического мира являются: 1) *принцип различия* (объекты логического мира должны отличаться друг от друга); 2) *принцип выделенности* (некоторые из них должны иметь особый статус, т.е. быть выделенными); 3) *принцип структуризации* (объекты логического мира должны формировать определенные структуры). Отметим, что эти принципы фактически определяют *логическую онтологию*. Варианты построения гранулярных, в частности, нечетких и лингвистических онтологий, изложены в [Тарасов и др., 2012].

Важнейшим инструментом для единого алгебраического представления и исследования логических систем служат логические матрицы Лукасевича-Тарского. Под *логической матрицей* обычно понимается тройка

$$LM = \langle V, O, D \rangle, \quad (1)$$

где V есть непустое множество значений истинности; O – множество операций над этими значениями истинности v из V ; а $D \subset V$ – множество выделенных значений истинности.

Очевидно, что логическая матрица (1) может задаваться парой

$$LM = \langle UA, D \rangle, \quad (1^*)$$

где UA – универсальная алгебра с сигнатурой O .

В качестве конкретных примеров можно привести логическую матрицу классической логики $LM_{C2} = \langle \{0,1\}, \{\lceil, \wedge, \vee, \rightarrow\}, \{1\} \rangle$ или логическую

трехзначной логики Клини $LM_{K3} = \langle \{0, 0.5, 1\}, \{\lceil, \wedge, \vee, \rightarrow_K\}, \{1\} \rangle$, где промежуточное значение истинности 0.5 есть «неопределенность», а основные логические операции (отрицание, конъюнкция, дизъюнкция, импликация) задаются с помощью соответствующих таблиц истинности. Еще одним примером служит бесконечнозначная логика Лукасевича $LM_{L\infty} = \langle [0, 1], \{\lceil, \wedge, \vee, \rightarrow_L\}, \{1\} \rangle$, где промежуточные значения суть градации возможности, и для $\forall v, w \in [0, 1]$, $\lceil v = 1 - v$, $v \wedge w = \min(v, w)$, $v \vee w = \max(v, w)$, $v \rightarrow_L w = \min(1, 1 - v + w)$.

Вариантом простого расширения логической матрицы для классической логики LM_{C2} является логическая матрица для модальной логики

$$LM_{ML} = \langle \{0,1\}, \{\lceil, \wedge, \vee, \rightarrow, \diamond\} \{1\} \rangle, \quad (2)$$

где \diamond есть слабый положительный модальный оператор, например, «разрешено». Связь между сильным \square и слабым \diamond положительными модальными операторами задается в виде $\square p = \lceil \diamond \lceil p$, т.е. «обязательно p означает, что не разрешено не p ».

Примером переосмысления LM_{C2} служит логическая матрица для нечеткой логики, основанной на треугольных нормах T и треугольных конормах S

$$LM_{GFL} = \langle [0, 1], \{n, T, S, I\}, [\alpha, 1] \rangle, \quad (3)$$

где n – произвольная операция отрицания, например $n(v) = 1 - v$, $\forall v \in [0, 1]$, $I(v, w) = S(n(v), w)$, $\forall v, w \in [0, 1]$, α – некоторый порог истинности, $\alpha \in (0.5, 1)$.

В многозначных логиках различают собственно выделенные (т.е. подобные истине) значения D^+ , позволяющие обобщить понятие тавтологии и антивыведенные (т.е. подобные лжи) значения D^- , позволяющие обобщить понятие лжи или противоречия, причем нередки случаи, когда

$$D^+ \cap D^- \neq \emptyset, \text{ а } D^+ \cup D^- \neq V.$$

Тогда обобщенная логическая матрица (квазиматрица) для многозначной логики принимает вид

$$QLM = \langle V, QF, D^+, D^- \rangle, \quad (1^{**})$$

где QF – множество квазифункций.

В [Тарасов, 2004] понятие логической матрицы было расширено на случай ценностей и оценок. Была предложена так называемая *псевдологическая матрица*

$$PLM = \langle M, O, E \rangle, \quad (4)$$

где M есть непустое множество оценок (модальных значений) истинности; O – множество операций над оценками (значениями ценности) m из M ; а $E \subset M$ – множество выделенных оценок. Здесь также можно рассматривать эталоны (положительные образцы) E^+ и антиэталонные оценки (отрицательные образцы) E^- .

Развитие аппарата логических матриц в плане построения гранулярных логических структур

может опираться на обобщенную матрицу Вуйцицкого LM_W :

$$LM_W = \langle A, C \rangle, \quad (5)$$

где A – алгебра соответствующего типа с универсумом V , а C – произвольное семейство подмножеств A .

Отметим, что в матрице Лукасевича-Тарского (1) предполагается возможность получения истинностной оценки высказывания вне зависимости от контекста, от условий оценки. В случае прагматических оценок истинности необходимо учитывать подобный контекст, например, путем задания онтологий с помощью множества различных логических миров (пространств), т.е. следует расширить классическую логическую матрицу (1), записав ее в виде гранулярной матрицы

$$GLM = \langle A, V, 2^V O_g, D_g^+, D_g^- \rangle, \quad (6)$$

где A – множество типов логических значений (например, истинностные значения, ценностные значения), V – множество логических значений, 2^V – множество гранулярных логических значений, O_g – множество логических операций с гранулярными логическими значениями, $D_g^+ \subset 2^V$, $D_g^- \subset 2^V$ – множества выделенных и антивыведенных гранул соответственно.

Для представления простейших логических миров вполне достаточно указать состав и мощность множества значений истинности V , состав, мощность и знак множества выделенных значений истинности D , где $D \subset V$, а также отношения порядка на множестве V с помощью диаграмм Хассе.

Ниже будем описывать логический мир как универсум логических значений V_i вместе с множеством выделенных значений D_j , понимаемых как абстрактные логические объекты и множеством отношений порядка R , определенных на V_i , т.е. как

$$LW = \langle V_i, R, D_j^z \rangle, \quad (7)$$

где $i=1,2,\dots,n,\dots,\infty$, $j < i$, $Z = \{+, -\}$ (см. [Тарасов, 2014]). Примерами различных отношений порядка из R являются порядки истинности $<_V$, \leq_V и информационные порядки $<_I$, \leq_I .

Тогда классический логический мир Фреге задается в виде $LW_{Cl2} = \langle V_2, <_V, D^+ \rangle$, где $V_2 = \{T, F\}$, $D^+ = \{T\}$, $D^- = \{F\}$, причем: 1) $D^+ \cap D^- = \emptyset$, 2) $D^+ \cup D^- = V_2$, а логический мир модализированных истинностей Н.Решера определяется как $LW_{R4} = \langle MV_4, <_M, <_V, D^+ \rangle$, где $MV_4 = M_2 \times V_2$, $M_2 = \{L, Q\}$, L – «необходимо», Q – «случайно», $<_M$ – отношение порядка на M_2 , $<_V$ – отношение порядка на V_2 , $D^+ = \{L, T\}$ (или иначе $M_2^* = \{P, 3\}$, P – «разрешено», 3 – «запрещено», $<_M^*$ – отношение порядка на M_2^*).

Логический мир Лукасевича выражается тройкой $LW_{L3} = \langle V_{L3}, <_V, D^+ \rangle$, где $V_{L3} = \{T, N, F\}$, $D^+ = \{T\}$, а нейтральное значение N («ни истина, ни

ложь») понимается как «возможность» или «безразличие».

В свою очередь, парapolный логический мир Клини уже содержит два отношения порядка $<_V$, \leq_I (порядок истинности и информационный порядок), а N трактуется как «неопределенность». Его можно формализовать в виде

$$LW_{K3} = \langle V_{K3}, <_V, \leq_I, D^+ \rangle, \quad (8)$$

где $V_{K3} = \{T, N, F\}$, $D^+ = \{T\}$.

Отметим, что для мира Клини справедливо условие 1) для классического мира Фреге, но не выполняется условие 2).

Паранепротиворечивый мир Васильева-Приста описывается в виде

$$LW_{VP3} = \langle V_{VP3}, <_V, \leq_I, D^+ \rangle, \quad (9)$$

где $V_{VP3} = \{T, B, F\}$, $D^+ = \{T, B\}$, B – «и истина, и ложь одновременно», что означает «противоречие».

Наконец, логический мир Данна-Белнапа имеет вид

$$LW_{DB4} = \langle V_{DB4}, \leq_V, \leq_I, D^+ \rangle, \quad (10)$$

где $V_{DB4} = \{T, B, N, F\}$, \leq_V – порядок истинности, \leq_I – информационный порядок, $D^+ = \{T, B\}$ (см. [Тарасов, 2014]).

Примеры наглядного представления различных логических миров даны на рис. 4а-г. Вариант визуализации мира Данна-Белнапа изображен на рис.5.

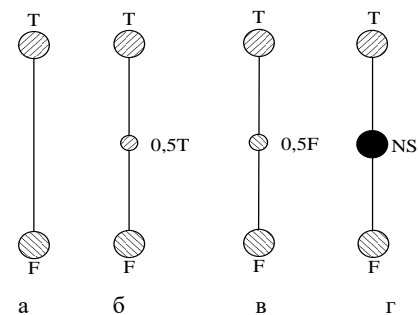


Рисунок 4 – Наглядное представление логических миров цепями с помеченными (заштрихованными) значениями истинности: а) логический мир Фреге; б) логический мир Гейтинга LW_{H3} , где 0.5 T есть «половинчатая истина»; в) логический мир Брауэра LW_{B3} , 0.5 F – «половинчатая ложь» г) логический мир Бочвара, LW_{B3} , значение NS (Non-Sense) есть сильная бессмыслица.

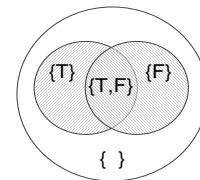


Рисунок 5 – Вариант графического представления логического мира Данна-Белнапа

При выборе направления косой штриховки для T и F мы исходили из ряда принципов организации образного мышления, описанных в [Поспелов, 1998], в частности, из принципов равновесия и простоты. У человека есть врожденная система эталонов, которые для него просты. Если речь идет

о геометрических фигурах, то это – квадрат и круг.

Что же касается направлений, то диагональное направление слева направо и вверх – это стандартный эталон возрастания или улучшения, а слева направо и вниз – убывания или ухудшения. Поскольку $T > F$, косая штриховка в первом направлении выбрана для иллюстрации предпочтительного значения «истина», а косая штриховка по второму направлению – для значения «ложь». Появление двойной штриховки (сетки) на рисунке 5 соответствует составному значению истинности $V = \{T, F\}$.

Логический мир для модальностей (рисунок 6) построен на базе представления положительных модальностей вертикальными линиями и отрицательных модальностей – горизонтальными линиями, а также с учетом силы модальности (сильные модальности отображены более жирными линиями по сравнению со слабыми). Переход от косой к прямой штриховке демонстрирует хорошо известный в логике факт, что сильная положительная модальность (например, алетическая модальность «необходимость») в определенном смысле сильнее истины, а сильная отрицательная модальность (например, «невозможность») – сильнее лжи.

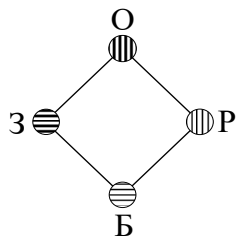


Рисунок 6 – Логический мир деонтических модальностей (норм): О – обязательно, Р – разрешено, Б – безразлично, З – запрещено

Логический мир де Клира (рисунок 7), связанный с диагностикой, можно трактовать как описание результатов измерений, связанных с получением положительных, отрицательных или нейтральных оценок сложного технического объекта («можно эксплуатировать», «нельзя эксплуатировать», «непонятно»).

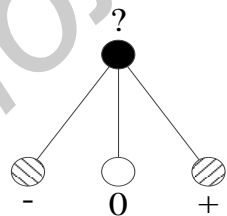


Рисунок 7 – Иллюстрация логического мира Де Клира

Наконец, диалогические миры иллюстрируют коммуникативную (распределенную) трактовку истины как продукта согласия (см. решетку согласия в [Тарасов, 2008]) или сведение ценности к победе в споре (рисунок 8).



Рисунок 8 – Представление диалогических миров:
а) мир переговоров (согласованная истина);
б) мир спора (ценность как победа в споре).

3. Трехзначные логические прагматики Васильева и его последователей

Логические модели понимания результатов измерений могут опираться на многозначные, паранепротиворечивые и парapolные логики. Одним из родоначальников этих логик был русский ученый из Казани Н.А.Васильев, который одним из первых сформулировал принцип логического плюрализма, отметив, что классическая логика является неединственной и неуниверсальной, подобно тому, как неединственной оказалась евклидова геометрия [Васильев, 1989]. Возвращаясь к идеям Дж.С.Милля о том, что законы логики являются обобщением опыта, он также считал, что выполнение этих законов и выбор логики зависит от свойств окружающей реальности или наших ощущений (т.е. выражаясь современным языком, полагал, что логика зависит от принятой системы онтологий). Согласно Н.А.Васильеву, реальная человеческая логика отличается двойственным характером, а именно, она полуэмпирична, полурациональна, и поэтому ей может быть противопоставлена чисто формальная и рациональная дисциплина – металогика. Все внеэмпирические элементы и отношения в логике составляют металогику.

В результате Н.А.Васильев предложил многомерную (воображаемую) логику [Васильев, 1989], а также ввел двухуровневую логическую структуру (эмпирическая логика и металогика). Он неоднократно подчеркивал, что существуют глубокие внутренние аналогии между геометрией Н.И.Лобачевского и воображаемой логикой. Исходным пунктом геометрии Лобачевского являлся отказ от знаменитого пятого постулата Эвклида о параллельных прямых, вследствие чего была построена геометрия, свободная от этого постулата. Аналогично отправной точкой логики Н.А.Васильева стал отказ от одного из важнейших законов аристотелевой логики – закона непротиворечия – и построение логики, свободной от этого закона.

Введение двухуровневой структуры позволяет глубже проникнуть в природу нашей логики, разделить в ней эмпирические и неэмпирические элементы. Изменяя онтологию, комбинируя различные свойства реальности, можно получить различные воображаемые логики. Этот метод в логике аналогичен экспериментальному и

сравнительному методам в естествознании [Васильев, 1989].

По сути, двухуровневая логическая структура может пониматься как ранний вариант логической грануляции, перехода от мелкозернистой к крупнозернистой прагматике.

Хорошей иллюстрацией варианта укрупненного представления результатов измерений средствами трехзначной логики и графики служит треугольник Васильева с заштрихованными вершинами (рисунок 9), описывающими положительное, отрицательное и противоречивое высказывание

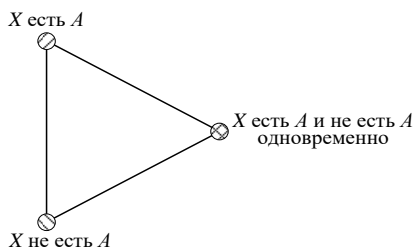


Рисунок 9 – Треугольник Васильева с заштрихованными вершинами

Здесь положительное высказывание « X есть A » соответствует предложению «значения измеренного параметра находятся в норме, т.е. в зеленой зоне», отрицательное высказывание « X не есть A » трактуется как «значения измеренного параметра существенно выходят за пределы нормы, т.е. находятся в красной зоне», а составное противоречивое высказывание « X есть A и не есть A одновременно» является примером логической гранулы, характеризующей «желтую зону» переопределенности, т.е. промежуточную область между нормой и ее отрицанием.

В классической логике некоторая теория называется *противоречивой*, если в ней можно одновременно доказать и предложение, и его отрицание. Если при этом в теории можно доказать и произвольное предложение, то она называется *тривиальной*. Тогда паранепротиворечивой логикой называется такая логика, которая не позволяет выводить из противоречия произвольное предложение. Иными словами, в ней не имеет место фундаментальный принцип классической логики: «из противоречия следует все, что угодно».

Объективной причиной появления и развития паранепротиворечивых логик явилось стремление отразить средствами логики специфику человеческого понимания переходных состояний. Примерами паранепротиворечивых логик служат: логика да Коста, логика Асеньо, логика Приста и др. (см.[Павлов, 2004]).

Как уже отмечалось, паранепротиворечивый логический мир характеризуется тройкой $V_{VP3} = \{T, B, F\}$, где $T = \{T\}$ и $F = \{F\}$ – два одноэлементных множества, а $B = \{T, F\}$ – двухэлементное множество. Определим на множестве V_3 два различных отношения порядка – порядок истинности $<_V$ и информационный порядок $<_I$. Очевидно, что $F <_V B <_V T$.

Для пояснения информационного порядка $<_I$ воспользуемся комбинаторным подходом А.Н. Колмогорова в теории информации, согласно которому количество информации прямо пропорционально мощности множества, тогда $F <_I B$ и $T <_I B$. Диаграммы Хассе со штриховкой, изображающие структуры паранепротиворечивого мира, представлены на рисунках.10 а и б, где упорядоченное множество $\langle V_3, <_V \rangle$ образует решетку L_3 , а $\langle V_3, <_I \rangle$ – верхнюю полурешетку. На рисунках 11 а, б и в показаны варианты задания выделенных и антивывделенных значений в трехзначных паранепротиворечивых логиках.

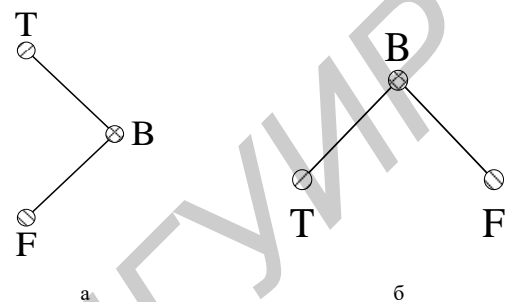


Рисунок 10 – Структуры паранепротиворечивого логического мира: а) логическая решетка L_3 ; б) информационная полурешетка, полученная поворотом L_3 на 90° влево.

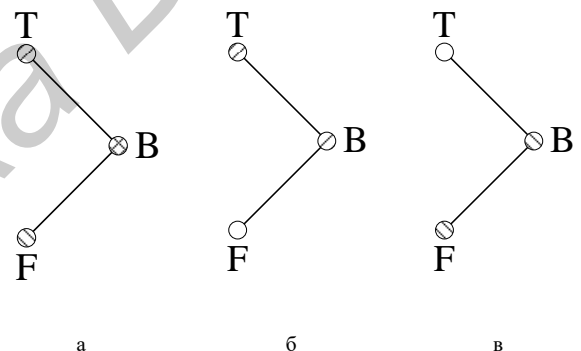


Рисунок 11 – Варианты задания выделенных (а, б) и антивывделенных (а, в) значений в трехзначных паранепротиворечивых логиках

3.1. Трехзначная логика Асеньо A_3

Канонической паранепротиворечивой логикой может считаться трехзначная логика Асеньо A_3 , которая описывается логической матрицей

$$LM_{A_3} = \langle \{T, B, F\}, \{\bar{\quad}, \wedge\}, \{T, B\} \rangle \quad (11)$$

Здесь значение B понимается как «антиномия». Базовые операции логики Асеньо задаются следующими таблицами истинности:

x	$\neg x$
T	F
B	B
F	T

$x \wedge y$	T	B	F
T	T	B	F
B	B	B	F
F	F	F	F

$x \vee y$	T	B	F
T	T	T	T
B	T	B	B
F	T	B	F

$x \rightarrow y$	T	B	F
T	T	B	F
B	T	B	B
F	T	T	T

Паранепротиворечивые логики Васильева и его последователей служат основой для представления трехзначной прагматики когнитивного *сенсора Васильева*, в котором полученная путем измерений информация гранулируется следующим образом:

- 1) Т – «измеренная истина» («параметр в норме» – показания сенсора находятся в «зеленой зоне»);
- 2) F – «измеренная ложь» («параметр не в норме», «отказ» – показания сенсора попали в «красную зону»);
- 3) В – «измеренное противоречие» или антиномия («пограничная ситуация», «предотказ» – показания сенсора локализованы в «желтой зоне»).

3.2. Трехзначная логика Приста Pr_3

Г. Прист построил трехзначную логику Pr_3 , введя в качестве третьего истинностного значения «парадоксально», которое принимается в качестве второго выделенного значения (первое – Т).

«Предложение будем называть парадоксальным, если оно истинно и ложно одновременно. Если предложение истинно и неложно, будем называть его «только истинным» и аналогично для «только ложного» предложения».

«Предложение истинно – его отрицание ложно. Отрицание истинного и ложного предложения истинно и ложно, т.е. парадоксально. Отрицание только истинного предложения только ложно. Отрицание только ложного предложения только истинно».

Таблица истинности импликации логики Приста соответствует таблице исходной импликации логики Клини K_3 (см. далее), отличаясь только двумя выделенными значениями.

3.3. Трехзначная логика осмысленности В.К.Финна F_3

Одним из новых примеров логики оценок является трехзначная логика осмысленности В.К.Финна [Финн, 2010], родственная трехзначной логике Д.А.Бочвара (см. рис.4г). Она соответствует концепции понимания как осмысления. Ее логическая матрица записывается в виде

$$LM_{F_3} = (\{O, H, B\}, \{\neg, \rightarrow, \wedge, \vee\}, \{O\}), \quad (12)$$

где О,Н,Б суть логические значения «осмысленно», «неопределенно», «бессмысленно» соответственно, причем различаются логики с «сильной и слабой бессмыслицей». Основные логические связки определяются с помощью следующих логических таблиц.

x	$\neg x$
О	Б
Н	Н
Б	О

$x \rightarrow y$	О	Н	Б
О	О	Н	Б
Н	О	О	Н
Б	Б	Н	О

$x \vee y$	О	Н	Б
О	О	О	Б
Н	О	Н	Б
Б	Б	Б	Б

$x \wedge y$	О	Н	Б
О	О	Н	Б
Н	Н	Н	Б
Б	Б	Б	Б

Предложенную логику F_3 , для которой в конъюнкции и дизъюнкции работает принцип «заражения бессмыслицей», можно назвать логикой с «сильной бессмыслицей». Но можно рассмотреть альтернативный вариант трехзначной логики осмысленности F_3' с другой дизъюнкцией

$x \vee y$	О	Н	Б
О	О	О	О
Н	О	Н	Н
Б	О	Н	Б

Это логика со «слабой бессмыслицей».

4. Трехзначные логические прагматики Клини

В условиях реального мира информация о некотором факте, выраженная предложением p , может быть недостаточно определенной. В своей работе «On a Notation for Ordinal Numbers» (см. [Клини, 2009]) С.К.Клини предложил трехзначную логику со значениями Т – истинно, F – ложно и N – не определено, для которой не работает закон исключенного третьего, но зато справедлив закон исключенного четвертого $T \vee F \vee N$.

Логика Клини K_3 стала одной из первых (если не первой) из многозначных логик, опирающейся на эпистемологические соображения, что объясняет естественность ее использования в информатике [Fitting, 1991]. По С.К.Клини, логические связки должны определяться регулярными таблицами в следующем смысле: данный столбец (строка) содержит Т в строке (столбце) для N только при условии, что этот столбец (строка) состоит целиком из Т. Аналогичное условие берется для F.

Параполный логический мир характеризуется тройкой $V_{K_3} = \{T, N, F\}$, где $T = \{T\}$ и $F = \{F\}$ – два одноэлементных множества, а $N = \emptyset$. Таким образом, логический мир Клини является основой для определения когнитивного *сенсора Клини* с трехзначной логической прагматикой:

- 1) Т – «измеренная истина» («параметр в норме» – показания сенсора находятся в «зеленой зоне»);
- 2) F – «измеренная ложь» («параметр не в норме», «отказ» – показания сенсора в «красной зоне»);
- 3) N – «неопределенность» (значения параметра неизвестны, что будем изображать синим цветом; это возможная ситуация в беспроводной сенсорной сети, когда сенсор исчерпал свой ресурс и «спит»).

Определим на множестве V_{K_3} два различных отношения порядка – порядок истинности $<_v$ и информационный порядок $<_i$. Здесь $F <_v N <_v T$, но $N <_i F$ и $N <_i T$. Соответственно, упорядоченное множество $\langle V_{K_3}, <_v \rangle$ образует решетку, а $\langle V_3, <_i \rangle$ – верхнюю полурешетку (рисунк 12).

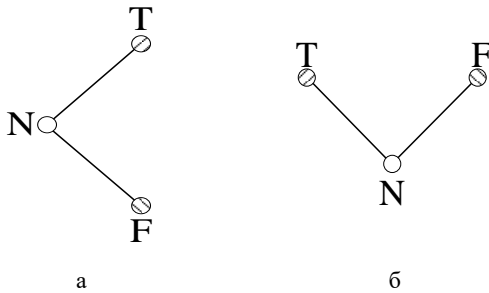


Рисунок 12 - Структуры парапологого логического мира:
 а) решетка сильной логики Клини K_3 ;
 б) нижняя полурешетка;

Логика Клини K_3 описывается логической матрицей

$$LM_{K_3} = \langle \{T, N, F\}, \{\neg, \vee\}, \{T\} \rangle, \quad (13)$$

где логические связи задаются следующими таблицами истинности:

x	$\neg x$
T	F
N	N
F	T

$x \vee y$	T	N	F
T	T	T	T
N	T	N	N
F	T	N	F

$x \wedge y$	T	N	F
T	T	N	F
N	N	N	F
F	F	F	F

$x \rightarrow y$	T	N	F
T	T	N	F
N	T	N	N
F	T	T	T

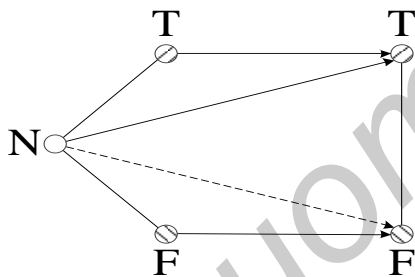


Рисунок 13 –Трехзначная прагматика логики Клини с неполной информацией и переход к металогики

Наконец, в случае интерпретации значения N как недоопределенности, неполноты информации можно в случае получения нужной информации в следующий интервал времени перейти к двузначной металогики, т.е. подтвердить значения «измеренная истину» или «измеренная ложь».

5. Четырехзначные логические прагматики Данна-Белнапа

Одним из известных способов грануляции информации в логической семантике является переход к *обобщенным значениям истинности* (по Дж. Данну [Dunn, 1976]), когда они задаются не на обычном множестве V , а на множестве всех подмножеств 2^V . Согласно Дж. Данну допускаются ситуации, когда высказывания являются одновременно и истинными, и ложными (отказ от принципа бивалентности), или не являются ни

истинными, ни ложными. Первый из таких нестандартных случаев называют «пресыщенной оценкой» (Glut), а второй – «истиннозначным провалом» (Gap). Таким образом, истинность понимается как функция множества, т.е. ее значениями могут быть не только элементы множества V , но и любые его подмножества, включая пустое множество \emptyset , т.е. осуществляется переход от V к 2^V . Эти подмножества называются обобщенными истинностными значениями и вполне соответствуют трактовке истинности по Фреге-Черчу как обобщенного объекта. В простейшем случае для $|V|= 2$ имеем $2^V = \{T, B, N, F\}$, где $B = \{T, F\}$, $N = \emptyset$ (логическая прагматика Белнапа).

При построении своей четырехзначной логики Белнап [Belnap, 1977; Белнап и др., 1981] рассмотрел два отношения частичного порядка: порядок истинности \leq_i и порядок включения на множестве $\{\emptyset, \{T\}, \{F\}, \{T, F\}\}$. В первом случае имеем $F < N < T$ и $F < B < T$, а значения B и N, находящиеся между F и T, являются несравнимыми. Во втором случае получаем $\emptyset \subset \{T\} \subset \{T, F\}$ и $\emptyset \subset \{F\} \subset \{T, F\}$, т.е. T и F являются несравнимыми. Первый порядок задает полную решетку значений истинности, которая названа Белнапом логической решеткой L_4 (рисунок 14), а второму порядку соответствует решетка Скотта A_4 (рисунок 15), которую в данном контексте следует именовать информационной решеткой). При этом решетка Скотта получается путем поворота логической решетки Белнапа против часовой стрелки на 90° .

Соответственно логика Белнапа служат основой для представления четырехзначной прагматики когнитивного *сенсора Белнапа*, в котором полученная путем измерений информация гранулируется следующим образом:

- 1) T – «измеренная истина» («параметр в норме» – показания сенсора находятся в «зеленой зоне»);
- 2) F – «измеренная ложь» («параметр не в норме», «отказ» – показания сенсора попали в «красную зону»);
- 3) B – «измеренное противоречие» или антиномия («пограничная ситуация», «предотказ» – показания сенсора локализованы в «желтой зоне»);
- 4) N – «неопределенность» (значения параметра неизвестны, поскольку датчик неисправен или «спит»; оценку N будем ассоциировать с синим цветом).

При этом в случае задачи мониторинга опасных состояний объектов инфраструктуры в логической матрице для белнаповского сенсора LM_{B_4} будем использовать антивыделенные значения F и N

$$LM_{B_4} = \langle \{T, B, N, F\}, \{\neg, \wedge, \vee, \rightarrow\} \{F, N\} \rangle. \quad (14)$$

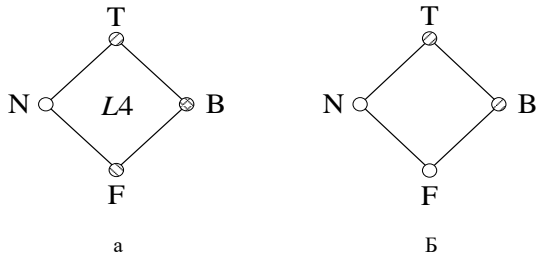


Рисунок 14 – Диаграммы Хассе со штриховкой для решетки Белнапа (а) и выделенных значений в логике Белнапа (б)

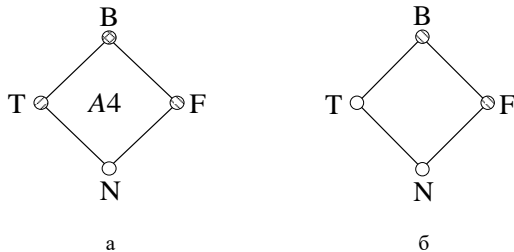


Рисунок 15 – Диаграммы Хассе со штриховкой для решетки Скотта (а) и антивывделенных значений в логике Скотта (б)

6. Интерпретации мультисенсорных данных: бирешеточные прагматики

Прагматика когнитивной сенсорной сети [Святкина и др.,2012], состоящей из n васьильевских сенсоров, будет описываться формулой 3^n , а сети из n белнаповских сенсоров – формулой 4^n , где n – целое число, $n > 1$. Так простейшая структура из двух белнаповских сенсоров образует множество прагматических значений $4^2=16$, сеть из трех датчиков – $4^3=64$ и т.д.

При построении логических гранул на основе мультисенсорных данных будем использовать произведения решеток и мультирешетки. Следуя Я.В.Шрамко [Шрамко,2002], назовем n -мерной мультирешеткой (или просто n -решеткой) структуру вида

$$ML = \langle X, \leq_1, \dots, \leq_n \rangle, \quad (15)$$

где X есть непустое множество (например, $X = X_1 \times \dots \times X_n$), а \leq_1, \dots, \leq_n – отношения частичного порядка, определенные на X , так что $(X, \leq_1), \dots, (X, \leq_n)$ суть различные решетки.

Для приложений в логике когнитивных сенсорных сетей важную роль играют правильные мультирешетки, в которых все частичные порядки можно определить взаимно независимым образом.

Далее возьмем в качестве базовой единицы когнитивной сенсорной сети минимальную микросеть, состоящую из двух датчиков. Тогда исходная задача состоит в построении прагматической структуры оценок измерений с помощью аппарата бирешеток. Бирешетку естественно определять с помощью понятий биупорядоченного множества и предбирешетки [Fitting, 1989].

Биупорядоченное множество есть структура вида $BOS = \langle X, \leq_1, \leq_2 \rangle$, где X – непустое множество, содержащее, по крайней мере, два

элемента, а \leq_1 и \leq_2 – два отношения частичного порядка. Эта структура становится предбирешеткой PBL , если оба упорядоченных множества $\langle X, \leq_1 \rangle$ и $\langle X, \leq_2 \rangle$ образуют полные решетки. Если $PBL = \langle X, \leq_1, \leq_2 \rangle$ – предбирешетка, то на ней можно задать одни операции объединения \vee и пересечения \wedge на решетке $\langle X, \leq_1 \rangle$, и другие операции объединения \oplus и пересечения \otimes на решётке $\langle X, \leq_2 \rangle$. Следует отметить, что бимодальные системы образуют предбирешетки (но не бирешетки).

Для построения бирешетки необходимо установить связь между двумя отношениями порядка. Основоположник теории бирешеток М.Гинзберг (см.[Ginsberg,1988]) делает это с помощью специальной операции составного неоднородного отрицания, которая является естественным расширением отрицания Белнапа.

Отрицание Гинзберга есть функция

$$\neg: X \rightarrow X, \quad (16)$$

которая удовлетворяет следующим условиям: $\forall x, y \in X$: 1) если $x \leq_1 y$, то $\neg x \geq_1 \neg y$; 2) если $x \leq_2 y$, то $\neg x \leq_2 \neg y$; 3) $\neg(\neg x) = x$.

Тогда бирешеткой называется четверка [Ginsberg, 1988]

$$BL = \langle X, \leq_1, \leq_2, \neg \rangle, \quad (17)$$

где $X \neq \emptyset$, $|X| \geq 4$, \leq_1 и \leq_2 – два различных отношения порядка, заданных на множестве X , а \neg есть отрицание Гинзберга (16), удовлетворяющее условиям 1) – 3).

Очевидно, что бирешетка может строиться как алгебра с двумя различными операциями пересечения и объединения соответственно, т.е. задаваться шестеркой:

$$BL = \langle X, \wedge, \vee, \otimes, \oplus, \neg \rangle, \quad (17^*)$$

где: 1) решетки $\langle X, \wedge, \vee \rangle$ и $\langle X, \otimes, \oplus \rangle$ – полные; 2) \neg есть отображение вида $\neg: X \rightarrow X$, такое что (а) $\neg^2 = 1$; (б) \neg есть гомоморфизм решетки $\langle X, \wedge, \vee \rangle$ в решетку $\langle X, \vee, \wedge \rangle$ и автоморфизм решетки $\langle X, \otimes, \oplus \rangle$.

Прагматика бисенсорных данных для двух сенсоров Белнапа изображена в виде двойной диаграммы Хассе (рисунок 16), а соответствующая общая карта отказов и предотказов приведена на рисунке 17. В логике Белнапа значения B и N рассматривают независимо друг от друга, поэтому при ее консервативном расширении значения B_1N_2 и N_1B_2 запрещены. Таким образом, для двух белнаповских сенсоров имеем $|V| = 14$.

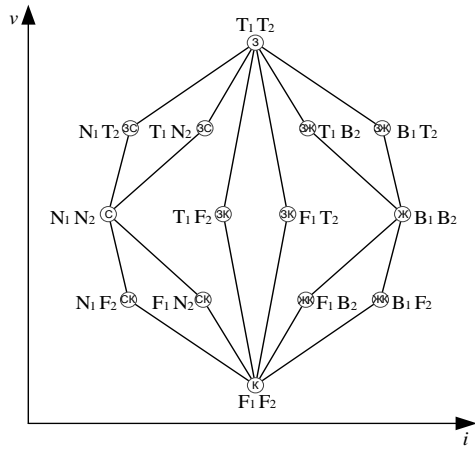


Рисунок 16 – Бирешоточное представление прагматики логических значений для структуры из двух белнаповских сенсоров

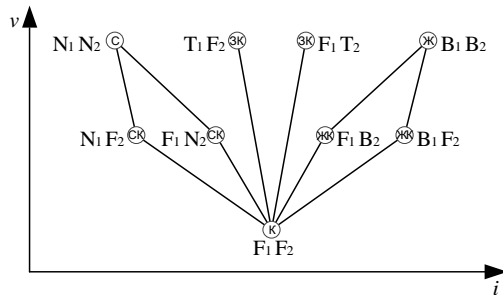


Рисунок 17 – Логическая карта отказов и предотказов

Рассмотрим все прагматические значения для структуры из двух белнаповских сенсоров. Здесь $T_1 T_2$ – «измеренная согласованная истина» (показания обоих сенсоров характеризуются значением «норма»); $F_1 F_2$ – «измеренная согласованная ложь» (оба сенсора сигнализируют о состоянии отказа); $T_1 B_2 \sim B_1 T_2$ – «измеренное частичное противоречие как предотказ 1-го рода» (один сенсор показывает значение «норма», а другой – «предотказ»); $T_1 N_2 \sim N_1 T_2$ – «измеренная частичная истина с неопределенностью» (один сенсор показывает значение «норма», а другой «спит»); $T_1 F_2 \sim F_1 T_2$ – «измеренное противоречие» (один сенсор показывает значение «норма», а другой – «отказ»); $B_1 B_2$ – «измеренное согласованное противоречие» (оба сенсора передают состояние «предотказ»); $N_1 N_2$ – «согласованная неопределенность» (оба сенсора исчерпали свои ресурсы или оба сенсора «спят»); $F_1 B_2 \sim B_1 F_2$ – «измеренное частичное противоречие как предотказ 2-го рода» (один сенсор показывает значение «отказ», а другой – «предотказ»); $F_1 N_2 \sim N_1 F_2$ – «измеренная частичная ложь с неопределенностью» (один сенсор показывает значение «отказ», а другой спит).

7. Методика логико-алгебраического синтеза когнитивных сенсоров

В результате предложена общая методика построения когнитивных сенсоров, представленная в виде алгоритма синтеза. Блок-схема этого алгоритма приведена на рисунке 18.

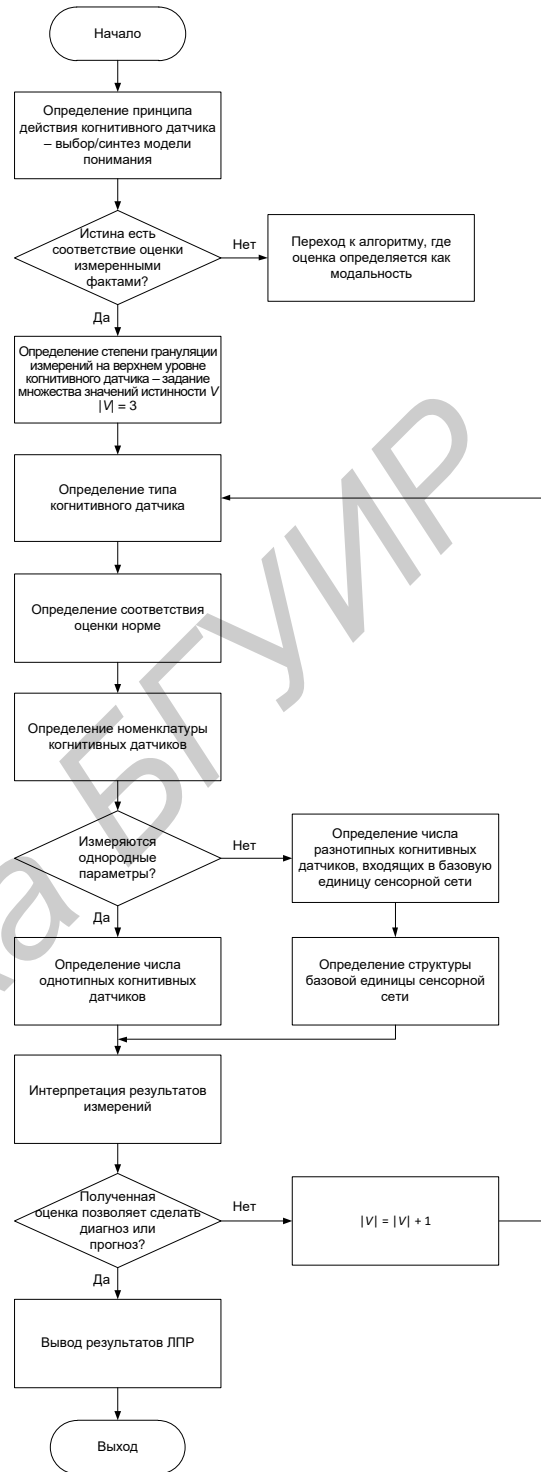


Рисунок 18 – Блок-схема алгоритма построения когнитивных датчиков для мониторинга объектов инфраструктуры

Из этого рисунка видно, что важнейшими этапами логико-алгебраического синтеза являются: 1) определение принципа действия когнитивного датчика, неотделимое от выбранной модели понимания; 2) задание степени точности измерений и степени грануляции прагматических оценок; 3) определение исходного типа когнитивного сенсора с возможностью его последующей замены на базе алгебраической модели структуры логических значений; 4) построение номенклатуры требуемых когнитивных датчиков и определение их числа; 5) предварительная и окончательная интерпретации

результатов измерений в контексте решения проблем мониторинга.

Заключение

Главные результаты данной работы состоят в формировании теоретических основ разработки открытых прагматических технологий синтеза интеллектуальных систем, развитии формального аппарата логических миров и логических матриц применительно к разработке когнитивных сенсоров, приложении идей и подходов иллюстративной и когнитивной графики в многозначных логиках для представления прагматики логических значений, создании методики логико-алгебраического синтеза когнитивных сенсоров.

Работа выполнена при финансовой поддержке РФФИ, проекты №14-07-00846, №14-07-00653 и №14-07-31317.

Библиографический список

- [Бахтин, 1979] Бахтин, М.М. К методологии гуманитарных наук/ М.М. Бахтин // Эстетика словесного творчества. – М.: Искусство, 1979.
- [Белнап и др., 1981] Белнап, Н. Логика вопросов и ответов: Пер. с англ./ Н. Белнап, Т. Стил / Под ред. В.А. Смирнова, В.К. Филина. – М.: Прогресс 1981.
- [Борисов и др., 2006] Борисов, А.В. Моделирование мнений агентов в многоагентных системах на основе четырехзначных семантик / А.В. Борисов, В.Б. Тарасов // Программные продукты и системы. – 2006. – №2. – С.47-50.
- [Будбаева и др., 1974] Будбаева, С.П. К исследованию и построению прагматических логик/С.П.Будбаева, Б.Н.Пятницкий // Философия и логика. – М.: Наука, 1974. – С.220-278.
- [Васильев, 1989] Васильев, Н.А. Воображаемая логика / Н.А. Васильев. – М.: Наука, 1989.
- [Вригт фон, 1986] Вригт фон, Г.Х. Логико-философские исследования. Избранные труды: Пер. с англ. / Г.Х. Вригт фон; – М.: Прогресс, 1986.
- [Голенков и др., 2013] Голенков, В.В. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем/ В.В.Голенков, Н.А. Гулякина//Открытые семантические технологии проектирования интеллектуальных систем. Материалы III-й международной научно-технической конференции (Минск, БГУИР, 21-23 февраля 2013 г.). – Минск: Изд-во БГУИР, 2013. – С.55-77.
- [Зенкин, 1991] Зенкин, А.А. Когнитивная компьютерная графика/ А.А.Зенкин; – М.: Наука, 1991.
- [Ивин, 2016] Ивин, А.А. Логика оценок и норм. Философские, методологические и прикладные аспекты. / А.А. Ивин; – М.: Проспект, 2016.
- [Ивин и др., 1998] Ивин, А.А. Словарь по логике/ А.А. Ивин, А.Л. Никифоров; – М.: ГИЦ ВЛАДОС, 1997.
- [Карнап, 2007] Карнап Р., О некоторых понятиях прагматики /Р.Карнап// Значение и необходимость. Исследование по семантике и модальной логике. – М.: Изд-во ЛКИ, 2007.
- [Карпенко, 2010] Карпенко, А.С. Развитие многозначной логики/ А.С. Карпенко; – М.: Изд-во ЛКИ, 2010.
- [Клини, 2009] Клини, С.К. Введение в метаматематику. 2-е изд. / С.К. Клини. – М.: Изд-во УРСС, 2009.
- [Кнорринг, 1992] Кнорринг, В.Г. Гносеотехника – техника познания/ В.Г. Кнорринг// Измерения. Контроль. Автоматизация. – 1992. – №.1-2. – С.3-12.
- [Кузнецов, 2008] Кузнецов, О.П. Формальный подход к понятию «знание» и проблема моделирования различных типов знания/ О.П.Кузнецов// Когнитивные исследования: Сборник научных трудов. Вып.2/ Под ред. В.Д.Соловьева, Т.В. Черниговской.–М.: Институт психологии РАН, 2008. –С.265-275.
- [Лукаевич, 1993] Лукаевич, Я. О детерминизме/ Я. Лукаевич// Логические исследования. Вып.2. – М.: Наука, 1993. – С. 190-205.
- [Новак и др., 2006] Новак, В. Математические принципы нечеткой логики: Пер. с англ. / В. Новак, И.Г. Перфильева, И. Мечкарж; – М.: Физматлит, 2006.
- [Павлов, 2004] Павлов, С.А. Логика с операторами истинности и ложности / С.А. Павлов; – М.: ИФ РАН, 2004.
- [Пирс, 2000] Пирс, Ч.С. Начала прагматизма. / Ч.С. Пирс; – СПб: Алестейя, 2000.
- [Попов, 2004] Попов, Э.В. Общение с ЭВМ на естественном языке. 2-е изд. / Э.В. Попов; – М.: Едиториал УРСС, 2004.
- [Поспелов, 1989] Поспелов, Д.А. Интеллектуальные интерфейсы для ЭВМ новых поколений/ Д.А. Поспелов // Электронная вычислительная техника. Сборник статей. Вып.3. – М.: Радио и связь, 1989. – С.4-20.
- [Поспелов, 1992] Поспелов, Д.А. Когнитивная графика – окно в новый мир / Д.А. Поспелов // Программные продукты и системы. – 1992. – №3. – С.4-6.
- [Поспелов, 1998] Поспелов, Д.А. Метафора, образ и символ в познании мира/ Д.А.Поспелов// Новости искусственного интеллекта. – 1998. – №3. – С.94-114.
- [Прокопчина, 2010] Прокопчина, С.В. Когнитивные измерения на основе байесовских интеллектуальных технологий/ С.В.Прокопчина// Сборник докладов XIII-й международной конференции по мягким вычислениям и измерениям (SCM'2010, Санкт-Петербург, СПбГЭТУ «ЛЭТИ», 23-25 июня 2010г.). – СПб: Изд-во СПбГЭТУ «ЛЭТИ», 2010. –С.28-34
- [Пуанкаре, 1990] Пуанкаре, А. О науке: Пер с франц./ А. Пуанкаре. – М.: Наука, 1990.
- [Розенберг, 1975] Розенберг, В.Я. Введение в теорию точности измерительных систем/ В.Я. Розенберг. – М.: Сов. Радио, 1975.
- [Святкина и др., 2012] Святкина, М.Н. Логика сенсорных сетей/М.Н.Святкина, В.Б.Тарасов//Гибридные и синергетические интеллектуальные системы: теория и практика. Материалы 1-го международного симпозиума (ГИСИС'2012, Калининград, Светлогорск, 29 июня – 2 июля 2012г.). – Калининград: Изд-во БФУ им. И.Канта, 2012. – Часть 2. – С.42-54.
- [Святкина и др., 2014] Святкина, М.Н. Системы приобретения знаний третьего поколения на основе когнитивных измерений/ М.Н.Святкина, В.Б.Тарасов // Труды 14-й национальной конференции по искусственному интеллекту с международным участием (КИИ-2014, Казань, 24-27 сентября 2014г.). Т.3. – Казань: Изд-во РИЦ «Школа», 2014. – С. 58-67.
- [Смирнова, 1986] Смирнова, Е.Д. Логическая семантика и философские основания логики/Е.Д.Смирнова. – М.: МГУ, 1986.
- [Тарасов, 2004] Тарасов, В.Б. От алгебраической модели Тарского к логико-семиотической матрице/ В.Б. Тарасов // Труды 9-й национальной конференции по искусственному интеллекту КИИ-2004 (Тверь, 28 сентября-2 октября 2004 г.). – М.: Физматлит, 2004. – Т.1. – С.311-322.
- [Тарасов, 2009] Тарасов, В.Б. От монологических к диалогическим подходам в искусственном интеллекте/ В.Б. Тарасов // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов V-й Международной научно-практической конференции (Коломна, 28-30 мая 2009 г.). – М.: Физматлит, 2009 – Т.1. – С.149-162.
- [Тарасов, 2013] Тарасов, В.Б. Гранулярные структуры измерений в интеллектуальных средах: васьевские и белнаповские сенсоры и модели их взаимодействия/ В.Б. Тарасов // Информационно-измерительные и управляющие системы. – 2013. – №5. – С.65-74.
- [Тарасов, 2014] Тарасов, В.Б. О методах построения гранулярных логических значений и структур / В.Б. Тарасов // Нечеткие системы и мягкие вычисления. Труды VI-й Всероссийской научно-практической конференции (Санкт-Петербург, СПИИРАН, 27-29 июня 2014 г.). Т.2. – СПб: Политехника-сервис, 2014. – С.33-44.
- [Тарасов, 2015] Тарасов, В.Б. Проблема понимания: настоящее и будущее искусственного интеллекта/ В.Б. Тарасов // Открытые семантические технологии проектирования интеллектуальных систем. Материалы V-й Международной научно-технической конференции (OSTIS-2015, Минск, БГУИР, 19-21 февраля 2015 г.). – Минск: Изд-во БГУИР, 2015. – С.25-42.
- [Тарасов и др., 2010] Тарасов, В.Б. Нечеткие лингвистические модели предпочтений когнитивных агентов/ В.Б. Тарасов, А.П. Калущкая// Труды международного конгресса по интеллектуальным системам и информационным технологиям (IS-IT'2010, Дивногорское, 2-9 сентября 2010 г.). – М.: Физматлит, 2010. – Т.2. – С.277-284.
- [Тарасов и др., 2012a] Тарасов, В.Б. Гранулярные, нечеткие и лингвистические онтологии для обеспечения взаимопонимания между когнитивными агентами/ В.Б.Тарасов, А.П.Калущкая, М.В.Святкина// Открытые семантические технологии проектирования интеллектуальных систем. Материалы II-й Международной научно-технической конференции (Минск, БГУИР, 16-18 февраля 2012 г.). – Минск: Изд-во БГУИР, 2012.
- [Тарасов и др., 2012б] Тарасов, В.Б. Интеллектуальная среда: когнитивно-регулятивный мета-агент на основе сенсорных сетей/ В.Б. Тарасов, А.П. Калущкая, М.Н. Святкина // Интеллектуальный анализ информации. Сборник трудов XII-й международной конференции им. Т.А.Таран (ИАИ-2012, Киев, 16-18 мая 2012 г.). – Киев: Просвіта, 2012. – С.203-214.
- [Тарасов и др., 2012в] Тарасов, В.Б. Логическая прагматика и анализ сенсорных данных: белнаповские сенсоры и их взаимодействия / В.Б.Тарасов, М.Н.Святкина // Труды международного конгресса по интеллектуальным системам и информационным технологиям (IS-IT'12, Дивногорское, Россия, 2-9 сентября 2012 г.). – М.: Физматлит, 2012. – Т.1. – С.458-467.
- [Тарасов и др., 2013] Тарасов, В.Б. Интеллектуальные системы на основе когнитивных измерений/ В.Б. Тарасов, М.Н. Святкина // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VII-й международной научно-практической конференции (ИММВ-2013, Коломна, 20-22 мая 2013 г.). – М.: Физматлит, 2013. – Т.2. – С.611-623.

[Тарасов и др., 2014] Тарасов, В.Б. Логическая прагматика в когнитивных измерениях / В.Б. Тарасов, М.Н. Святкина // Нечеткие системы и мягкие вычисления. Труды VI-й Всероссийской научно-практической конференции (Санкт-Петербург, СПИИРАН, 27-29 июня 2014 г.). Т.1. – СПб: Политехника-сервис, 2014. – С.155-168.

[Тарский, 1999] Тарский, А. Понятие истины в языках дедуктивных наук. Философия и логика Львовско-Варшавской школы. / А. Тарский; – М.: РОССПЭН, 1999. – С.19-155.

[Финн, 2004] Финн, В.К. Об интеллектуальном анализе данных/ В.К. Финн // Новости искусственного интеллекта. – 2004. – №3. – С. 3-18.

[Финн, 2010] Финн, В.К. О логико-семиотических проблемах теории понимания текстов/ В.К. Финн // Научно-техническая информация. Серия 2. Информационные процессы и системы. – 2010. – №9. – С.1-11.

[Финн, 2011] Финн, В.К. Искусственный интеллект: методология, применения, философия. / В.К. Финн; – М.: КРАСАНД, 2011.

[Философия, 2004] Философия: Энциклопедический словарь/ Под ред. А.А.Ивина – М.: Гардарики, 2004.

[Фреге, 2012] Фреге, Г. Логика и логическая семантика: Пер. с нем. Изд.2-е, испр./ Г. Фреге – М.: Книжный дом «ЛИБРОКОМ», 2012.

[Шабельников и др., 2009] Шабельников, А.Н. Интеллектуальные системы распределенного мониторинга на основе беспроводных сенсорных сетей с использованием системы мобильных объектов/ А.Н. Шабельников, В.А. Шабельников, С.М. Ковалев// Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов V-й Международной научно-технической конференции (Коломна, 28-30 мая 2009 г.). В 2-х томах. Т.1. – М.: Физматлит, 2009. – С. 538-543.

[Шрамко, 2002] Шрамко, Я.В. Обобщенные истинностные значения: решетки и мультирешетки / Я.В. Шрамко// Логические исследования. Вып.9. – М.: Наука, 2002. – С.264-291.

[Шрамко, 2009] Шрамко, Я.В. Истина и ложь: что такое истинностные значения и для чего они нужны? / Я.В. Шрамко // Логос. – 2009. – №2(70). – С.96-121.

[Ajdukiewicz, 1974] Ajdukiewicz, K. Pragmatic Logic. / K. Ajdukiewicz; – Warsaw: D. Reidel Publishing Company, 1974.

[Belnap, 1977] Belnap, N. A Useful Four-Valued Logic / N. Belnap// Modern Uses of Multiple-Valued Logic/ Ed. by J.M.Dunn and G.Epstein. – Dordrecht: D.Reidel, 1977. – P.8-37.

[Dunn, 1976] Dunn, J.M. An Intuitive Semantics for First Degree Entailment and Coupled Trees / J.M. Dunn// Philosophical Studies. – 1976. – Vol.29. – P.149-168.

[Dunn et al., 2001] Dunn, J.M. Algebraic Methods in Philosophical Logic/ J.M. Dunn, G. Hardegree; – Oxford: Oxford University Press, 2001.

[Fitting, 1989] Fitting, M. Bilattices and the Theory of Truth / M.Fitting//Journal of Philosophical Logic. – 1989. – Vol.18. – P.225-256.

[Fitting, 1991] Fitting, M. Kleene's Logic, Generalized/ M. Fitting// Journal of Logic Computation. – 1991. – Vol.1. – P.797-810.

[Ginsberg, 1988] Ginsberg, M. Multivalued Logics: A Uniform Approach to Reasoning in Artificial Intelligence/ M. Ginsberg //Computational Intelligence. – 1988. – Vol.4. – P. 265-316.

[Gottwald, 2001] Gottwald, S. A Treatise on Many-Valued Logics. / S. Gottwald; – Baldock: Research Studies Press, 2001.

[Hajek, 1998] Hajek, P. Metamathematics of Fuzzy Logics. / P. Hajek; – Dordrecht: Kluwer Academic Publishers, 1997.

[Moore et al., 1973] Moore, J. How Can MERLIN Understand? / J. Moore, A. Newell // Knowledge and Cognition. – Baltimore: Lawrence Erlbaum Associates, 1973.

[Simon, 1977] Simon, H. Artificial Intelligence Systems that Understand / H. Simon // Proceedings of the 5th International Joint Conference on Artificial Intelligence. – Boston: MIT, 1977. – P.1059-1073.

[Shramko et al., 2012] Shramko, Y. Truth and Falsehood – An Inquiry into Generalized Logical Values./ Y. Shramko, H. Wansing. – Heidelberg: Springer, 2012.

[Svyatkina et al., 2013] Svyatkina, M.N. On Granules and Granular Structures for Multi-Valued Logics / M.N. Svyatkina, V.B. Tarassov // Handbook of the 4th World Congress and School on Universal Logics (UNILOG'2013, Rio de Janeiro, Brazil, March 29-April 07, 2013)/ Ed. by J.-Y. Beziau et al. – Rio de Janeiro: Editora Kelps, Goiania, 2013. – P.297-300.

[Tarassov et al., 2013a] Tarassov, V.B. On Soft Measurements and Data Mining Based on Granular Pragmatics, Multi-Valued and Fuzzy Logics / V.B. Tarassov, M.N. Svyatkina// Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting/ Ed. by W. Pedrycz, M.Z. Reformat (IFSA/NAFIPS' 2013, Edmonton, Canada, June 24- 28, 2013). – Edmonton: IEEE, 2013. – P. 968-973.

[Tarassov et al., 2013b] Tarassov, V.B. Cognitive Measurements for Intelligent Systems/ V.B.Tarassov, M.N.Svyatkina // Proceedings of the 7th International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control (ICSCW'2013, Izmir, Turkey, September 2-3, 2013)/ Ed. by R.A. Aliev, K.W.Bonfig, M.Jamshidi, I.B.Turksen.– Kaufering: b-Quadrat Verlag, 2013. – P.189-202.

[Tarassov et al., 2014] Tarassov, V.B. Using Zadeh's Granulation Concept: Granular Logics and Their Application to Sensor Data Analysis/ V.B. Tarassov, M.N. Svyatkina // Recent Developments and New Directions in Soft Computing. – New York: SpringerVerlag, 2014. – P.453-466.

[Winograd et al., 1986] Winograd, T. Understanding Computers and Cognition: a New Foundation for Design/ T. Winograd, F. Flores.– Norwood: Ablex, 1986.

[Zadeh, 1997] Zadeh, L.A. Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic/ L.A.Zadeh// Fuzzy Sets and Systems. –1997. – Vol.90. – P.111-127.

LOGICAL-ALGEBRAIC METHODS IN CONSTRUCTING COGNITIVE SENSORS

Svyatkina M.N., Tarassov V.B.

Bauman Moscow State Technical University, Moscow, Russia

maria.svyatkina@gmail.com
vbulbov@yahoo.com

The main goal of the paper is the development of open pragmatic technologies for intelligent systems. To show some basic ideas of this approach a complex problem of monitoring railway infrastructure objects is faced. A core component of intelligent monitoring system is a hybrid knowledge acquisition/ discovery subsystem that includes sensor minid facilities, expert knowledge elicitation and ontological engineering tools. Various measurements play a leading part in performing monitoring tasks. Therefore, some particular features of measurement as a cognitive process are considered. Here measurement is tightly connected with evaluation and reasoning procedures. The earlier introduced concept of cognitive measurement is discussed in detail. Cognitive measurement is viewed as a hierarchical information granulation process performed with using cognitive sensors. A cognitive sensor is conceived as both information-measurement and understanding device able to judge about quantitative measurement results and reason for the sake of object's diagnostics and its future state prognosis. The operation of cognitive sensor is based on logical pragmatics of measurement. In this context some definitions, concepts and formal models of understanding are discussed. A general logical-algebraic approach to constructing understanding models on the basis of logical values and norms is proposed. In this context various logical systems are represented in an algebraic way by chains, lattices, semi-lattices and product lattices with employing different order relations. A keystone of our approach consists in merging formal specification and visual representation of logical world concepts. Therefore, some new pragmatic representations of logical values based on colour metaphor and geometric patterns are introduced. These representations implementing the ideas of both illustrative and cognitive graphics into logics underly the concepts of specific cognitive sensors – Vasiliev's sensor, Kleene's sensor, Belnap's sensor, as well as bilattice- and multilattice-based cognitive sensor networks. As a result, a general logical-algebraic methodology of synthesizing cognitive sensors has been developed and the appropriate algorithm has been presented.

Keywords: Artificial Intelligence, Intelligent Monitoring System, Knowledge Acquisition, Sensor Data Mining and Knowledge Discovery, Multi-Valued Logics, Cognitive Measurements, Information Granulation, Pragmatics, Understanding, Cognitive Sensor, Comprehension, Interpretation, Cognitive Graphics.



УДК 004.822:514

ИЛЛЮЗИИ ВОСПРИЯТИЯ И ПРЕДСТАВЛЕНИЯ: ИНФОРМАЦИОННЫЙ ПОДХОД

Фоминых И.Б.

Национальный исследовательский университет «МЭИ», Москва, Россия

igborfomin@mail.ru

Рассматриваются роль иллюзий в познании мира и формировании адекватного поведения живых систем. Особая роль отводится иллюзиям ассимиляции и контраста. На основе информационного подхода выявляется природа этих иллюзий. Приводятся примеры иллюзий.

Ключевые слова: роль иллюзий; иллюзии ассимиляции и контраста; информационный подход.

Введение

Иллюзии - это пример создания новой информации субъектом - и в этом отношении они являются переходной ступенью к творчеству. Они возникают, когда часть признаков стимула не задана или задана недостаточно определенно (например, мала интенсивность стимула) и субъект имеет возможность выбирать значение признака свободно в соответствии со сложившимся представлением. Это представление и вызывает искажение недостаточно жестко заданных признаков, причем искажения, двух родов: ассимиляция и контраст. Если задать признаки более жестко (например, сконцентрировав на них внимание), то иллюзия часто уменьшается или исчезает.

Будем рассматривать природу иллюзий в рамках информационного подхода, одним из ключевых результатов которого является сформулированный Г.А. Голицыным принцип максимума взаимной информации [Golitsyn G. and other, 1995], [Голицын Г.А. и др., 1996]. В соответствии с этим подходом для образа характерно холмообразное распределение вероятности, а для понятия - равномерное.

1. Иллюзии восприятия

Рассмотрим несколько примеров иллюзий, показывающих, что последние мало зависят от механизмов восприятия. Исследователи, занимавшиеся восприятием, часто демонстрировали следующий эффектный опыт: опустите левую руку в холодную воду, правую в горячую. Дайте им привыкнуть к температуре

воды. После этого перенесите обе руки в воду средней температуры.левой руке она покажется теплой, правой прохладной.

Рассмотрим другой пример, приведенный на рисунке 1: один и тот же центральный круг кажется больше, когда он окружен маленькими кружками, чем тот же круг в окружении больших, или на рисунке 2: один и тот же серый цвет на черном фоне кажется светлее, чем на белом. Обратим внимание на то, что физиологические механизмы восприятия температуры, размеров и цвета совершенно различны, тогда как наблюдаемая закономерность одна и та же: видимое значение признака (температура, размер, цвет) отклоняется от действительного в сторону, **контрастную** к фону.

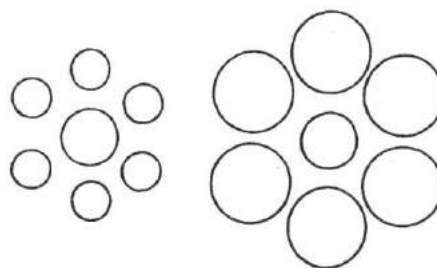


Рисунок 1- Иллюзия контрастного фона (размеры).

В дальнейшем будем рассматривать главным образом иллюзии зрения - ввиду их наглядности и разнообразия, но помнить, что иллюзии свойственны и другим видам восприятия, а также памяти и познанию вообще. Во всех этих случаях встречаются однотипные иллюзии, а это говорит о том, что они, видимо, обусловлены чисто информационными законами и не зависят от механизмов, реализующих эти законы.

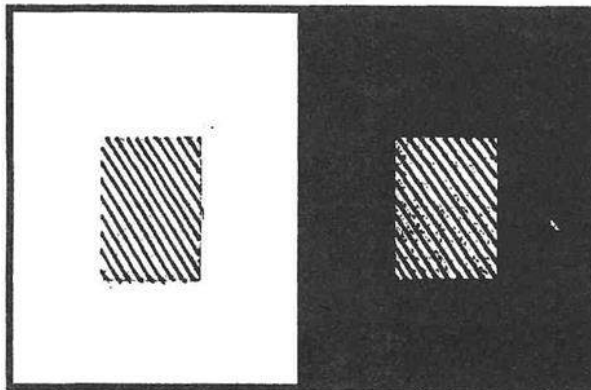


Рисунок 2 - Иллюзия контрастного фона (цвет).

Многие исследователи согласны с тем, что существуют, по-видимому, только два независимых вида иллюзий - *ассимиляция* и *контраст*. Все остальные являются либо их следствиями, либо их комбинациями. Чтобы пояснить, в чем сущность этих двух иллюзий, напомним классические опыты Д.Н.Узнадзе [Узнадзе Д.Н., 1961].

Испытуемому предъявляют два круга, из которых левый несколько больше правого (рис. 3а).

Эти круги являются установочным (обучающим) стимулом. Внимание испытуемого жестко фиксируется на этом стимуле в течение времени, достаточного, чтобы сложилось соответствующее представление. Затем испытуемого освобождают от установочного стимула и на короткое время ему предъявляют тестовый стимул - два одинаковых круга (3 б). Обычно эти круги ему не кажутся одинаковыми:

а) либо левый круг кажется больше правого - и тогда имеет место иллюзия ассимиляции (уподобления): тестовые круги как бы уподобляются установочным;

б) либо левый круг кажется меньше правого - и тогда имеет место иллюзия контраста.

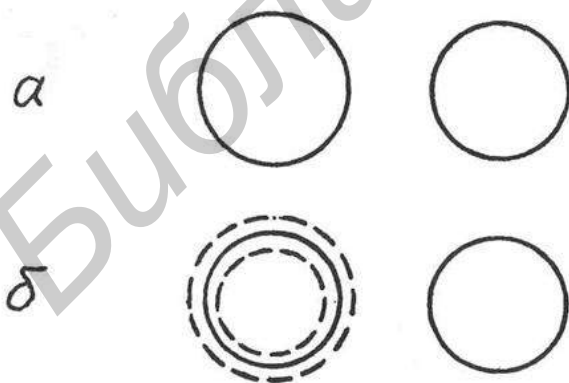


Рисунок 3. Установочные и тестовые стимулы.

Почему в одном случае возникает ассимиляция, в другом - при тех же, казалось бы, условиях - контраст? Ответ: на самом деле условия не те же. В зависимости от длительности и интенсивности установочного стимула и от индивидуальности

испытуемого, «холм» распределения вероятностей образа может иметь разную высоту и ширину.

Тестовый стимул может попадать в разные зоны вокруг вершины холма y^* , поэтому разным оказывается и результат. Согласно информационному подходу иллюзия - это изменение свободной переменной y в сторону максимальной информативности. Свобода в этом случае ограничена действующим (тестовым) стимулом x . Но все же этот стимул (из-за неизбежного шума) обычно задает не определенное значение y , а только распределение вероятностей, внутри которого y субъекта остается свобода выбора ощущения y . И он выбирает это ощущение оптимальным образом, т.е. сдвигает его (по отношению к стимулу x) в сторону большей информативности (рис. 4). Во второй фазе выгодно увеличивать вероятность, а для этого ощущение y сдвигается к вершине холма, т.е. к установке y^* . Так возникает иллюзия ассимиляции. Формальное выражение для нее можно записать так:

$$(y - \bar{y}) = \frac{(y^* - y)}{k} \quad (1)$$

Здесь $k > 1$ - некоторый «коэффициент ослабления». Выражение означает, что ощущение y сдвинуто относительно тестового значения y в ту же сторону, что и установка \bar{y} относительно y , - но на величину в k раз меньшую.

В первой фазе, напротив, выгодно уменьшать вероятность - и ощущение y двигается в сторону, обратную установке y^* . Так возникает иллюзия контраста:

$$(y - \bar{y}) = \frac{(y^* - y)}{k} \quad (2)$$

Таким образом, зона вокруг установки является зоной ассимиляции, тогда как периферийная зона является зоной контраста.

Пример иллюзии контраста - рис. 2. Здесь установочный стимул - цвет фона, тестовый стимул - цвет серого пятна. Представление о цвете тестового стимула y отклоняется от истинного значения y^* (серого) в сторону, обратную установке y^* , задаваемой фоном. На белом фоне это отклонение в сторону черного, на черном - в сторону белого. Возникает иллюзия именно контраста потому, что цвет серого пятна достаточно далеко отстоит от цвета фона, а сам фон достаточно однороден, т.е. дисперсия цвета фона мала и распределение вероятностей различных оттенков цвета является достаточно узким. При этом чем больше дисперсия тестового стимула, тем сильнее будет иллюзия контраста.

Можно представить себе другую ситуацию, когда фон является «пестрым» (большая дисперсия цвета), а серый цвет тестового пятна достаточно

близок к цвету фона. В этом случае пятно может потеряться на фоне, т.е. будет ассимилировано. Причем чем больше дисперсия тестового стимула, тем сильнее иллюзия ассимиляции. Именно такого рода иллюзия обычно используется для маскировки, когда пестро раскрашенный объект легко теряется на таком же пестром фоне.

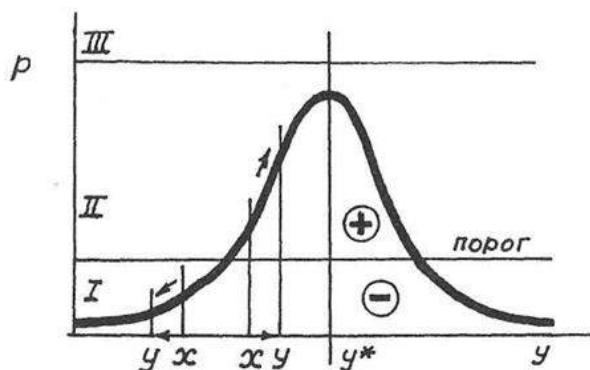


Рисунок 4 - Сдвиг по информативности

Подчеркнем еще раз роль неопределенности в возникновении иллюзии. Беглый взгляд, рассеянное внимание, краткая экспозиция, слабое освещение, высокий уровень шума, малая интенсивность тестового стимула x , и т.п. факторы увеличивают неопределенность порождаемого им ощущения y . Именно в таких условиях переменная y наиболее свободна и наблюдается наибольшая выраженность иллюзии. При внимательном взгляде, ярком освещении, длительной экспозиции тестового стимула неопределенность уменьшается или вовсе исчезает, - а имеете с ней и иллюзия: представление y жестко соответствует стимулу x .

Следует сказать, что ассимиляция и контраст - явления относительные и, зависят от того, по отношению к какой установке они рассматриваются. Контраст по отношению к белому может выглядеть как ассимиляция по отношению к черному. Например, если видимый серый цвет кажется темнее истинного, то это можно рассматривать и как контраст по отношению к белому, и в то же время как ассимиляцию серого цвета черным.

Пример: если после восприятия яркого белого цвета (установочный стимул) мы попадаем в темноту, то в первое время, обычно, ничего не видим. Это значит, что зона ассимиляции белого сузилась, а зона контраста (= ассимиляции черным) приблизилась вплотную к белому, так что практически все оттенки серого цвета (тестовый стимул), позволяющие различать предметы, кажутся нам одинаково черными, т.е. ассимилируются черным. Однако по мере адаптации к темноте зона ассимиляции черного сужается, зона контраста - расширяется. И все больше оттенков серого цвета (сначала наиболее светлые, а затем и все более

темные) постепенно попадают в зону контраста, отделяются от черного и начинают восприниматься.

На рис. 5 показано, как воспринимается острый угол. Субъективный образ угла искажается и напоминает рогатку. Среди искажений можно отметить как важнейшие следующие два:

1) вершина угла (точка слияния сторон) отступает внутрь угла;

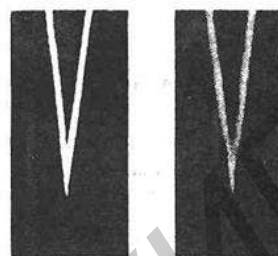


Рисунок 5 - Субъективное восприятие угла.

2) вследствие этого угол кажется в среднем больше, чем он есть на самом деле.

Насколько эти выводы подтверждаются экспериментом?

Последний эффект - преувеличение острого угла - хорошо известен и сам по себе, известно и то, что он служит причиной других, более сложных иллюзий [Артамонов И.Д., 1969], [Толанский С., 1967]. Некоторые из них представлены на рис. 6, 7. Особенно эффектно выглядит та, что показана на рис. 6а. Её можно продемонстрировать в динамике: сначала начертить несколько длинных параллельных линий, затем пересечь их косыми отрезками, как показано на рисунке. Параллельность исчезает. Отрезки стремятся «развернуть» линии в сторону увеличения острого угла между линиями и отрезками. В результате создается впечатление взаимного наклона линий.

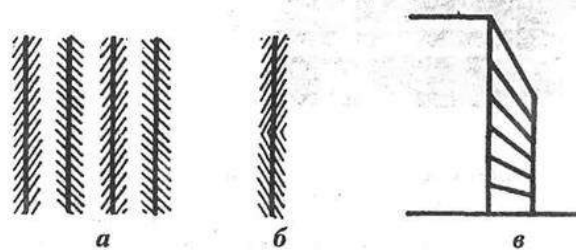


Рисунок 6. Иллюзия параллельности.

Интересна «архитектурная» иллюзия, схематически изображенная на рис. 6в. Иллюзия возникает, если вы рассматриваете сбоку здание с рядами горизонтальных ленточных окон (такие окна можно видеть, например, у здания Центра международной торговли на Краснопресненской набережной). Поскольку при взгляде сбоку окна образуют с вертикальными кромками здания острые углы, то кажется, что эти кромки не вертикальны, а «заваливаются» вправо. Эффект бывает настолько

сильным и неприятным, что для борьбы с ним архитекторы применяют специальные, довольно дорогостоящие, приемы, например, устанавливают на фасаде вертикальные пилоны.

Что касается «отступления вершины» угла, то хотя этот, эффект сам по себе менее известен, но из него также вытекает ряд более сложных и хорошо известных иллюзий. Такова, например, иллюзия Поггендорфа (рис. 7 а). Если попросить зрителя, не пользуясь линейкой, продолжить отрезок АВ по другую сторону вертикальной полосы, то он наверняка ошибется и нарисует в качестве продолжения что-нибудь вроде отрезка CD. Между тем истинное продолжение - отрезок EF в чем можно убедиться; приложив линейку. Тут оба эффекта - и отступление вершины, и преувеличение угла - действуют совместно и заставляют нас видеть продолжение ближе, чем оно лежит на самом деле. На рис. 7 б показан результат моделирования этой иллюзии. Видно, что на «субъективной картине» косые отрезки действительно смещены друг относительно друга и пересекают, полосу под большим углом, чем на самом деле.

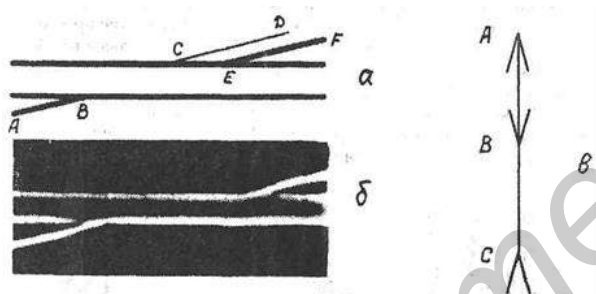


Рисунок 7 - Иллюзии продолжения отрезка и стрелы.

Аналогичным образом объясняется и иллюзия Мюллера Лайера («иллюзия стрелы»), одна из самых известных зрительных иллюзий. На рис. 7 в отрезок АВ кажется короче отрезка ВС, хотя на самом деле они равны. Причина — отступление вершин внутрь углов, из-за чего отрезок АВ преуменьшается, а отрезок ВС - преувеличивается.

Можно сказать, что иллюзия контраста порождает целое семейство производных иллюзий. Тоже можно утверждать относительно иллюзии ассимиляции. В частности, многочисленные факты, демонстрирующие влияние прошлого опыта, стереотипов и предвзятых представлений на восприятие, часто основаны на иллюзии ассимиляции. Вот ещё пример.

Вернемся к иллюзии Поггендорфа: если наклонная линия прерывается вертикальной полосой, то она кажется изломанной (рис. 2.7а). Но если полоса -- это колонна, а линия - ручка стоящей за ней метлы, то, как показывает эксперимент, иллюзия резко уменьшается, а то и вовсе исчезает. Причина в том, что мы *знаем*, что ручка метлы не

может быть изломанной, т.е. у нас есть определенное представление о ручке, и это **представление** ассимилирует (уподобляет себе) воспринимаемый стимул. Иллюзия ассимиляции уничтожает иллюзию контраста.

К иллюзиям ассимиляции следует отнести и феномен «проекции», когда в расплывчатых пятнах, неопределенных формах, которые сами по себе не образуют никаких осмысленных, знакомых по прошлому опыту, образов, субъект «видит» представления, которыми наполнена его память. При этом разные субъекты в одном и том же воспринимаемом материале видят разное - в зависимости от разницы в прошлом опыте. На этом основан известный тест Роршаха и аналогичные тесты, позволяющие вскрыть содержание памяти испытуемого, которое сам он не осознает.

2. Иллюзии представления

Но иллюзия ассимиляции не только позволяет видеть то, чего нет в действительности. Иногда она, напротив, позволяет обнаружить замаскированный объект, выделить слабый сигнал на фоне шума. Известно, например, что если ищешь какой-нибудь хорошо замаскированный объект (гриб в лесу, птицу на ветке, зайца «на загадочной картинке»), то представление об этом объекте помогает поиску, оно как бы заранее настраивает наше восприятие на нужный объект, подобно тому как радиоприемник настраивается на нужную волну. Представление снижает порог восприятия для нужного стимула.

Примером может служить влияние сложившейся гипотезы на восприятие фактов ученым. Отметим, что под гипотезой понимается частный случай представления, когда факты выступают в роли стимулов. Рассмотрим несколько типичных ситуаций.

Правильная гипотеза, направляя внимание на определенные стимулы, порождает эффект сенсбилизации и усиления этих стимулов и благодаря этому позволяет обнаружить даже слабый и зашумленный сигнал. В качестве примера можно привести открытие Нептуна: верная гипотеза позволила Леверрье проделать расчеты, которые сузили область поиска и направили внимание и телескопы астрономов на нужный участок неба. Без этого едва светящаяся звездочка еще долго не была бы замечена. Верная гипотеза не менее, чем телескоп и микроскоп, увеличивает остроту наших чувств, служит усилителем для слабых сигналов и позволяет выделить их на фоне шума.

Вот, что сказано об этом у Дж. Конрада: «Лоцман видит лучше, чем посторонний человек, ибо хорошо знает местность; это знание, словно более острое зрение, позволяет ему уловить очертание мелькнувшего предмета, проникнуть сквозь завесу тумана, опускающуюся над землей во

время шторма. Лощман точно определяет контуры берега, затянутого туманным покровом, находит опознавательные знаки, наполовину погребенные в беззвездной ночи, как в могиле. *Он узнает, ибо уже знает*» [Конрад Дж., 1959].

Напротив, отсутствие правильной гипотезы, наличие предрассудков и предвзятых представлений порой делает ученого буквально слепым, мешает увидеть очевидные, но необычные факты. Так, в 1933 г. немецкий физик Кунце наблюдал в камере Вильсона частицу в 200 раз тяжелее электрона. Это был μ -мезон. Однако Кунце не поверил в наблюдение, отнеся его к ошибкам опыта. Повторно μ -мезон был открыт в 1937 г. американским физиком К.Андерсеном - после того, как в 1935 г. Х.Юкава опубликовал свою гипотезу, предсказывающую существование мезона.

Еще пример: в конце XVIII века Парижская Академия наук отказалась рассматривать сообщения о якобы падающих с неба камнях, как «противоречащие научным представлениям». Так на сто лет были «закрыты» метеориты: наука упорно отказывалась их видеть. Одна из причин этой слепоты - отсутствие сколько-нибудь разумной гипотезы, способной объяснить их происхождение.

Еще более отрицательным может оказаться влияние ложной гипотезы, особенно если факты имеют неопределенный характер, допускают различную интерпретацию. Здесь также имеет место иллюзия ассимиляции, но ассимиляции истинных фактов ложной гипотезой. Ученый, всюду находящий факты, подтверждающие его гипотезу и не замечающий фактов, ее опровергающих, больной, одержимый манией преследования и на каждом шагу встречающий подтверждения своей правоты - все они являются жертвами иллюзии ассимиляции.

П.Дирак рассказывает, как ложная доминирующая идея долгое время мешала открытию позитрона и вела к неверному истолкованию наблюдаемых фактов: «Считалось, что в Природе существуют всего две основные частицы: электрон и протон. Их нужно было всего две, потому что есть всего два вида электрического заряда: отрицательный и положительный. Если есть одна частица для отрицательного заряда, одна для положительного, то вроде бы все в порядке: двух частиц достаточно. Никакие другие частицы не нужны. Эта идея тогда господствовала, Позитроны не наблюдались потому, что люди закрывали глаза на все свидетельства в пользу их существования» [Дирак П., 1983].

История географических открытий полна примеров того, как предвзятая ложная гипотеза ассимилировала, подчиняла себе наблюдаемые факты. Например, известно, что Колумб до самой

смерти был убежден, что им открыта Индия, хотя множество фактов говорило против этого. Ряд курьезных недоразумений был связан с гипотезой об огромном Южном материке, который якобы должен занимать большую часть южного полушария. Эта Terra Australis со времен античности сделалась прямо-таки навязчивой идеей многих географов и мореплавателей и породила немало забавных ошибок. Одну из них совершил Тасман, первооткрыватель Новой Зеландии. Вот как пишет об этом Э.Раквитц «Вывод, который сделал Тасман из своего путешествия, прямо-таки поражает своей нелепостью... Дело в том, что земляк Тасмана Ле Мер в 1616 году увидел перед крайней оконечностью Южной Америки неизвестную землю и назвал ее Землей Штатов. Ле Мер был глубоко убежден, что им открыт Южный материк, хотя при более внимательном наблюдении можно было без труда убедиться, что речь идет всего лишь об острове площадью 50 на 10 километров. Однако Тасман не видел причин усомниться в показаниях своего земляка... И когда Тасман узрел vzdымающиеся посреди бескрайнего океана скалы Новой Зеландии, он, равно как и вся его команда, сразу уверовал, что эти скалы в южной части Тихого океана имеют непосредственное отношение к той земле, которую нанес на свою карту Ле Мер. Традиционные представления о географии нашей планеты были настолько прочными, что люди, ни секунды не колеблясь, мысленно соединили два клочка суши, находившиеся друг от друга на расстоянии 10 тысяч километров, в единое целое» [Раквитц Э., 1969].

Тасман, в сущности, проделал то, что проделывает наше воображение при восприятии, например, когда оно соединяет разрозненные детали в целостный контур несуществующего треугольника.

Ассимиляция фактов предвзятой гипотезой — вещь настолько серьезная, что с ней приходится считаться во многих видах практической деятельности (особенно там, где она может привести к тяжелым последствиям) и принимать специальные меры для ее нейтрализации. Так, в судебной практике сложился веками отработанный принцип, согласно которому истолкованием известных следствию фактов должны заниматься, по крайней мере, две стороны - обвинение и защита, заинтересованные в подтверждении прямо противоположных гипотез: «виновен» и «невиновен». Без этой заинтересованности важные факты могут ускользнуть от внимания, будут неверно истолкованы и т.п.

Элементы подобной практики есть и в среде ученых в виде дискуссий, защиты диссертаций и проектов и пр. К сожалению, когда ученый остается один на один со своей гипотезой, ему становится трудно сохранить объективность. Ч.Дарвин отмечал,

например, что приходится особенно тщательно фиксировать факты, противоречащие его теории, ибо они имеют коварную способность ускользать от внимания и быстро забываться. Конечно, виноваты не сами факты - их вытесняет из сознания доминирующая установка.

Заключение

Таким образом, при анализе иллюзий речь идет не столько о познании, сколько о воображении. Воображаемый мир часто строится по контрасту с реальным, как нечто «иное», по многим признакам «противоположное» действительности. Мечта о «земле обетованной» - в противоположность опустылевшему египетскому плену, мечта о «свободной Америке» - в противоположность угнетению по эту сторону океана, гармоничные «дети природы» как контраст опустошенным пасынкам цивилизации, утопия о светлом будущем, в котором все сегодняшние минусы изменят знак на плюсы, и т.п.

Воображение очень сильно связано с процессом генерации идей, и здесь уже только один шаг до решения творческих задач. Напомним, что в информационном подходе под творческой задачей понимается задача, условия которой выглядят противоречивыми и несовместимыми, а под решением творческой задачи - разрешение противоречий [Фоминых И.Б., 2002], но важно, что оно не абсолютно и является следствием ограниченности исходного множества представлений субъекта. Для решения задачи необходимо выйти за эти пределы.

Работа выполнена при поддержке гранта РФФИ 15-07-02320 «Гибридные модели когнитивных агентов на основе активных темпоральных логик, исчисления образов и эмоциональных оценок».

Библиографический список

- [Golitsyn G. and other,1995] Golitsyn G., Petrov V. Information and Creation.- Basel: Birkhauser Verlag, 1995.
- [Голицын Г.А. и др., 1996] Голицын Г.А., Фоминых И.Б. Нейронные сети и экспертные системы: перспективы интеграций// Новости искусственного интеллекта. - М.: АИИ, №4, 1996.
- [Узнадзе Д.Н., 1961] Узнадзе Д.Н. Экспериментальные основы психологии установки. Тбилиси: Изд-во АН Груз. ССР, 1961, - 210с.
- [Артамонов И.Д., 1969] Артамонов И.Д. Иллюзии зрения. - М.: Наука, 1969. - 224с.
- [Толаиский С., 1967] Толаиский С. Оптические иллюзии. - М.: Мир, 1967. - 128с.
- [Конрад Дж., 1959] Конрад Дж. Конец рабства. // Конрад Дж. Избранное в 2-х томах, т. 1. - М.: Гос. изд-во художественной литературы, 1959. - 590с.
- [Дирак П., 1983] Пути физики. - М.: Энергоатомиздат, 1983. - 88с.
- [Раквитц Э., 1969]. Раквитц Э. Чужеземные тропы, незнакомые моря. - М.: Молодая гвардия, 1969.

[Фоминых И.Б.,2002] Фоминых И.Б. О технологии решения творческих задач// В сб. трудов 8 Национ. Конф. по искусств. интеллекту "КИИ-2002", т.1,М.:изд.физ.-мат.литературы,2002.

ILLUSIONS OF PERCEPTION AND REPRESENTATION: INFORMATION APPROACH

Fominykh I.B.

National Research University «MEI» Moscow, Russia

igborfomin@mail.ru

We consider the role of illusion in the knowledge of the world and the formation of appropriate behavior of living systems. A special role is played by the illusions of assimilation and contrast. On the basis of the information approach reveals the nature of these illusions. Examples of illusions.

Introduction

Illusions arise when the signs of the stimulus is given or if ill-defined and the subject is able to choose freely characteristic value in accordance with the prevailing view.

We shall consider the nature of the illusions in the framework of the informational approach, one of the key results of which is formulated G.A. Golitsyn principle of maximum mutual information . In accordance with this approach for the image characteristic undulating probability distribution, and for the concept - uniform.

Main Part

Many researchers agree that there are apparently only two independent kinds of illusions - assimilation and contrast. All others are either consequences of them, or combinations thereof.

To clarify the nature of the illusions of perception are considered classic D.Uznadze experiments and their interpretation in terms of the information approach. Then we discuss the illusion of subjective perception of an acute angle, parallelism, projections and illusions of the boom.

The illusion of assimilation not only allows you to see what is not there in reality. Sometimes it is can detect camouflaged objects, highlight a weak signal in noise. Following is the illusion of representation, including the assimilation of the true facts of false hypothesis

Conclusion

The analysis of the illusions it is not so much about cognition, but about imagination. Imagination is very much related to the process of generating ideas, and here only one step to solve creative problems. Recall that in the information under the creative approach the task is understood the task, the terms of which appear contradictory and inconsistent, and under the decision of creative problem - resolution of conflicts.



УДК 004.92

ОНТОЛОГИЧЕСКИЙ ПОДХОД К ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Коршунов С.А.*

* *Институт динамики систем и теории управления имени В.М. Матросова СО РАН,
г. Иркутск, Россия*
grey.for@gmail.com

В работе рассмотрен подход к визуализации результатов имитационного моделирования на основе онтологий и концепция программного средства, реализующего этот подход. Разрабатываемое средство позволит повысить эффективность процесса визуализации результатов имитационного моделирования, автоматизировав построение визуальных моделей. Также описана архитектура предлагаемого программного средства, аспекты его программной реализации, предлагаемые модели онтологий, приведены примеры визуальных сцен, созданных при помощи предлагаемого средства.

Ключевые слова: онтология, визуализация, имитационное моделирование, машинная графика.

Введение

Проблема визуализации результатов имитационного моделирования является достаточно актуальной на данный момент, что обусловлено широким применением данного подхода в самых различных областях. При этом, визуализация результатов имитационного моделирования традиционно считается второстепенной задачей при исследовании и моделировании систем различной природы. Подобное отношение обуславливает недостаточное теоретическое и практическое развитие данной области.

В частности, современные средства визуализации результатов имитационного моделирования являются либо «встроенными» системами и обеспечивают низкую когнитивность [Зенкин, 1991] (свойство, характеризующее насколько полно визуальные модели отражают основные свойства реальных объектов-прототипов) создаваемых пространственно-временных сцен (на уровне пикселей и абстрактных графовых структур), используя простейшие универсальные алгоритмы и математические выражения, либо «внешними» специализированными программными комплексами, требующими навыков профессиональных дизайнеров и использующих сложные математические и физические абстракции.

При этом отсутствуют системы визуализации, использующие унифицированный подход, например, на основе онтологического моделирования, и обеспечивающие синтез

(автоматическое создание) описания пространственно-временных сцен.

Создание графических объектов, соответствующих определенной предметной области, является довольно трудоемкой задачей, для решения которой исследователи в своих работах чаще всего используют различные графические пакеты (3DSMax, Blender, Maya и др.). При этом выбор определенного пакета обусловлен опытом работы исследователя с выбранным программным обеспечением.

Стоит отметить, что графические пакеты являются довольно сложными системами и в том случае, если опыт работы с ними отсутствует, время, затраченное на их освоение, может существенно замедлить процесс исследовательской работы. Из этого можно сделать вывод, что использование стороннего программного обеспечения является достаточно неэффективным способом визуализации.

Наиболее перспективное направление решения данной проблемы – автоматизация исследований при помощи создания пространственно-временных сцен на основе онтологии.

1. Основные принципы разработки программного средства визуализации

Предлагаемое программное средство призвано учесть недостатки существующих систем и обеспечить:

- возможность создания сложных визуальных объектов на основе геометрических примитивов;
- отображение динамики поведения визуальных моделей (анимация, динамическое изменение текстур);
- использование принципов порождающего программирования для синтеза программного кода, описывающего визуальные объекты и пространственно-временные сцены;
- веб-интерфейс создания визуальных объектов и пространственно-временных сцен.
- независимость от имитационной модели;
- возможность адаптации к предметной области;
- отчуждаемость программного кода.
- отсутствие необходимости установки дополнительного программного обеспечения (библиотек и т.д.);

С точки зрения визуализации, степень адаптации к предметной области определяется способностью отобразить все необходимые объекты исследуемой области, что невозможно при использовании ограниченных библиотек визуальных объектов. В этих условиях возникает проблема создания визуальных объектов для исследуемой области.

Наиболее перспективным направлением решения этой проблемы является их создание на основе онтологического моделирования. При таком подходе предлагается инкапсулировать свойства и поведение созданных объектов в объектах онтологии, создав иерархию онтологий (рис. 1).

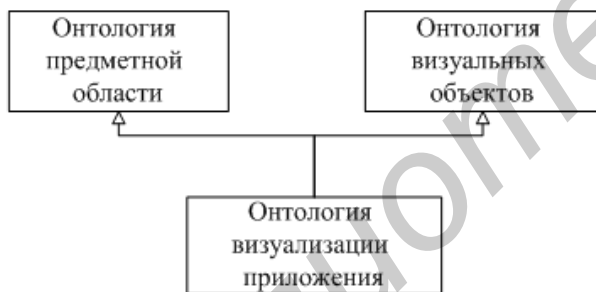


Рисунок 1 -- Иерархия онтологий

При таком подходе хранение онтологий предлагается осуществить с помощью базы данных, что исключает обработку семантически незначимой информацией (служебные теги XML), а также обеспечивает высокое быстродействие и поддержку целостности данных. При этом обеспечивается возможность импорта/экспорта иерархии понятий, их свойств и отношений между ними в форматах OWL и RDF.

Дополнительную сложность решения задачи визуализации придает необходимость моделирования поведения визуализируемых объектов. Данная проблема может быть решена путем императивного описания их поведения в онтологии и его реализации с помощью продукционной модели представления знаний.

Объектно-ориентированная модель некоторой предметной области D , т.е. онтология предметной области Ont^D , включает предметные сущности (классы и экземпляры классов) и отношения между ними:

$$Ont^D = \langle BT_D, CN, Pr, Obj, R^D \rangle, \quad (1)$$

где Ont^D – онтология предметной области, BT_D – перечень базовых типов данных, $BT_D = \{ \text{литерал, объект, коллекция} \}$, CN – имена классов, Pr – имена свойств классов, Obj – понятия (константы, объекты) предметной области, R^D – конечное множество отношений между концептами заданной предметной области,

$$R^D = \{ R_{isa}^D, R_{pr}^D, R_c^D \}, \quad (2)$$

при этом R_{isa}^D – отношение наследования между классами CN , R_{pr}^D – отношение между классами и свойствами, $pr \in R_{pr}^D \langle cn, cn_bt \rangle$, где $pr \in Pr$, $cn \in CN$, $cn_bt \in CN \cup BT_D$, последнее означает, что в предметной области D свойство с именем pr_i характеризует класс cn_j , а значениями свойства могут быть элементы типа другого класса из CN или основного множества типов BT_D , R_c^D – причинно-следственные отношения между концептами.

Онтология визуальных объектов включает классы и экземпляры понятий, описывающие все доступные визуальные объекты и визуальные эффекты:

$$Ont^{VO} = \langle BT_VP, Pr^{VP}, Obj^{VP}, R_{pr}^{VP}, T_VE, Pr^{VE}, Obj^{VE}, R_{pr}^{VE} \rangle, \quad (3)$$

где Ont^{VO} – онтология визуальных объектов, BT_VP – базовые типы визуальных примитивов, Pr^{VP} – свойства визуальных примитивов, Obj^{VP} – собственно визуальные примитивы, R_{pr}^{VP} – отношения между базовыми типами и свойствами визуальных примитивов, Pr^{VE} – свойства визуальных эффектов, T_VE – типы визуальных эффектов, Obj^{VE} – визуальные эффекты, R_{pr}^{VE} – отношения между типами и свойствами визуальных эффектов.

$$Ont^{AppV} = \langle Obj^{AppV}, Rule_{vs} \rangle, \quad (4)$$

где Ont^{AppV} – онтология визуализации приложения, Obj^{AppV} – визуальные объекты,

$$Obj^{AppV} = \langle Obj^D, Obj^{VP}, Obj^{VE} \rangle, \quad (5)$$

где $Rule_{vs}$ – правила, описывающие пространственное расположение объектов на сцене приложения и их поведение, представлены в виде цепочки логических правил «если, то», где в качестве условий будут выступать переменные входных данных имитационной модели, а в качестве действий на условие – изменение определенных визуальных объектов.

При отображении онтологии визуальных объектов и онтологии приложения на модель хранения данных получаем ее структуру (рис 2).

Согласно данной модели, каждому объекту предметной области будет соответствовать визуальный объект, наиболее полно отображающий все его основные свойства. В свою очередь, каждый визуальный объект может поддерживать несколько различных визуальных эффектов (анимаций), которые могут изменять определенные свойства объекта. Каждый экземпляр анимации содержит ссылку на ее тип и на атрибуты/свойства объекта, которые будут изменены в процессе анимации.

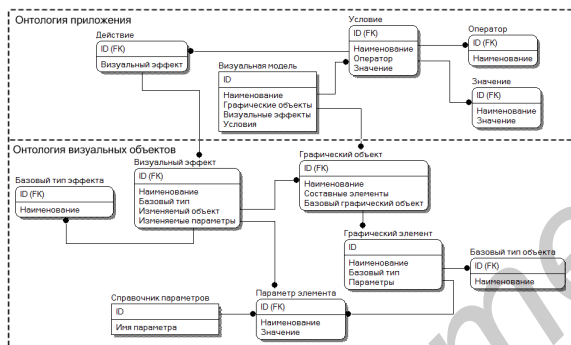


Рисунок 2 -- Логическая модель онтологий

На основе предложенной онтологической модели необходимо разработать отображения, позволяющие синтезировать программный код визуализации, в частности:

$$R_K^{VP} : Obj^{VP} \rightarrow K^{VP}, \quad (6)$$

где K^{VP} – программный код примитива, $R_K^{VE} : Obj^{VE} \rightarrow K^{VE}$, K^{VE} – программный код визуального эффекта,

$$R_K^{Rule} : Rule_{vs} \rightarrow K^{Rule}, \quad (7)$$

где K^{Rule} – программный код построения визуальной модели, а также заключительное отображение

$$R^K : K^{VP} \otimes K^{VE} \otimes K^{Rule} \rightarrow K^{AppV}, \quad (8)$$

описывающее динамическую визуальную модель предметной области, где K^{AppV} – программный код, \otimes – обобщенная операция композиции программного кода.

Задача состоит в том, чтобы разработать алгоритмы, реализующие перечисленные отображения: R_K^{VP} , R_K^{VE} , R_K^{Rule} , R^K .

2. Архитектура программного средства.

В соответствии с представленными основными принципами разработки сформирована общая концепция архитектуры программного средства (рис. 3). Программное средство состоит из нескольких компонентов, которые можно разделить по принадлежности к серверной или клиентской частям архитектуры.

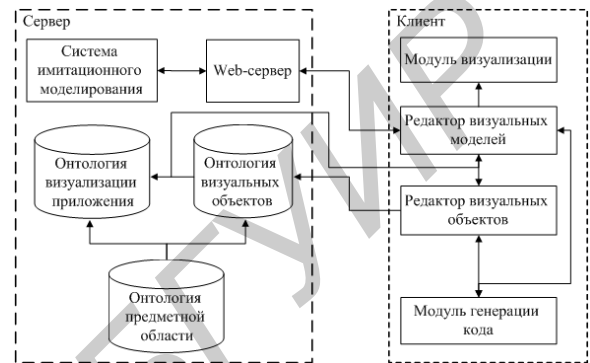


Рис.3. Общая архитектура программного средства.

Клиентские компоненты:

- Редактор визуальных объектов позволяет пользователю создавать свои визуальные объекты разной степени сложности, используя в качестве основы геометрические примитивы (прямоугольник, сфера, цилиндр и т.д.) (рис.3).
- Редактор визуальных моделей для описания структуры и поведения пространственно-временных сцен в виде правил [Коршунов и др., 2015].
- Модуль визуализации позволяет на основе сформированного описания сгенерировать трехмерную пространственно-временную сцену, соответствующую моделируемой области и содержащую визуальные объекты, созданные пользователем. В качестве средства для отображения трехмерной сцены предлагается использовать браузер.
- Модуль генерации JavaScript-кода пространственно-временной сцены и ее объектов отображаемый при помощи графического API WebGL [WebGL].

Серверные компоненты:

- Имитационная модель – источник входных данных для визуализации. В частности, предполагается апробация программного средства при взаимодействии с системой имитационного моделирования на основе агентного подхода [Николайчук и др., 2010].
- Сервер обмена данными между имитационной моделью и модулем визуализации.
- Онтология визуальных объектов, содержащая созданные пользователем визуальные примитивы, которые впоследствии можно

использовать в качестве основы для новых объектов, дополняя или изменяя их структуру.

- Онтология визуализации приложения, содержащая конкретные объекты, предназначенные для моделирования исследуемой области, которые будут находиться в составе визуальной модели.
- Онтология предметной области, содержащая классы и понятия исследуемой области.
- Сгенерированный код будет отчуждаем и может отображаться вне программного средства при помощи любого браузера.

3. Предлагаемые методы реализации компонентов

Программное средство предлагается реализовать в виде веб-приложения, предоставив пользователю доступ к нему через браузер. Главное преимущество такого подхода – отсутствие затрат и дополнительных сложностей, связанных с установкой, обновлением и поддержкой программного средства.

Классическая клиент-серверная архитектура обладает существенным недостатком – обмен данными инициируется по запросу клиента. Данный недостаток можно устранить, используя протокол WebSocket, который осуществляет асинхронный обмен сообщениями между браузером и веб-сервером в режиме реального времени и дает возможность инициировать передачу данных сервером [WebSocket]. Также, клиентская реализация протокола позволит интегрировать программное средство визуализации в систему имитационного моделирования.

В качестве основы для модуля визуализации и графического редактора предлагается использовать графический API WebGL, позволяющую создавать на JavaScript интерактивную графику. Как и WebSocket-протокол, WebGL поддерживается всеми основными браузерами, что позволяет реализовать клиентскую часть в виде веб-страницы.

Заключение

Авторами предложен подход к автоматизации процесса визуализации результатов имитационного моделирования при помощи генерации кода визуальных моделей на основе онтологического описания их структуры и поведения, который позволит пользователям, не имеющим высокой программистской квалификации визуализировать исследуемую область, не прибегая к помощи сторонних графических редакторов, требующих значительных временных затрат на освоение. Предложены возможные технологии реализации компонентов, для нужд программного средства были разработаны модели ряда онтологий (предметной области, графических объектов и визуализации приложения).

Реализация такого подхода в программном средстве позволит создавать подсистемы

визуализации, которые в дальнейшем могут быть интегрированы в имитационные системы.

Результаты, представленные в статье, получены при частичной финансовой поддержке РФФИ, проекты 14-07-05641 «Разработка принципов, моделей и методов создания и поддержки интеллектуальных мультиагентных систем для прогнозирования техногенных чрезвычайных ситуаций», 16-37-00122 «Модели, методы и средства синтеза императивного описания пространственно-временных сцен на основе онтологий и порождающего программирования», 15-37-20655 «Разработка моделей, методов и средств сервисно-ориентированной технологии синтеза баз знаний производственных экспертных систем на основе трансформации концептуальных моделей».

Библиографический список

- [Зенкин, 1991] Когнитивная компьютерная графика. Под ред. Д.А. Поспелова. М.: Наука, 1991. – 192 с.
- [Коршунов и др., 2015] Коршунов С.А., Николайчук О.А., Павлов А.И. Web-ориентированный компонент производственной экспертной системы // Программные продукты и системы. – 2015. – №2. – С.20-25.
- [WebGL] Документация спецификации Web-based Graphics Library (WebGL). 2015. URL: <https://www.khronos.org/registry/webgl/specs/latest/1.0/> (дата обращения: 22.06.2015).
- [Николайчук и др., 2010] Николайчук О.А., Павлов А.И., Юрин А.Ю. Система имитационного моделирования динамики состояний сложных технических систем на основе агентного подхода // Автоматизация в промышленности. – 2010, №7. – С.44-48.
- [WebSocket] Документация спецификации WebSocket. 2015. URL: <https://tools.ietf.org/html/rfc6455> (дата обращения: 22.06.2015).

ONTOLOGY-BASED APPROACH FOR VISUALIZATION OF SIMULATION RESULTS

Korshunov S.A. *

** Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of the Russian Academy of Sciences (IDSTU SB RAS), Irkutsk, Russia*

grey.for@gmail.com

The paper describes the approach of graphical objects creation based on domain ontology description, as well as the hierarchy of the ontology, which includes domain ontology, graphical objects ontology and application's visualization ontology. Concept of software is described, that uses this approach for simulation visualization. The basic requirements for software are formulated, the implementation of which eliminates the most disadvantages of existing imaging systems.

The main components of proposed software architecture considered to be: the simulation model as an input source; graphics editor that lets to create graphical objects based on geometric primitives; editor of visual models for visual scene's code generation; visualization module to create visual scenes based on generated code; transfer data web-server; domain ontology, graphical objects ontology and application's visualization ontology.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

КОМПЛЕКСИРОВАНИЕ ТЕКСТОВЫХ ОПИСАНИЙ И СЕМАНТИЗИРОВАННОЙ ГРАФИКИ В РЕШЕНИИ ПРОЕКТНЫХ ЗАДАЧ

Соснин П.И., Галочкин М.В., Лунецкас А.А.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

sosnin@ulstu.ru

m.galochkin@ulstu.ru

lunacorp@inbox.ru

В статье представляются средства псевдо кодовой программируемой графики поддержки проектирования автоматизированных систем на концептуальном этапе. Специфику подхода определяет преобразование текстовых единиц в семантические граф-схемы в условиях взаимодействия с онтологией проекта. Возможность обратного преобразования схемы после её коррекции позволяет итеративно доводить её и исследуемую текстовую единицу до взаимно согласованных состояний, констатирующих и регистрирующих проверяемую версию понимания проектировщиком освоенного текста.

Ключевые слова: автоматизированное проектирование, графика, понимание, прецедент, семантик.

Введение

В проектировании автоматизированных систем широко используются средства текстового и табличного сопровождения, а также средства чертёжной и псевдо графики, обслуживающие создание диаграмм и виде block-and-line схем. Эффективные и мощные средства графики предоставляются для моделирования реальных объектов и процессов, а также для их образных представлений. Проектировщикам предоставляется и эффективная схемотехническая графика, которая в основном ориентирована на чертёжную регистрацию результатов проектирования. Такие представления и преобразования не только поддерживают работу на этапах принятия проектных решений, но и доведены до их нормативного включения в технологические и производственные процессы.

Использование графических моделей в современной программной инженерии стало не только повсеместным, но и вошло в ряд стандартов (UML, BPMN, VPEL и т. д.). Акцент в разработках сложных автоматизированных систем в последнее время существенным образом сместился в область концептуального проектирования, в процессе которого создаются и используются множества концептуальных моделей с разной степенью общности и детализации. Роль таких моделей выполняют “block and line” схемы разных типов, чаще называемые «диаграммами». Эволюция “block

and line” средств моделирования сложных систем, интенсивно использующих программное обеспечение, привела к разработке унифицированного языка моделирования (Unified Modeling Language, UML).

Отметим, что текстовое и табличное сопровождение в САПР заимствует практически всё достигнутое в разработках текстовых и табличных процессоров, но чаще всего только в привязках к решению задач формирования и регистрации проектной документации. Однако, как в поддержке работ с графикой, так и в работе с текстами практически отсутствует выход на автоматизированное решение задач семантики, занимающих очень важное место в оперативной работе проектировщика и особенно процессах принятия проектных решений.

Для проектировщика выход на ту семантику, для которой полезно её графическое представление, начинается с первых шагов его взаимодействия с техническим заданием и особенно принципиален на этапах концептуального проектирования, когда только еще нащупываются потенциальные прототипы решений, когда выход на реальную геометрию и нормативную графическую (образную) регистрацию преждевременен [Галочкин, 2015]. На этом этапе основная информация поступает к проектировщику через тексты, во взаимодействии с которыми он практически лишён автоматизированной графической поддержки. Такое

положение дел послужило основанием для разработки средств автоматизированного преобразования текстовых описаний ситуаций в семантические граф-схемы и схемы других типов. Предлагаемые средства реализованы как расширение инструментально-моделирующей среды OwnWIQA [Соснин, 2010].

1. Ряд контекстуальных определений

1.1. Задача

Для определённости, раскроем содержание ряда объектов и процессов, которые принципиальны для исследуемого случая (case study). О задаче говорят в том случае, когда для достижения некоторой цели необходимо предпринять действия из некоторого доступного набора действий, а выбор действий и их композиция (поиск метода решения задачи) еще не произведены. Для любой задачи наиболее важными структурными элементами являются представление задачи и решение или стратегия поиска решения. Под представлением задачи понимают ее формализованное описание в виде модели, преобразования, а которой должны привести к искомому результату.

Считается, что задача сформулирована, если: задача представлена в виде текста с осмысленной терминологией; терминология задачи общая для задачной и решающей системы определена и формально описана (или допускается такое описание) исходная ситуация; известно целевое состояние или заданы его свойства; решающая система обладает набором действий и реализующих их средств для разрешения задачной ситуации.

Текст любой задачной ситуации (как текущая постановка задачи) является ее знаковой моделью. Знаковые модели задачи содержат неопределенности, которые снимаются частично, уточняя последующую знаковую модель, с каждым полезным шагом в процессе поиска метода решения задачи. Существенный вклад в уменьшение неопределенностей вносит добавление к текстовой информации полезных графико-семантических схем, отражающих образные представления исследуемой ситуации.

Принципиальным указателем на результат решения задачи является ее цель, которая формулируется на ранних этапах решения, причём, можно определить цель, не определяя, ни как её достичь, ничем, ни когда. Анализ формулировки цели, позволяющий обобщённо определиться с ответами на вопросы как?, чем? и когда?, приводит к исходной постановке задачи, которую опасно понять неправильно. Именно с понимания обобщённой постановки задачи начинается процесс её понимания, результат которого (шаг за шагом решения) совершенствует результат понимания задачи, которое требуется представлять в явной форме как для проектировщика, ответственного за решение задачи, так и для его коллег по коллективу.

1.2. Понимание

Понимание одна из важнейших сторон человеческого освоения мира, характеризующая качество и степень этого освоения. Всякое понимание является не простым отражением действительности, а скрашено целенаправленной деятельностью, ее задачами. Более того, процесс понимания развивается как бы в рамках познавательной деятельности, реализуя свой диалогический, «вопросно-ответный» характер. Диалог - путь не только к пониманию, но и к взаимопониманию.

Понимание, даже если оно ложно, необходимо для развития знания как средства его систематизации и совершенствования. Для современного ученого понимание уже не сводится к построению наглядного представления об исследуемых явлениях. Под пониманием обычно имеется в виду умение сконструировать из выраженных в формальном виде характеристик явления некоторую его возможную модель. Понимание связано с предварительным знанием основных характеристик изучаемых объектов, оно имеет отношение к анализу структуры этого знания и к эффективному его упорядочиванию.

Понятными считаются те факты, явления, теории, которые укладываются в рациональную схему такого "канона" как картина мира и тем самым оправдывают предварительные ожидания исследователя.

«Понимание» следует понимать, как субъективную, индивидуальную способность личности к овладению какими-либо знаниями или навыками, а также сам процесс и результат реализации этой способности. Часто понимание приходит в результате объяснения. Понимание неотделимо от оценочной деятельности сознания. Нельзя разделить понимание и оценку: они одновременны и составляют единый целостный акт.

Различают три уровня понимания: понимание того, что нам в основе своей уже известно; понимание того, что требует формирования или усвоения новых понятий; понимание, предполагающее формирование или усвоение новых умственных процедур.

1.3. Понятия

В решение проектных задач понимание следует включать как специфический вид деятельности, в осуществлении которого используются определённые методы и средства. К числу принципиальных средств такой деятельности относятся понятия (в англоязычной терминологии - концепты).

Понятие является практически полезным средством для измерения значения языковой конструкции, например, в тексте постановки задачи. Оно может выражаться аналитически (развернутое языковое выражение) в форме определенной

дескрипции, или с помощью термина, то есть синтетически. Понятие является логической категорией, границы которой очерчиваются содержанием понятия и его текущим объемом. Формальным представлением границ можно считать определение понятия. Определение фиксирует "место", которое конкретное понятие занимает в системе других понятий. Схемой стандартного определения понятий является "класс, свойства, примеры", такая схема подчеркивает классифицирующую функцию понятия.

Понятие, являясь предельным обобщением, одновременно является и результатом свертывания всех тех многообразных восприятия и представлений, которые возникают в процессе взаимодействия человека со средой.

Инструментальное определение понятия должно опираться на язык адекватного представления понятийного содержания [Guarino, 2009]. Этот язык должен ориентироваться на анализ и синтез языковых выражений в задачах понимания текстов и текстовых объяснений.

1.4. Образ

Учитывая, что образ является полезным инструментом понимания и объяснения и дополняет символические представления, раскроем его содержательную интерпретацию.

Понятие "образ" относится к категории первичных понятий, которое находит широкое применение в речевой и научной практике. В общем случае понятие образа имеет много толкований и существует только в форме толкований. К числу типичных толкований этого понятия следует отнести толкование образа в виде: одной из форм отражения реальности; результата восприятия как целостного отражения предметов, ситуаций и событий. Приведенных толкований достаточно для того, чтобы указать место локализации конкретного образа – мозговые структуры человека. Конкретные образы часто получают материальное представление в форме моделей образа, которые широко используются в языке, знаковых моделях и других средствах практической деятельности.

Представления следует считать высшей формой чувственного отражения действительности в виде наглядно-образного знания. Конкретное представление может иметь разложение по системе образов. Решающим значением для проверки и коррекции образа является включенность восприятия, использующего этот образ, в процессы практической деятельности, общения и научного исследования.

На основе имеющихся образов можно формулировать новые понятия и в этом проявляется связь образов с интуицией. Интуиция достаточно часто выражается как процесс и результат обратного комбинирования. Целостность и устойчивость результата, комбинирования образов можно

включить в число критериев приемки этого результата.

Оперирование образами относится к категории качественных методов познания. Познавательный образ – одна из форм существования научного знания. Степень подобия между таким образом и отражением им фрагментом обычно выше, чем у соответствующего формального представления. Образную природу имеют такие важнейшие образования, как: системы первичных понятий, основы любой науки, научная картина мира, цель. Достаточно часто не остается ничего другого, кроме доверия к объяснительной силе образа.

Этот факт ставит вопрос о необходимости создавать условия, способствующие формированию и эффективному использованию образов, опираясь на их эвристические способности. К разряду таких условий относятся наличие моделей образов, языка для выражения моделей образам и средств для оперирования моделями. Целостность образа, наличие или отсутствие конкурирующих образов определяют степень правдоподобности образа и его убедительность. Специальные доказательства для образа не требуются. Доказательство заменяет правдоподобность образованное конструкции.

Конкретное толкование или некоторое множество толкований понятия образа можно считать "образом образа", что служит информативным примером употребления этого понятия. Пример раскрывает связи образа с проблемой понимания (толкования) отображаемых объектов и явлений, идея образного представления пригодна в качестве гипотезы, объясняющей понимание языка и речи» Мы понимаем языковые конструкции благодаря тем образам, которые они способны вызывать. В этом плане образ существенным образом связан с содержательной информацией.

Главным для образа является целостность в существенном, а не в дополнительных деталях. Условие сохранения образа используется как критерий контроля за ходом рассуждений и их соответствия исходным концепциям. Образы, а тем более модели образов, доступны изучению и систематизации. Простейшей формой систематизации знаний об образах является классифицирование. В основу классификаций образов могут быть положены: уровень отражения (чувственный, познавательный); порождающий объект (фрагмент действительности, знак); причинность (спонтанный, управляемый); целенаправленность (целостность представления, объяснение, обоснование, контроль состояний; значимость (доминантный, подчиненный, дополнительный, конкурирующий).

2. Понятийно-образное комплексирование в процессе понимания

Конструктивный процесс понимания при решении проектных задач следует осуществлять с помощью подходящих рассуждений проектировщика, использующего подходящие понятия и модели образов. В такой работе в его активность включена активность его мозга, в котором за действия с понятиями «отвечает» левое полушарие мозга, а за действия с образами – правое полушарие мозга (рисунок 1).

На рисунке 1 обобщённо отражена функциональная «ответственность» каждого из полушарий, а также операционные условия, способствующие конструктивной работе как с процессом понимания, так и с его результатом, который находит свое выражение в понятийно-образных составляющих, включённых постановку решаемой задачи.

Схема демонстрирует, что для конструктивной поддержки понимания, необходимо иметь и использовать в инструментальной среде библиотеки понятий и образов, а также средства для оперативного формирования понятий и образов при взаимодействии с постановкой задачи, её развитием и коррекцией. К числу таких средств относятся специализированные текстовый и графический редакторы.

3. Понятийно-образное поддержка понимания в среде WIQA

Инструментальная среда WIQA изначально разрабатывалась для моделирования совокупностей технологических (нормативных) и предметных проектных задач, которые приходится решать разработчикам АС. Как источник нормативных

задач были использованы типовые задачи потоков работ концептуального проектирования в технологии Rational Unified Process (RUP). По его предназначению и содержанию, комплекс WIQA разрабатывался как специализированная АСТ технологического типа.

Одной из важнейших особенностей концептуального проектирования в среде WIQA является потенциальное применение всех средств этого инструментария к любой задаче Z_i дерева $T(\{Z_i\})$ проектных задач $\{Z_i\}$, если в этом будет необходимость. В таком применении задача Z_i представляется её вопросно-ответной моделью (QA-моделью, $QA(Z_i)$), структурирующей процесс решения задачи в формах вопросно-ответных рассуждений. К специфике инструментария WIQA относится встроенная в него семантическая память вопросно-ответного типа и концептуально-алгоритмический язык программирования, определённый над этой памятью [Sosnin, 2014] и позволяющий разрабатывать необходимые расширения.

К числу таких расширений относятся средства, обеспечивающие создание и использование Базы Опыта (проектной организации, разрабатывающей семейства АС), в состав которой входят «База прецедентов» и «Онтология» [Sosnin, 2015]. Именно эти компоненты несут основную функциональную нагрузку в конструктивной работе с пониманием при решении проектных задач (рисунок 2).

На рисунке 2 показано, что по ходу решения задачи для неё создаётся интегрированная модель прецедента, предназначенная для осуществления и облегчения её повторного использования.

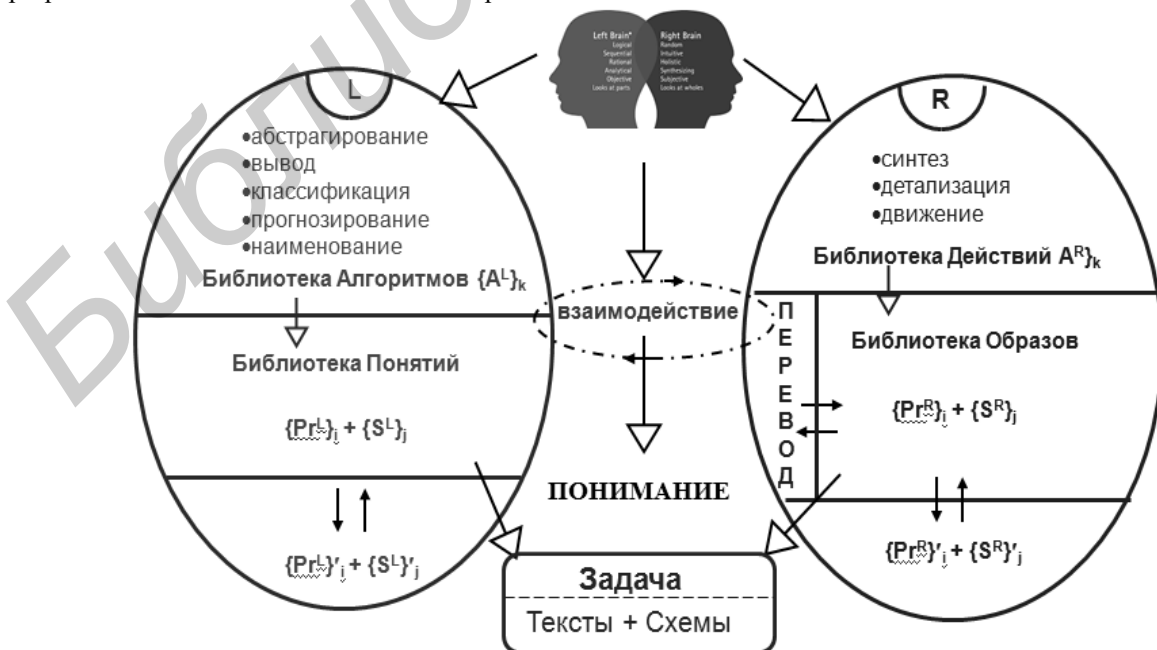


Рисунок 1– Комплексирование понятийно-образных действий

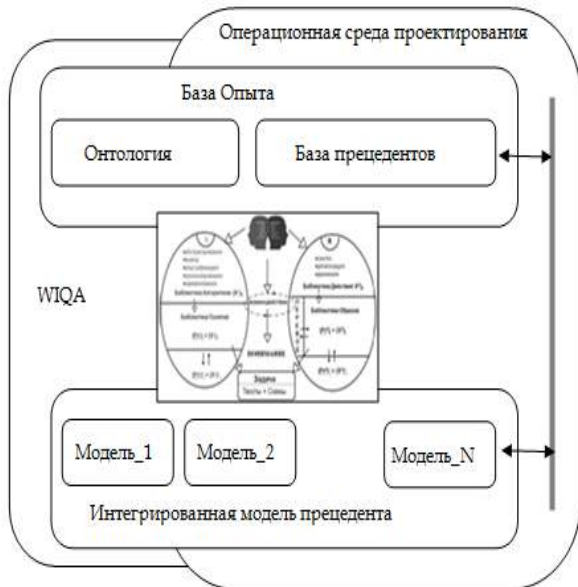


Рисунок 2 – Среда достижения понимания и представления его результата

Интегрированная модель понимается и строится по образцу интеллектуально-обработанного условного рефлекса как очередная единица опыта, которую создал и которой овладел проектировщик, решивший соответствующую задачу. В результате интеллектуальной обработки, процесс которой нормативно определен и поддерживается инструментарием WIQA [Sosnin, 2015], порождается совокупность специализированных моделей прецедента, приведенных на рисунке 3.

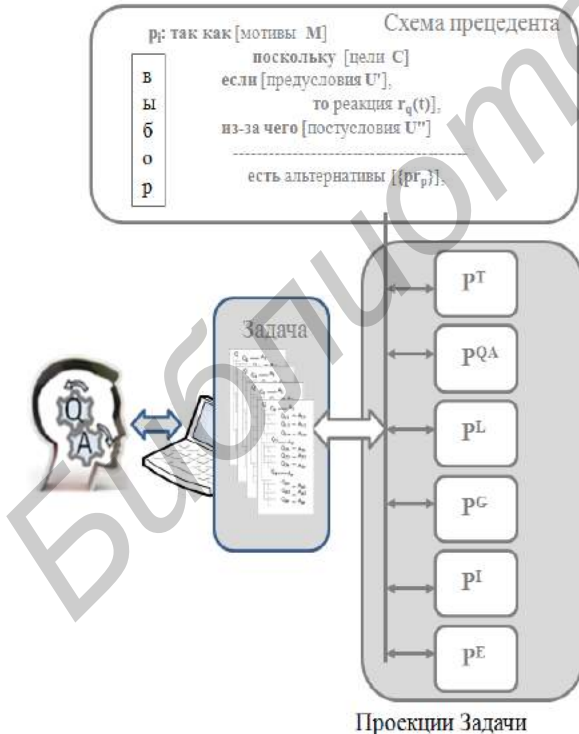


Рисунок 3 – Нормативная схема модели прецедента

Модель прецедента P включает: текстовую составляющую P^T , в виде постановки задачи; логическую составляющую P^L , формула которой представлена на рисунке 3 в верхней его части;

вопросно-ответную модель P^{QA} задачи Z_k ; графическое представление прецедента P^G ; исходный псевдокод P^I и исполняемый код P^E . Нормативная модель прецедента построена таким образом, чтобы она раскрывала концептуальное содержание задачи и представляла её концептуально-алгоритмическое решение (псевдокод решения). По этой причине в модели принципиальное место занимают конструкции на естественно-профессиональном языке L^P предметной области $\{Z_k\}$.

По ходу решения определённой проектной задачи соответствующая модель прецедента формируется шаг за шагом с возвратом «назад» при необходимости коррекций. Как и процесс решения задачи, процесс формирования модели прецедента является итерационным, что образно отражено на рисунке 4, где раскрывается «жизненный цикл» порождения прецедента [Sosnin, 2014].

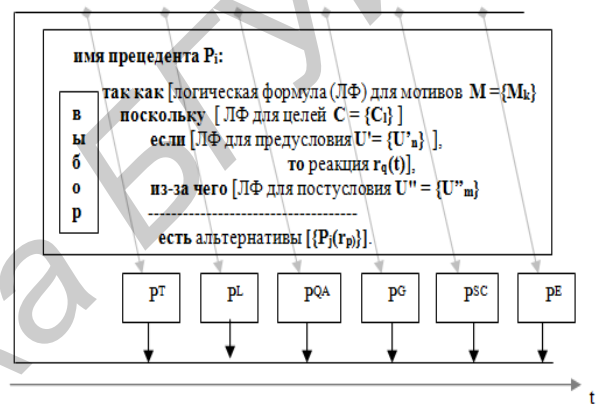


Рисунок 4 – Жизненный цикл модели прецедента

Следует отметить, что интегрированная модель прецедента используется как та «целостность», в рамках которой осуществляется процесс понимания и регистрируются его результаты (состояния), достигаемые по ходу решения задачи. Процесс развёртывается с использованием принципа «дополнительно наговоренное дорисовывается, а дополнительно нарисованное оговаривается».

В каждом из достигнутых состояний понимания, находящих свой конструктивное выражение в версии заполнения нормативной схемы модели прецедента, используются понятия, присутствующие в онтологии, и образно-семантические схемы, для построения которых применяется специализированный графический редактор. Семантика схем определяется теми именами концептов, который введены в схемы для обозначения блоков и связей между ними.

К специфике графического редактора относится то, что для него разработана утилита, обеспечивающая перевод текстовых фрагментов, например, постановки задачи, в их семантическую граф-схему (СГС). Более того, построенная схема может быть преобразована в псевдо-кодovou программу её рисования, в том числе и пошагового рисования. С полученным псевдокодом можно

проводить полезные модификации, симулируя различные версии рисования, способствующие пониманию рисуемого.

Так, например, для задачи для текста задачи «Определить форму поверхности жидкости в сосуде, скользящем без трения по наклонной плоскости», после его перевода в пролого-подобную форму

Скользит (сосуд, жидкость; наклонная плоскость), без трения.

Определить (форма, жидкость, поверхность),

результат перевода будет иметь вид, приведённый на рисунке 5.

Более того, созданную схему можно подключить к «Онтологии» [Sosnin, 2015] и, за счёт информации о понятиях, детализировать схему, а также дополнить её комментариями. Подключение образно отражено на рисунке 6, на котором показано, что взаимодействие с онтологией поддерживается через специальное окно, в котором, например для «связи», раскрываются все варианты систематизации, зарегистрированные для неё в онтологии (род—вид, часть-целое, ассоциация и другие).

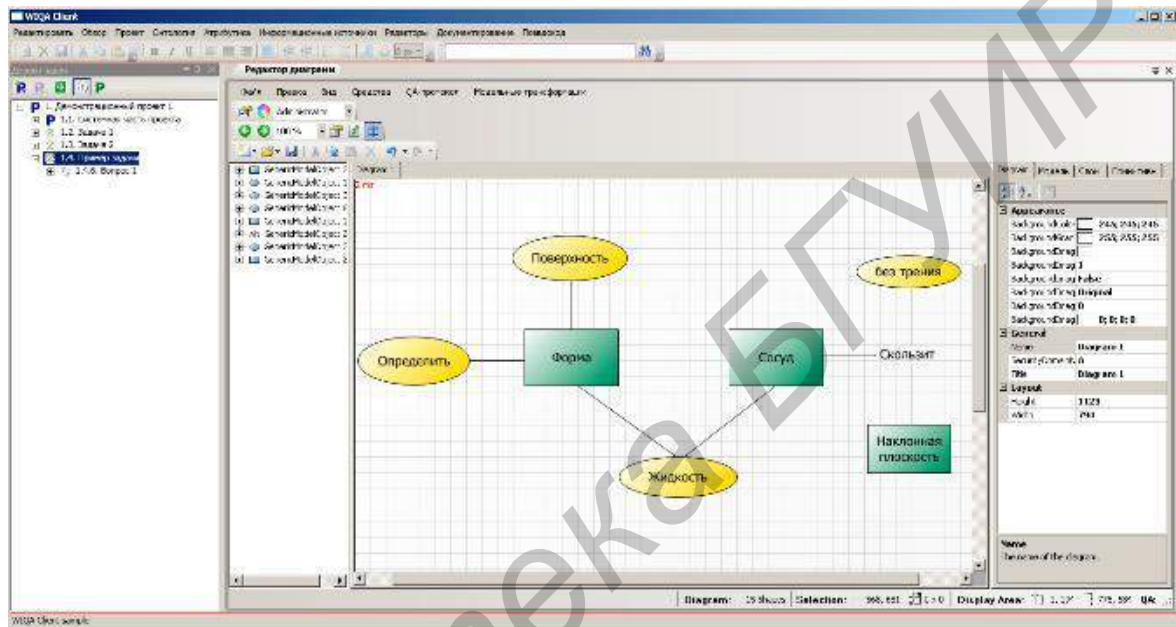


Рисунок 5 – Формирование семантической граф-схемы текстовых единиц

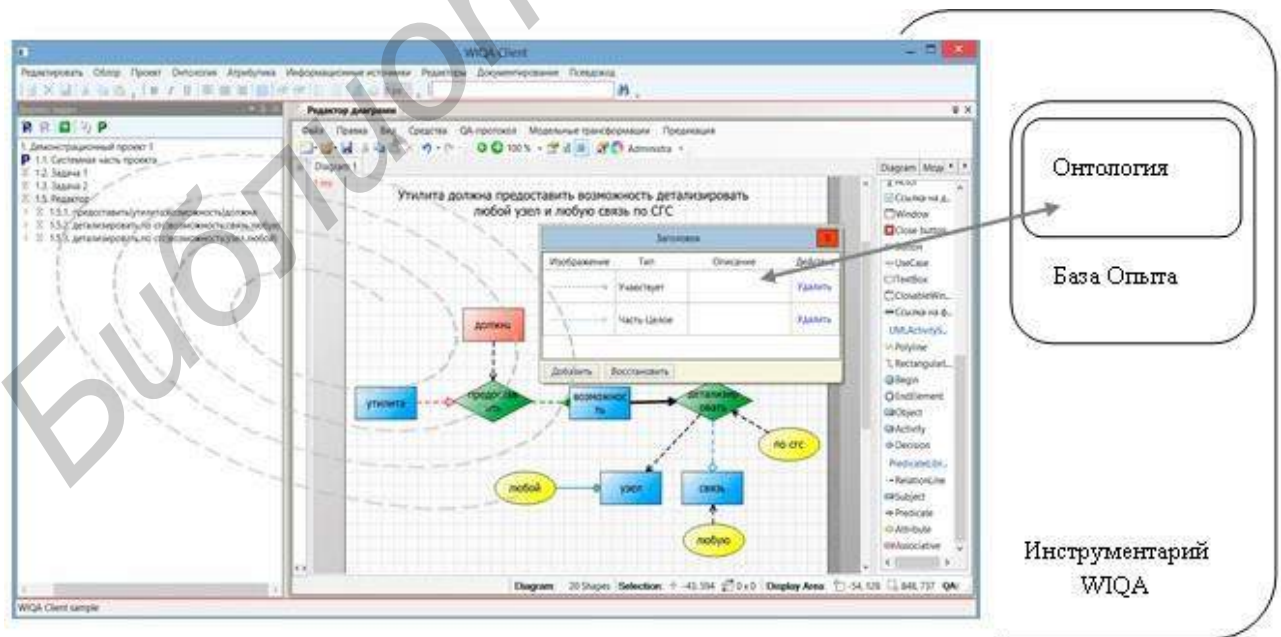


Рисунок 6 – Формирование семантической граф-схемы текстовых единиц

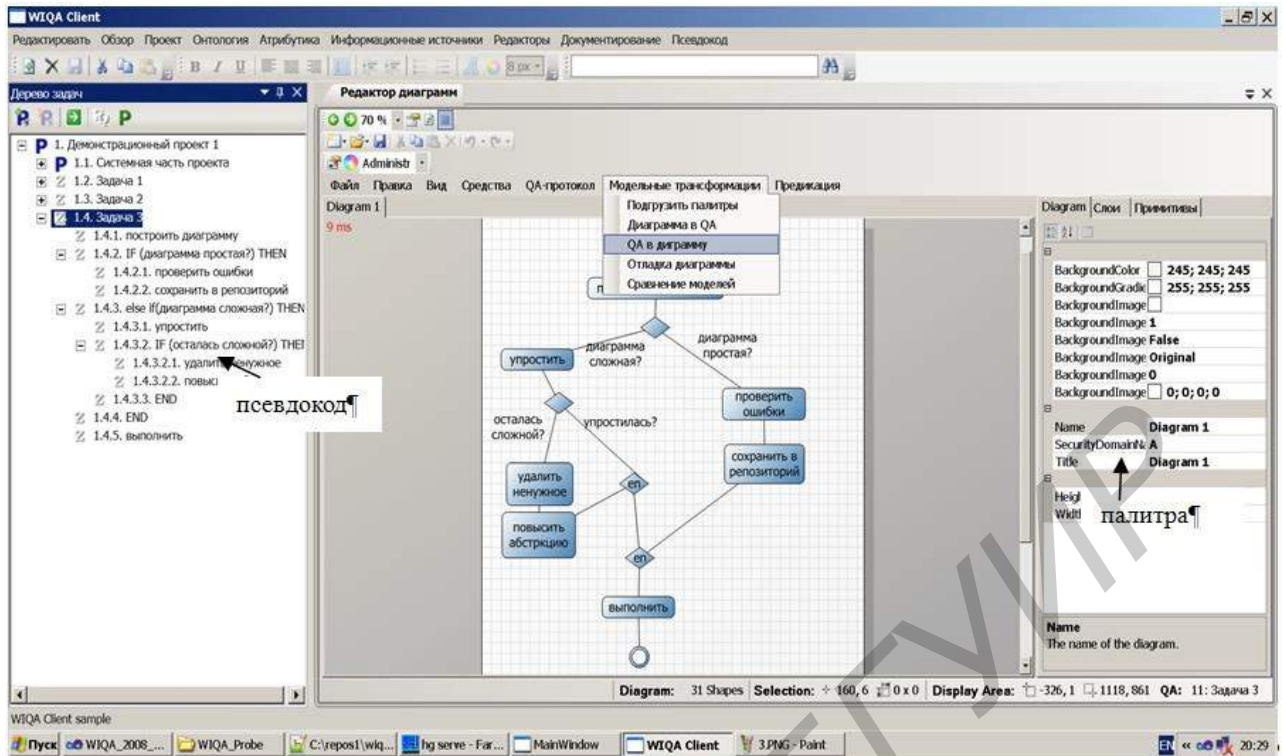


Рисунок 7 – Построение диаграмм активностей

На рисунке 6 также отражена (спиралью) динамика итеративного согласования текстового описания с его СГС. В основе такого согласования лежит обратная связь СГС с её пролого-подобным аналогом. Любые коррекции на схеме приводят к коррекциям текстовых аналогов. Этот механизм полезен при переводе сложных предложений текстов в простые, которые и образуют их пролого-подобные версии. Отметим, что текст T, его пролого-подобная версия и согласованная с ними СГС конструктивно выражают то понимание, которое создатель этих конструктов связал с ними или с которыми согласен их «читатель».

Средства для работы с графикой в инструментари WIQA созданы не только для построения СГС и их использования. Их предназначение построение и семантизация «block-and-line» диаграмм любых типов, включая диаграммы UML. На рисунке 7 приведено их применение в построении диаграмм активностей.

В этом случае принципиальную роль выполняет отображение строящейся или уже построенной диаграммы на её аналог в форме псевдо кодовой программы, обеспечивающей повторное рисование диаграммы в поле графического редактора. Если пролого-подобный аналог декларативно выражает содержание СГС, то псевдо кодовая программа построения диаграммы (не только для диаграмм активностей) выражает концептуально-алгоритмическое содержание её диаграммного аналога, который может быть также подключен к онтологии и откорректирован, например лексически.

Псевдо кодовое программное представление диаграмм очень важный вид инноваций для предметной области Model Driven Development (Разработка систем, управляемая моделями). Такой вид разработок считается одним из наиболее перспективных в создании сложных систем, интенсивно использующих программное обеспечение. В основе этого вида разработок лежит явное и конструктивное выражение «понимания».

Заключение

В статье представляется подход, позволяющий управлять процессами понимания проектировщика при решении задач в проектировании автоматизированных систем. Такое управление дополняет работу левого и правого полушарий мозга проектировщика, способствуя объединению и переплетению логических и образных процессов и их результатов в целостности, обслуживающие явное и конструктивное представление как процесса понимания, так и его результата.

Для поддержки процесса понимания и представления его результата используется инструментально-моделирующая среда WIQA, предназначенная для обслуживания концептуального проектирования АС. Особую роль в поддержке выполняет взаимодействие с Базой опыта и такими её компонентами как «Онтология» и «База прецедентов». Типовой формой регистрации достигнутого понимания является нормативная схема модели прецедента.

Библиографический список

[Галочкин, 2015] Галочкин, М.В., Соснин, П. И. Средства псевдокодовой программируемой графики в проектировании автоматизированных систем. Автоматизация процессов управления. 2015. № 1 (39). С. 82-88

[Соснин, 2010] Соснин, П. И. Вопросно-ответное программирование человеко-компьютерной деятельности – Ульяновск : УлГТУ, 2010. – 240 с.

[Guarino, 2009] Guarino, N., Oberle, D., Staab, S.: What is an Ontology? In S. Staab and R. Studer (eds.), Handbook on Ontologies, Second Edition. International handbooks on information systems. Springer Verlag, 2009, pp. 1-17.

[Sosnin, 2014] Sosnin, P. A Personal Ontology of an Individual Occupational Experience. In Proc. of the 8th IEEE International Conference on Application of Information and Communication Technologies (AICT), Astana, Kazakhstan, 2014, pp. 451-457.

[Sosnin, 2015] Sosnin, P. An Ontological Support for Interactions with Experience in Designing the Software Intensive Systems. LNCS 9158, Springer, Heidelberg, 2015, pp. 387-400.

AGGREGATION OF TEXTUAL DESCRIPTION WITH SEMANTICIZED GRAPHICS IN SOLVING THE PROJECT TASKS

Sosnin P.I., Galochkin M.V., Lunecaks A.A.

Ulyanovsk State Technical University

sosnin@ulstu.ru

m.galochkin@ulstu.ru

lunacorp@inbox.ru

The article presents a number of means that help to build graphical models for textual units in real-time solving the project tasks. They also help to translate graphical diagrams in their conceptually algorithmic descriptions.

Key words: Automated designing, graphics, precedent, semantics, understanding

Introduction

Offered means allow improving the interaction of a designer with generated and used textual descriptions by automated constructing semantic images extracted from these texts. Such constructing involves both hemispheres of the designer's brain where the logical and figurative processes are combined in the most effective wholeness that positively influences onto creative actions of the designer.

To translate the definite part of the investigated text (for example, the statement of the project task) in a corresponding semantic image, the designer can use a specialized linguistic processor, convertor and graphical editor that supports creating the schemes with programmable interpretations concerned their semantic content. The process of such investigation is implemented in the toolkit OwnWIQA. This toolkit also support back word transformation. This tool also supports inverse to programmable forms. This innovation is especially useful for Model Driven Development estimated as promising way of increasing the level of success in designing the software intensive systems.

Main Part

In the work with project tasks, the described approach provides including the transformations of textual units (from statements of tasks or personal or collective reasoning) in their graphical analogs with programmable structures. For the designer, such transformations help to combine the use of symbolic and corresponding graphical actions of the left and right hemisphere in coordination. Transformations visually support these actions at the level of common semantics.

The main aim of transformations is to build figuratively FS-scheme for the textual unit. Step by step, from one state to another, this FS-scheme (presented in the block-and-line view) is evolved by its filling the additional semantic information. At some of these steps, the FS-scheme is presented in forms of pseudo code programs that make easier corrections of the FS-scheme and interactions with it.

Creating the FS-scheme, the designer builds own understanding of the corresponding text. Step by step created understanding is corrected and evolved. Moreover, understanding is expressed in the visual form by the structure of the semantic FS-scheme and its content. By the feedback, this result can be used for correcting the corresponding text if it is allowed for the designer. In this case, the designer iteratively builds a figuratively symbolic expression of achieved understanding in the form that can be checked by other members of the collaborative work in the process of designing.

Tools of OwnWIQA for working with graphics also support the construction and semantization of «block-and-line» diagrams of the other types, including UML diagrams.

Conclusion

The described way of coordinated building the textual units and their graphical models supports the constructive work with understanding in solving the project task. The way also supports the transformation of diagrams to their executable program forms. This possibility facilitates to the success in designing the software intensive systems.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

КОМПЛЕКС ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ, РЕАЛИЗУЮЩИХ ЛОГИКУ ОЦЕНКИ ПСИХОФИЗИОЛОГИЧЕСКОГО СОСТОЯНИЯ ЧЕЛОВЕКА

Порядин А.Е., Сидоркина И.Г.

*Поволжский государственный технологический университет,
г. Йошкар-Ола, Республика Марий Эл, Россия*

lazy.ant10@gmail.com

igs592000@mail.ru

В работе рассматриваются алгоритмы диагностики, реализованные в исследуемых инструментальных средствах оценки психофизиологического состояния человека.

Сформулированы требования к системе поддержки принятия решений для диагностики психофизиологических особенностей человека. Предложенный к использованию алгоритм обладает следующими преимуществами: проведение диагностики с использованием мобильных устройств, гарантирует сохранность данных в течение длительного периода времени, позволяет выявлять тенденции характерные для людей определенной профессии, либо проживающих в определенной местности.

Ключевые слова: диагностика психофизиологических особенностей, системы поддержки принятия решений, нейронные сети, психосемантика, семантическое пространство.

Введение

До недавнего времени для диагностики психологического и психофизиологического состояния человека активно использовали бланко-бумажные тесты. Но на сегодняшний день в таком подходе выявлен ряд серьезных проблем. Одним из самых крупных недостатков стало снижение уровня достоверности результатов таких тестов [Колочкин и др., 2014]. С развитием интернета уже не составляет труда найти «ключи» и описания к интерпретации результатов для всех стандартизированных тестов. Также появилось множество изданий, консультирующих испытуемых о том, как обойти тесты, применяемые для профессионального отбора и ориентации. Наиболее важным недостатком бумажно-бланковых методов является отсутствие учета эмоционально-экспрессивного компонента поведения человека во время прохождения теста, что не позволяет выявить отношение к тому или иному стимулу рассматриваемому в данный момент времени и предоставляет возможность испытуемому скрыть свою истинную реакцию.

Также бланко-бумажные методики не учитывают мотивацию испытуемого. Вне сомнения мотивация испытуемого находящегося на реабилитации и испытуемого стремящегося пройти профессиональный отбор совершенно разная. В некоторых случаях стоит учитывать открытость

людей. Часто при прохождении медосмотров и диагностики люди умалчивают об изменениях в самочувствии, настроении или состоянии. Причины подобных поступков могут сильно различаться, от страха потерять работу до банального нежелания проявить слабость.

Вышеперечисленные проблемы привели к тому, что в настоящее время много внимания уделяется вопросу получения объективной информации и построения достоверных выводов по результатам диагностики. Современным решением, способным удовлетворить текущие потребности, являются аппаратно-программные комплексы (АПК). При этом часть комплексов обладают функционалом для получения дополнительной информации во время проведения диагностики, например, используя электроэнцефалограмму (ЭЭГ), которая помогает предоставить более объективные выводы.

Исследуемые инструментальные комплексы не только устраняют недостатки использовавшихся ранее методик, но и обладают рядом преимуществ, возникшим в результате автоматизации процесса. Например, позволяют полностью исключить межличностное взаимодействие в процессе диагностики, что экономит время специалиста и добавляет объективность в результаты тестирования. Подобные системы не только обладают гораздо большим набором тестов, чем самые лучшие специалисты, но и позволяют

получить результат в короткое время. А самое главное они являются гораздо более доступным решением, чем поиск и найм хорошего специалиста, особенно для предприятий, которым необходим мониторинг состояния сотрудников на удаленных объектах, либо простых пользователях мобильных устройств которым интересно узнать о состоянии своего организма.

Рассмотренные в данной статье системы и их аналоги получили широкое распространение во многих областях деятельности человека. Они помогают профессиональной ориентации, оценке профессиональной пригодности, самопознанию и самосовершенствованию. Также могут быть полезны для формирования индивидуального режима труда и отдыха, что позволяет повысить эффективность рабочей и учебной деятельности.

Целью данной статьи является исследование алгоритмов работы наиболее известных на сегодняшний день комплексов, используемых для оценки психофизиологического состояния человека, с целью формирования требований к разрабатываемой системе поддержки принятия решений в области диагностики психофизиологических особенностей человека. Учитывая отсутствие полного описания функционала и возможностей для части аппаратно-программных комплексов (АПК), анализ был проведен только по общедоступной информации, без сравнения одинаковых функциональных особенностей каждого АПК.

1. Нейронные сети для диагностики состояния человека в комплексе «Реакор»

Психофизиологический комплекс «Реакор» [Захаров и др., 2004] обладает следующим функционалом:

- мониторинг психофизиологических показателей человека;
- оценка качества профессиональной деятельности;
- проведение процедур, способствующих улучшению психофизиологического состояния;
- формирование навыков контроля состояния и оптимального функционирования;
- повышение адаптационных возможностей.

Позволяет обрабатывать полученные данные различными математическими методами, например, статистический анализ с построением гистограмм распределения и оценка значимых изменений на разных этапах исследования.

Для идентификации состояний человека, во время прохождения процедур и выполнения заданий, используется ЭЭГ. Сложность диагностики связана с обработкой большого количества быстро меняющихся сигналов во множестве областей

мозга. Традиционный анализ ЭЭГ учитывает изменение лишь одного параметра, но многие авторы упоминают высокую значимость функциональных связей, устанавливаемых синхронной ритмикой в различных отделах головного мозга, для корректной оценки состояния человека.

2. Использование метода психосемантики на примере комплексов АПК «MindReader» и АПК «БиоМышь»

Другим подходом к оценке психофизиологического состояния является метод психосемантики. Метод основан на изучении форм знаний, таких как символы, образы, вербальные формы и символические действия в индивидуальном сознании. Данный подход позволяет анализировать влияние мотивационных факторов и эмоциональных состояний испытуемого на формирование системы значений. Основным инструментом в методе экспериментальной психосемантики являются построенные экспертом субъективные семантические пространства. Пример семантического пространства личностных качеств представлен на рисунке 1.



Рисунок 1 – Сказочные персонажи в семантическом пространстве (F1, F2) личностных качеств

Проблемой метода является вмешательство сознания в процесс тестирования, что искажает результаты, однако существующие на данный момент решения разработаны с учетом данного недостатка.

Современные системы используют метод психологической обратной связи, при этом алгоритм работы в большинстве случаев не отличается. Тестовыми стимулами являются сгруппированные по семантическому значению слова, фразы и образы, ассоциирующиеся в индивидуальном опыте с предметом или событием, способным вызвать какую-либо реакцию у испытуемого. Стимулы делятся на две группы. Первая группа должна иметь однозначное толкование и отражать конкретную сферу жизнедеятельности, например, работа, семья, привычки, знания. Вторая группа является обобщенным бессмысленным буквосочетанием всех

слов, представленных в первой группе. Во время тестирования стимулы данных двух групп последовательно чередуются. Вмешательство сознания предполагается избежать с помощью двойного маскирования при предъявлении стимулов. Время предъявления стимула составляет приблизительно 30-40 мсек, после его заменяет маскирующая последовательность символов, примерно на 200-300 мсек, далее пауза и предъявление нового стимула. Предполагается что при предъявлении стимула таким образом человек не успевает его осознать, но этого времени хватает для фиксации стимула зрительным анализатором.

Комплекс, работающий по данной методике - «MindReader». [Колочкин и др., 2014] Процедура диагностики производится в полностью неосознанном режиме для испытуемого. На экране без пауз представляют стимулы: слова, фразы или изображения, сгруппированные по темам. Скорость показа подбирается таким образом, что человек не успевает прочитать слово или рассмотреть изображение, но психика воспринимает их на подсознательном уровне и реагирует, при этом регистрируется время реакции с помощью нажатия на кнопку. Значимые и актуальные для испытуемого стимулы вызывают отклонение от среднего времени реакции. Испытуемый видит лишь сменяющиеся стимулы и может не догадываться о целях тестирования. Данный подход, по мнению разработчиков позволяет исключить субъективность принятия решений. Другим примером использования данной методики является АПК «БиоМышь». Для фиксации реакции на стимулы используется датчик пульса, что позволяет отслеживать изменения в ритме сердечных сокращений и анализировать его вариабельность.

Рассмотренные выше инструментальные средства способны диагностировать наличие невротических комплексов, установить мотивы поведения, причины стрессовых состояний, признаки ранних проявлений психических расстройств.

3. Использование специальных математических моделей

Наиболее подробно алгоритм работы данного подхода описан в [Величко, 2010]. Разработанная система позволяет решить две основные задачи:

- Представление общего заключения о текущем психофизиологическом состоянии спортсмена.
- Мониторинг состояния спортсмена для прогнозирования соревновательной готовности.

Для решения первой задачи были формализованы результаты различных методик путем приведения всех параметров к универсальной бальной шкале с использованием решающих правил. Последующий математический анализ всех

параметров производит комплексную оценку психофизиологического состояния и формирует заключение в понятной спортсмену форме. Общая схема оценки психофизиологического состояния спортсмена представлена на рисунке 2.



Рисунок 2 – Схема оценки психофизиологического состояния спортсмена

Для решения второй задачи был реализован алгоритм, работающий в два этапа:

- 1) разведочный анализ, обеспечивающий обработку выбросов в исходных данных;
- 2) сплайн-аппроксимация.

По результатам работы данного алгоритма составляется отчет с прогнозированием уровня соревновательной готовности спортсмена.

4. Обоснование выбора алгоритма для разработки системы диагностики психофизиологических особенностей

Исследуемый комплекс инструментальных средств в области диагностики психофизиологических особенностей человека разрабатывается в виде клиентских приложений под настольные компьютеры и мобильные устройства. Обработка и хранение результатов диагностики проводятся на сервере. Учитывая нелинейность данных, объем, большие размерности, зависимость результата от множества переменных и большое количество репрезентативных данных для обучения, на данный момент, лучшим алгоритмом для подобной задачи будет использование нейронных сетей.

Как известно человек довольно сложная, стохастическая система. [Горбунов, 2005] Психофизиологическое состояние зависит от состояния множества подсистем, например, физиологическое состояние, эмоциональная область, когнитивная сфера, поведенческий уровень. В свою очередь общее физиологическое состояние зависит от состояния отдельных органов. Эмоциональная сфера зависит от окружающей среды, настроения. После подобного рассмотрения человека как сложной системы можно заметить, что многослойные нейронные сети являются очень удобным инструментом по двум причинам:

- Нелинейность взаимоотношений между первым и последним слоями можно определить количеством слоев и количеством нейронов в каждом слое.

- Каждый промежуточный слой символизирует один из уровней системы и отдельно интерпретируется.

Заключение

Бумажно-бланковые методики устарели, они больше не могут обеспечить объективную оценку. Современным этапом развития средств диагностики и оценки психофизиологических особенностей стали АПК и мобильные приложения. Такие комплексы не идеальны, у текущих решений есть ряд проблем, например, необходимость в дополнительном оборудовании. Разрабатываемая и исследуемая система решает часть проблем, связанных с возможностью использования мобильных устройств для диагностики и хранением данных. Система использует периодическое подключение к интернету, для передачи данных. Предложенный алгоритм повысит точность диагностики и сможет выявлять тенденции в изменении психологического состояния людей с определенной сферой деятельности или проживающих в определенной местности. Хранение данных на удаленном сервере гарантирует сохранность данных в течении длительного времени и возможность передать эти данные специалистам своей организации для анализа.

Библиографический список

[Колочкин и др., 2014] Психофизиологическая объективизация результатов психологического тестирования. Проблемы и пути их решения / Колочкин С.Н. [и др.] // Фундаментальные исследования 2014. №9-10. С.2317-2321.

[Захаров и др., 2004] Экспериментальные исследования и анализ психофизиологического состояния и деятельности человека-оператора / Захаров Е.С. [и др.]; // Известия ЮФУ. Технические науки. 2004. №6. С.39-41.

[Величко, 2010] Величко Е. Н. Программно-аппаратный комплекс оценки психофизиологического состояния спортсмена [Текст]: автореф. дис. на соиск. учен. степ. канд. техн. наук : 05.11.17 / Величко Е.Н. // ИТМО. – Санкт-Петербург. 2010. – 18 с

[Горбунов, 2005] Горбунов И. А. Диагностические возможности психофизиологических характеристик человека [Текст]: автореф. дис. на соиск. учен. степ. канд. псих. наук : 19.00.02 / Горбунов, И.А. // СПбГУ. – Санкт-Петербург. 2005. – 26 с.

SET OF SOFTWARE TOOLS WITH IMPLEMENTING LOGIC FOR EVALUATION OF PSYCHOPHYSIOLOGICAL STATE

Poryadin A.Y., Sidorkina I.G.

*Volga State University of Technology,
YoshkarOla, Republic of Mari El, Russia*

lazy.ant10@gmail.com

igs592000@mail.ru

This document presents an analysis of existing software tools for the evaluation of human psychophysiological state, special attention is paid to the diagnostic algorithms. In the end of given document were formulated requirements for the modern decision

support system for the diagnosis of human psychophysiological state.

Keywords: psychophysiological evaluation of human state, decision support system, neural networks, psychological feedback.

Introduction

Using old methods (papers and blanks) for evaluation of human psychophysiological state has many shortcomings, which raises doubts about the correctness of the results. Today we have new diagnostic tools, such as hardware-software complexes. These complexes allow perform more objective and accurate evaluation, faster and without the specialists.

Main Part

In this part of document was performed analysis of the main algorithms for the evaluation of human psychophysiological state.

The first approach is the use of neural networks for diagnosis. Example of the implementation of that algorithm it is complex "Peakop". The second approach is the use of psychological feedback for diagnosis. Example of the implementation of that algorithm it is complex "MindReader". The third approach is to use sets of mathematical models for diagnosis. Used in a variety of modern systems.

At the end of this section have been formulated requirements for the analysis algorithms of modern decision support system for the evaluation of human psychophysiological state.

Conclusion

In this part was reviewed advantages and the main differences between the system proposed to the development and the systems analyzed in this document.



УДК 004.822:514

СЕМАНТИЧЕСКИЕ АСПЕКТЫ ПРЕДСТАВЛЕНИЯ МЕТАДАНЫХ В СИСТЕМЕ «КОРПУС-МЕНЕДЖЕР»

Невзорова О.А., Мухамедшин Д.Р., Курманбакиев М.И.

*НИИ «Прикладная семиотика» Академии Наук Республики Татарстан,
г. Казань, Россия*

onevzoro@gmail.com

damirmuh@gmail.com

write@marat.link

В статье рассказывается об опыте разработки модели представления метаданных в системе управления лингвистическими данными (корпус-менеджер), предназначенной для работы с электронным корпусом текстов татарского языка. В частности, в работе затронуты семантические аспекты представления метаданных.

Ключевые слова: метаданные, корпус-менеджер, семантическая сеть, RDF.

Введение

Одним из важных аспектов эффективной работы с электронным корпусом текстов является корректное описание так называемой «внешней» разметки метаданных текстов, которая включает библиографические данные, а также тематические, социологические и типологические характеристики документов. Наиболее полное описание метаданных поможет расширить возможности поискового механизма корпуса, что в свою очередь позволит расширить область различных исследований по изучению языка с использованием корпуса.

Для представления метаданных в текстовых документах разработан ряд международных стандартов и рекомендаций. Наиболее ранние рекомендации связаны с представлением метаданных в виде семейства форматов библиографических записей MARC, в частности его «русифицированная» версия RUSMARC. Данный формат позволяет довольно широко описывать текстовые данные (книги, журналы и пр.). Но использование форматов MARC при каталогизации изданий требует специальной подготовки, и, к сожалению, методические рекомендации, при использовании данного стандарта, довольно часто полностью не соблюдаются.

Специальные рекомендации EAGLES Preliminary recommendations on text typology (EAG-TCWG - TTYR/) разработаны для представления метаданных именно для использования в

электронных корпусах. Консультативной группой экспертов по техническим стандартам языка (Expert Advisory Group on Language Engineering Standards, EAGLES). Данные рекомендации наиболее полно отражают объективный набор метаданных, необходимый для представления данных электронного корпуса и его потенциальных приложений.

В рекомендациях выделено два класса факторов метаданных, влияющих на язык текстов: внешний и внутренний [Sinclair, 1996].

К внешним факторам относят:

- E.1. Origin – факторы, касающиеся происхождения текста, которые считают оказавшими влияние на структуру или содержание текста, такие как автор текста, его переводчик, правообладатель, язык оригинала текста, время написания и прочее.
- E.2. State – факторы, касающиеся внешнего вида текста, его расположение и отношение относительно нетекстовых материалов, такие как тип носителя текста, отношения с нелингвистической коммуникативной средой (изображения, диаграммы и прочие нетекстовые элементы), аспекты в области дизайна, оказавшие влияние на язык и прочее.
- E.3. Aims – факторы, касающиеся причин создания текста и его влияния на аудиторию, такие как размер аудитории, профессиональная область аудитории, связи автора с аудиторией, тип текста и прочее.

К внутренним факторам относят:

- I.1. Topic – предметная область текста
- I.2. Style — стилистические особенности текста.

Удобным вариантом представления модели метаданных является концепция Дублинского Ядра (Dublin Core) [Hillman, 2005], предназначенная для унификации метаданных широкого диапазона ресурсов. Простой набор Дублинского ядра (Dublin Core Metadata Element Set; DCMES) включает в себя 15 элементов: contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title, type [DCMES, 2012]. Данные 15 элементов были включены в стандарты IETF RFC 5013 (RFC5013) / (2007), ISO Standard 15836:2009 (ISO15836) / (2009), ANSI/NISO Standard Z39.85-2012 (NISOZ3985) / (2013). Квалифицированный набор данных, помимо перечисленных выше 15 элементов, может включать в себя дополнительно 54 элемента [DCMI Metadata Terms, 2012].

DCMI представлен в форматах DC-TEXT, HTML, XML, RDF. В настоящий момент Dublin Core активно развивается, разрабатываются новые элементы описания метаданных, расширяется сфера задач, в которых применим данный словарь.

В настоящей статье предложена достаточно полная модель описания метаданных в электронном корпусе, построенная с учетом рекомендаций EAGLES и DCMI.

1. Прототип корпус-менеджера

В настоящее время разработаны различные по функционалу системы «корпус-менеджеры», предназначенные для решения определенного круга задач. Можно указать ряд наиболее актуальных разработок для целей настоящей статьи. Восточно-армянский национальный корпус (<http://eanc.net>), на платформе которого располагается электронный корпус татарского языка «Туган Тел» (<http://web-corpora.net/TatarCorpus>), имеет 5 различных видов поиска: прямой поиск по слову, прямой поиск по лемме, обратный поиск, точный и неточный поиск. Функционал, реализованный в платформе Восточно-армянского национального корпуса, является базовым для корпус-менеджера, представленного в настоящей статье.

Национальный корпус русского языка (<http://ruscorpora.ru>) обладает функционалом, схожим с платформой Восточно-армянского национального корпуса. Отличительной особенностью платформы Национального корпуса русского языка является поддержка расширенного синтаксиса поисковых запросов при прямом поиске, а именно поддержка минус-слов, поиска по части слова и логических операторов.

На основе открытых решений авторами разработан новый корпус-менеджер [Невзорова и др., 2015а; Невзорова и др., 2015б; Nevzorova et al., 2015]. Функционал системы в основных функциях

соответствует функционалу платформ Национального корпуса русского языка и Восточно-армянского национального корпуса, но реализация конкретных задач позволяет говорить о существенном приросте эффективности данной платформы как со стороны обширности функционала, так и со стороны скорости взаимодействия пользователя с системой. Основными преимуществами корпус-менеджера, разработанного авторами, также являются готовая поддержка татарского языка и возможность быстрой интеграции с электронными корпусами других языков, в первую очередь, тюркских, поддержка произвольных морфологических формул, выявление логических ошибок, открытость используемых технологий.

Помимо разработки расширенного поискового функционала перед авторами стояли задачи оптимизации времени исполнения поисковых запросов (менее 1 секунды на запрос), поддержка произвольных морфологических формул с использованием операторов И, ИЛИ, НЕ и выставления приоритетов выполнения при помощи скобок, а также выявление логических ошибок в формулах. Примером логической ошибки является противоречивая формула «!(N|V),INF_1», которая означает «НЕ имя существительное (N) И НЕ глагол (V) И инфинитив, оканчивающийся на аффикс - *ырга* (INF_1)». Противоречивость данной формулы заключается в том, что все элементы, относящиеся к классу «INF_1» также относятся и к классу «V», но в первой части все элементы класса «V» исключаются, соответственно, результатов по данному поисковому запросу существовать не может.

2. Метаданные в корпус-менеджере

Исходными данными для системы корпус-менеджер являются текстовые документы с морфологической разметкой, которая автоматически производится с использованием модуля двухуровневого морфологического анализа татарского языка, реализованного в программном инструментарии РС-КИММО.

Объем татарского корпуса на декабрь 2015 года составляет более 82 млн словоформ. Корпус содержит тексты различных жанров (художественная литература, тексты СМИ, тексты официальных документов, учебная литература, научные публикации и др.) [Suleymanov, 2013].

В соответствии с рекомендациями EAGLES в наборе метаданных для описания текстового документа в электронном корпусе выделяются три блока (внешние факторы, внутренние факторы и технические метаданные).

Внешние факторы:

- Тип текста (оригинал или перевод);
- Название;
- Автор;

- Переводчик;
- Издание;
- Издательство;
- Язык;
- Дата создания;
- Объем в словах;
- Количество слов в русском файле;
- Источник оригинала;
- Источник перевода;
- Ключевые слова;
- Информация об авторском праве;
- Краткое описание;
- Примечание.

Внутренние факторы:

- Стиль;
- Категория/тематика;
- Место.

Технические метаданные:

- Номер (ID);
- Название исходного файла;
- Русский файл;
- Флаг проверки данных модератором.

Для представления документа в корпусе необходимо, чтобы этот документ имел обязательный (минимальный) набор метаданных, относящийся к внешним факторам, а именно:

- Название;
- Объем в словах;
- Номер (ID);
- Имя исходного файла.

Семантика документа представляется посредством внутренних факторов, таких как узкая тематика текста, хронотоп и др. Целью будущих исследований является разработка методов автоматического определения значений внутренних факторов метаданных, что позволит использовать корпусные документы в широком классе приложений, связанных с обработкой текстов.

3. Модель представления метаданных в корпус-менеджере

Так как метаданные документов могут состоять из неограниченного числа свойств и в будущем дополняться новыми данными, модель представления этих метаданных должна обеспечивать полноту и масштабируемость. Универсальным решением задачи, стоящей перед авторами, является использование семантической сети, представленной в модели представления данных RDF. Таким образом, информационной моделью метаданных корпуса является семантическая сеть. В качестве основополагающей модели было принято решение использовать рекомендации DCMI [DCMI Metadata Terms, 2012], в

которых описаны все метаданные, присутствующие у существующих документов электронного корпуса.

Использование модели RDF обусловлено в первую очередь тем, что модель данных, построенную на основе модели RDF можно очень просто масштабировать в ширину, добавляя новые объекты и связи, но не затрагивая основную массу данных. Именно простота модели RDF помогает строить понятные запросы к БД даже при высокой сложности поискового запроса и использовать при этом конечное количество таблиц, о чем говорится ниже в этой статье.

Часто бывает, что у документа отсутствует часть метаданных и её невозможно получить из текста документа. В этих случаях одним из возможных решений может быть использование внешних источников данных, таких как Open Library [openlibrary.org], Computer Science Bibliography, библиотека конгресса США [loc.gov]. Использование модели RDF может помочь в связывании этих внешних данных с элементами метаописания документов.

Такой подход к тому же позволит описать семантику выходных данных внутри HTML-документа, не теряя полноты данных, при помощи рекомендаций RDFa.

Заполнение метаданных является одной из самых ресурсоемких задач. В связи с этим авторами разрабатываются методы автоматизации этого процесса путем получения метаданных из документа и использования внешних источников данных. Решение данной задачи и является основной целью разработки структуры метаданных, которая позволила бы автоматизировать заполнение метаданных, исключить ошибки, возникающие из-за человеческого фактора.

Список свойств метаописания, представленный выше, позволяет определить основные объекты и связи между ними для представления метаданных. Они показаны на Рисунке 1 и в Таблице 1.

Таблица 1 – Основные объекты модели метаданных в корпус-менеджере

Свойства	Связь	Тип данных/ Класс
Номер (ID)	hasID	Число
Исходный файл, Объем в словах	hasSourceFile	SourceFile
Тип текста	hasType	Строка
Язык	hasLanguage	Строка
Название	hasName	Строка
Автор	hasAuthor	Author
Переводчик, Русский файл, Количество	hasTranslation	Translation

слов в русском файле, Источник перевода		
Стиль	hasStyle	Строка
Дата создания	hasCreationDate	Дата
Источник оригинала	hasSource	Строка
Издание, Издательство	hasEdition	Edition
Категория/ тематика	hasCategory	Category
Место	hasPlace	Place
Ключевые слова	hasKeyword	Строка
Информация об авторском праве	hasCopyright	Строка
Краткое описание	hasDescription	Строка
Примечание	hasNote	Строка
Флаг проверки данных модератором	isChecked	Логический

Некоторые метаданные корпуса Татарского языка объединяются в классы и имеют свойственные только им связи с объектами:

- SourceFile (Исходный файл):
 - hasFilename («имеет имя файла»);
 - hasContentSize («имеет размер содержимого»);
- Person (Человек):
 - hasName («имеет имя»);
 - hasSurname («имеет фамилию»);
 - hasMiddleName («имеет отчество»);
 - hasPseudonym («имеет псевдоним»);
 - hasWritingVariant («имеет вариант написания»);
 - hasBirthDate («имеет дату рождения»);

- hasDeathDate («имеет дату смерти»);
- hasBirthplace («имеет место рождения»);
- Organization (Организация):
 - hasName («имеет название»);
 - hasLegalName («имеет юридическое название»);
 - hasAddress («имеет адрес»);
 - hasWebsite («имеет веб-сайт»);
 - hasEmailAddress («имеет адрес электронной почты»);
 - hasFoundingDate («имеет дату основания»);
- Author (Автор):
 - isA («является», субъектом для связи является объект класса Person или Organization);
- Translation (Перевод):
 - hasTranslator («имеет переводчика», субъектом для связи является объект класса Person или Organization);
 - hasSourceDocument («имеет исходный документ», применим, если исходный документ является элементом множества Документы);
 - hasSourceFile («имеет исходный файл»);
- Edition (Издание):
 - hasPublishingHouse («имеет издательство», субъектом для связи является объект класса Organization);
 - hasPublishingDate («имеет дату издательства»);
- Category (Категория):
 - hasName («имеет название»);
 - hasParentCategory («имеет родительскую категорию», субъектом для связи является объект класса Category);
- Place (Место):
 - hasType («имеет тип», например, физическое или вымышленное);
 - hasName («имеет название»);
 - hasAddress («имеет адрес»);
 - hasCoordinates («имеет координаты»);

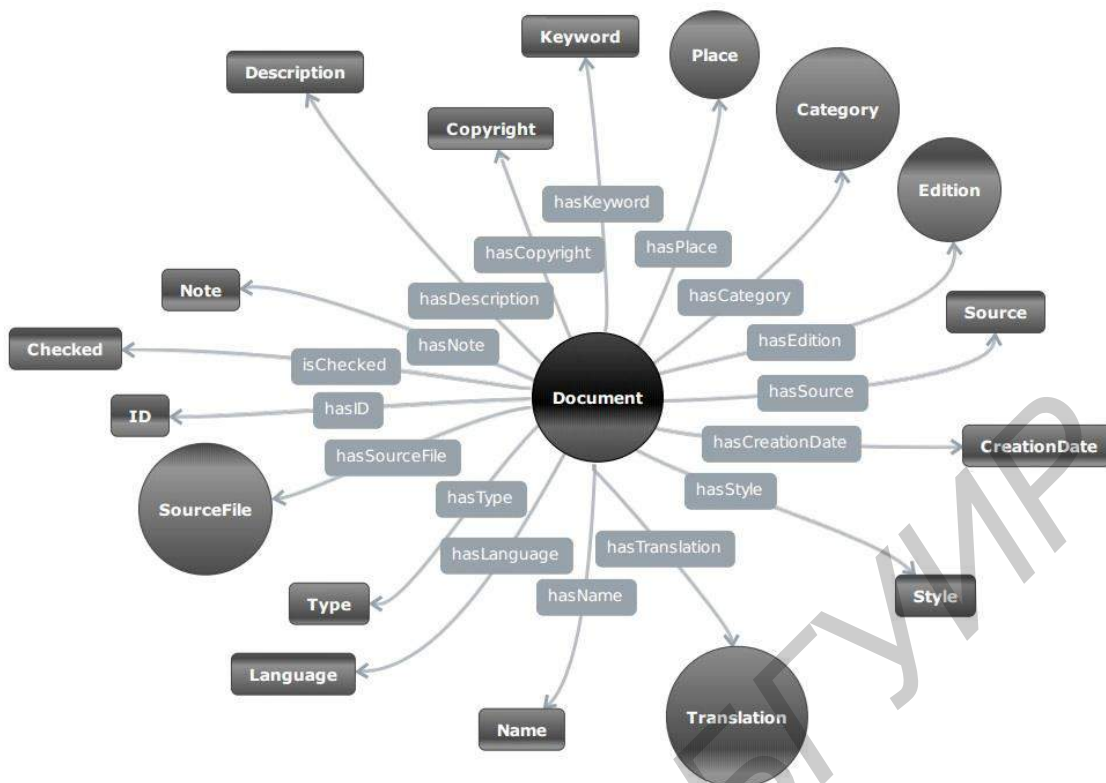


Рисунок 1 – Структура метаданных в корпус-менеджере

Заключение

4. Технические аспекты представления метаданных

Для функционирования корпус-менеджера используется следующее открытое программное обеспечение: веб-сервер Apache (или Nginx), интерпретатор PHP, СУБД MariaDB, in-memory хранилище Redis (кэширующий сервис), сервер очереди MemcacheQ. Всё программное обеспечение распространяется с открытым исходным кодом и может быть свободно использовано в некоммерческих целях.

Метаданные документов и сами документы хранятся в БД MariaDB, эта БД является реляционной, поэтому для хранения в ней данных, представляемых в модели RDF достаточно иметь одну таблицу *metadata* с полями *subject*, *predicate*, *object*. Эти три поля соответствуют стандарту RDF.

Для того, чтобы оптимизировать скорость выполнения запросов на выборку, в полях таблицы *metadata* хранятся лишь идентификаторы соответствующих субъектов, предикат и объектов. Сами субъекты, предикаты и объекты хранятся в отдельных таблицах и связаны с таблицей *metadata*.

Так, таблица *subjects* имеет поля *id*, *subject* и *type*, таблица *predicates* – *id*, *predicate*, таблица *objects* – *id*, *object*, *type*. К тому же в БД существует таблица *types*, которая содержит в себе типы субъектов и объектов для правильной работы с данными. Таблица *types* имеет поля *id*, *type*.

Предложенная в данной работе модель представления метаданных реализована в модуле управления контентом системы управления лингвистическими данными, работающей с электронным корпусом татарского языка.

Данная модель позволяет масштабировать метаданные без ущерба для существующих данных. При этом модель предоставляет возможность максимально полно описывать документы электронного корпуса. Использование семантической сети и модели представления данных RDF в качестве основы для модели представления метаданных в корпус-менеджере позволяет не только оптимизировать работу поискового модуля и находить документы по любому свойству, но и описывать семантику выходных метаданных непосредственно в HTML-документе. Эти и другие преимущества модели представления метаданных, описанной авторами в данной статье, позволяют говорить о том, что выбранное направление является оптимальным при решении поставленной задачи.

Использование семантических сетей для представления данных лингвистических корпусов позволяет покрыть широкий спектр задач. В статье описаны лишь некоторые из них. В перспективе предполагается использование этой же схемы для реализации морфологического анализатора и решения задачи снятия омонимии.

Общий подход к решению задач в поисковой системе по лингвистическому корпусу может

позволить использовать эту систему не только с электронным корпусом текстов на татарском языке, но и с корпусами других языков, не прибегая к значительным изменениям в системе.

На данный момент активно ведется изучение возможностей заполнения метаданных внутреннего фактора с помощью полуавтоматического или автоматического семантического аннотирования. Предполагается реализация данной возможности с помощью методов идентификации именованных сущностей и отнесения их к определенному семантическому классу.

Библиографический список

[Невзорова, 2015] Невзорова О.А., Мухамедшин Д.Р., Билалов Р.Р. Семантические аспекты представления и обработки поисковых запросов в системе корпус-менеджер // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2015): материалы II Междунар. научн.-техн. конф. (Минск, 19-21 февраля 2015 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР, 2015. – С. 439-444.

[Невзорова, 2015] Невзорова О.А., Мухамедшин Д.Р., Билалов Р.Р. Корпус-менеджер для тюркских языков: основная функциональность // Труды международной конференции «Корпусная лингвистика - 2015». – СПб.: С.-Петербургский гос. Университет, филологический факультет, 2015. – С. 344-350.

[Suleymanov, 2013] Dzhavdet Suleymanov, Olga Nevzorova, Ayrat Gatiatullin, Rinat Gilmullin, Bulat Khakimov National corpus of the Tatar language “Tugan Tel”: Grammatical Annotation and Implementation // Procedia - Social and Behavioral Sciences (2013), pp. 68-74.

[Невзорова, 2015] Nevzorova O., Mukhamedshin D., Bilalov R. Search Engine for the 'Tugan Tel' Tatar National Corpus: Main Decisions // Proceedings of the International Conference “Turkic Languages Processing” TurkLang-2015 — Kazan, 2015. - Pp. 236-244.

[Sinclair, 1996] J. McH. Sinclair, J. Ball. EAGLES Preliminary Recommendations on Text Typology EAG---TCWG---TTYP/P: [Электронный ресурс]. 1996. URL: <http://www.ilc.cnr.it/EAGLES96/texttyp/texttyp.html> (Дата обращения: 18.12.2015)

[Hillman, 2005] Diane Hillmann. Using Dublin Core: [Электронный ресурс]. 2005. URL: <http://dublincore.org/documents/usageguide/> (Дата обращения: 25.11.2015)

[DCMES, 2012] Dublin Core Metadata Element Set, Version 1.1 [Электронный ресурс]. 2012. URL: <http://dublincore.org/documents/dces/> (Дата обращения: 25.11.2015)

[DCMI Metadata Terms, 2012] DCMI Metadata Terms [Электронный ресурс]. 2012. URL: <http://dublincore.org/documents/dcmi-terms/> (Дата обращения: 25.11.2015)

SEMANTIC ASPECTS OF METADATA REPRESENTATION IN CORPUS MANAGER SYSTEM

Nevzorova O.A., Mukhamedshin D.R.,
Kurmanbakiev M.I.

*Research Institute of Applied Semiotics of the
Academy of Sciences of Tatarstan Republic,
Kazan, Russia*

onevzoro@gmail.com

damirmuh@gmail.com

write@marat.link

The article tells about the experience of the development model of metadata representation in

linguistic data management system (corpus manager), which is designed to work with corpus of texts of the Tatar language.

Introduction

At present time there are different recommendations (or standards) for annotating text document. In this paper we propose the structure of the document metadata based on the well-known standards such as EAGLES and DCMI.

Main Part

The volume of the Tatar corpus is more than 82 million word forms in June 2015. The corpus contains the texts of different genres (belles-lettres, mass media texts, texts of official documents, educational literature, scientific publications, etc.). Each corpus document has a meta-description: ID; the name of the source file; type of text (original or translation); language; name; author; translator; style; date of creation; the amount of words the source of original; the source of translation; edition; Publishing house; category/theme; place; keywords; information about copyright; short description and others.

Since the metadata can consist of an unlimited number of properties and will be complemented with new data in the future, the model of metadata representation must ensure completeness and scalability. A universal solution of the problem is to use a semantic network, represented in the RDF model of data.

List of properties of meta-description presented above allows you to define the main objects and relationships between them to represent metadata (see Table 1 and the list below).

Metadata of documents and the documents themselves are stored in the MariaDB database. This is relational database, so for storing therein the data submitted in the RDF model is sufficient to have one table ‘metadata’ with fields ‘subject’, ‘predicate’ and ‘object’. These three fields conform to the RDF standard.

Conclusion

The model of metadata representation proposed in this paper is implemented in content management module of linguistic data management system, which works with electronic corpus of the Tatar language.

The use of semantic networks for data representation in linguistic corpora allows covering a wide spectrum of problems. This article describes a few of them. In the future, it is planned to use the same scheme for the implementation of the morphological analyzer and solving the problem of disambiguation.

A general approach to solving problems in the search engine for linguistic corpora may allow the use of this system not only for Tatar corpus, but also for another corpora (another languages) without significant changes in the system.



УДК 004.822

СОЗДАНИЕ ЛЕКСИКОНА ОЦЕНОЧНЫХ СЛОВ РУССКОГО ЯЗЫКА РУСЕНТИЛЕКС

Лукашевич Н.В., Левчик А.В.

*Московский государственный университет им. М.В. Ломоносова,
г. Москва, Россия*

louk_nat@mail.ru

**RJ Games*

endight@gmail.com

В данной статье описан новый лексикон оценочных слов и выражений русского языка РуСентиЛекс. Данный лексикон был собран из нескольких источников: оценочные слова из тезауруса русского языка РуТез, сленговые слова из Твиттера и слова с позитивными или негативными ассоциациями (коннотациями) из корпуса новостей. Для многозначных слов, имеющих различную оценочную направленность (тональность) при использовании в разных значениях, установлены связи значений с соответствующими понятиями в тезаурусе русского языка РуТез, что может облегчить выбор соответствующего значения слова в конкретной предметной области или конкретном контексте.

Ключевые слова: анализ тональности; оценочная лексика; лексическое значение; тезаурус; лексикон; машинное обучение

Введение

Автоматический анализ тональности текстов используется во многих практических приложениях, включая анализ пользовательских отзывов, постов в социальных сетях, новостных статей и др. Важным компонентом таких систем являются словари оценочных слов и выражений, которые могут быть использованы как в инженерных подходах к анализу тональности, основанных на словарях и правилах [Taboada et al., 2011], так и быть источниками признаков в подходах, основанных на машинном обучении [Mohammad et al., 2013; Severyn, Moschitti, 2015].

Для английского языка было создано большое количество различных словарей, которые разрабатывались экспертами [Wilson et al., 2005] или в процессе краудсорсинга [Mohammad, Turney, 2013]. Для других языков использовались в основном различные подходы автоматического порождения словарей оценочной лексики [Chetviorkin, Loukachevitch, 2012; Perez-Rosas et al., 2012; San Vicente et al., 2014; Yang et al., 2013].

Известно, что состав тональных словарей в значительной степени зависит от предметной области, поэтому множество работ посвящено

извлечению или настройке тональных словарей на конкретную предметную область [Blitzer et al., 2007; Choi, Cardie 2009; Lau et al., 2011].

При этом авторы других работ показывают [Mansour et al., 2013], что комбинация обучающих данных из нескольких областей в подходах, основанных на машинном обучении, повышает качество классификатора текстов по тональности в каждой из областей. Это означает, что существует относительно стабильное множество общих (межпредметных) оценочных слов с относительно стабильной оценочной ориентацией. Кроме того, как было показано в [Mohammad et al., 2013], признаки для тональной классификации текстов, порождаемые на основе существующих словарей, полезны для улучшения качества работы систем анализа тональности, снижают зависимость от обучающих данных. Таким образом, для любого естественного языка полезно наличие словаря оценочной лексики, созданного вручную.

При этом создание такого общего словаря оценочной лексики не так просто осуществить, поскольку, несмотря на то, что есть очевидные оценочные слова (*хороший*, *плохой* и т.п.), для анализа большого объема лексики с этой точки зрения может понадобиться просмотреть много слов,

поэтому оправданным является подход на основе частичной автоматизации этого процесса.

Данная статья описывает новый лексикон оценочных слов и выражений русского языка РуСентиЛекс. Лексикон был создан автоматизированно, на основе комбинации автоматических методов извлечения оценочных слов из текстов и последующего их ручного просмотра и описания. Созданный словарь содержит более десяти тысяч русских слов и выражений, выражающих некоторую оценку. Для многозначных слов, которые имеют в разных значениях разную тональность, указаны отсылки на соответствующие понятия тезауруса русского языка РуТез, что может помочь разрешать тональную неоднозначность слова в конкретной предметной области или контексте.

Статья организована следующим образом. В первой секции представлены близкие работы по созданию словарей оценочной лексики. Во второй секции описывается структура созданного словаря оценочной лексики русского языка РуСентиЛекс. Третий раздел представляет методы и источники, на основе которых был собран данный лексикон.

1. Обзор близких работ

Вручную созданные словари оценочной лексики могут быть представлены в виде простых списков слов с некоторыми атрибутами. Также разметка тональности слов может быть выполнена с учетом значений слов, так, что каждое значение слова получает свою отдельную оценку тональности.

Один из известных словарей оценочной лексики английского языка MPQA [Wilson et al., 2005] был составлен из нескольких источников (ручных и автоматически порожденных словарей оценочных слов) и содержит свыше 8000 отдельных слов. Слова в словаре размечены метками полярности (позитивный, негативный или нейтральный) и оценочные слова снабжены пометами силы оценочного содержания (сильный или слабый).

Вручную созданный английский оценочный словарь AFINN [Nielsen, 2009] был специально дополнен ругательными и сленговыми словами для адаптации его к автоматическому анализу сообщений в социальных сетях. Он содержит около 2400 слов, помеченных числовым весом полярности, изменяющегося от -5 (очень негативный) до +5 (очень позитивный).

В работе [Baccianella et al., 2010] описывается словарь SentiWordNet, который основан на тезаурусе английского языка WordNet. Он получен в результате автоматической разметки синсетов (=наборов синонимов) WordNet в результате чего каждому синсету поставлено в соответствие три числа, которые обозначают долю позитивности, негативности и нейтральности слов из данного синсета. Таким образом, разные значения одного и

того же слова могут иметь различные оценки тональности.

В словаре SenticNet [Cambria et al., 2010] слова и выражения размечены по четырем измерениям: приятность (pleasantness), внимание (attention), восприимчивость (sensitivity), склонность (aptitude). Для получения числовых оценок авторы использовали оценочные слова и соответствующие веса, определенные в Hourglass of Emotions [Cambria et al., 2012] как начальное множество для получения оценок тональности для остальных понятий. Авторы словаря данного словаря уделяют особое внимание выражениям, в состав которых входят градуальные прилагательные, которые не имеют априорной тональности (*большой* и др.). Последняя версия SenticNet содержит около 30 тысяч слов и выражений.

В работе [Zasko-Zielinska et al., 2015] описывается процесс аннотирования по тональности лексических единиц, описанных в польском ворднете plWordNet. Лексическая единица в данном случае представляет собой пару (лемма, номер_значения). Для примерно 30 тысяч лексических единиц (прилагательных и существительных) были указаны полярность (позитивно, негативно, нейтрально) и оценка интенсивности (сильная и слабая). Кроме того, лексическим единицам были приписаны идентификаторы основных эмоций: радость (joy), доверие (trust), страх (fear), удивление (surprise), грусть (sadness), отвращение (disgust), гнев (anger), предвкушение (anticipation). Из размеченного набора лексических единиц 30% были позитивными или негативными.

Лексикон оценочных ассоциаций (Word-Emotion Association) Исследовательского центра Канады (NRC Canada) был создан с помощью краудсорсинга и содержит слова и выражения, которые имеют ассоциации с тональностью и определенными эмоциями [Mohammad, Turney, 2013]. Эмоциональная разметка осуществлялась по шести категориям.

В работе [Chetviorkin, Loukachevitch, 2012] описывается подход к автоматическому созданию словаря оценочной лексики в области товаров и услуг для русского языка ProductSentiRus¹. Словарь ProductSentiRus был получен применением обученной модели к наборам отзывов в нескольких предметных областях. Словарь представлен как список 5 тысяч слов, упорядоченных по мере снижения вычисленной вероятности их оценочности без указания позитивной или негативной тональности.

2. Структура лексикона РуСентиЛекс

По своей структуре лексикон РуСентиЛекс представляет собой упорядоченный по алфавиту список слов и выражений. Он содержит следующие

¹ <http://www.cir.ru/SentiLexicon/ProductSentiRus.txt>

типы русскоязычных слов, значения которых связаны с тональностью:

– слова (выражения) литературного русского языка, для которых хотя бы одно значение имеет оценочный компонент, что означает, что слово в этом значении либо явно выражает отношение к обсуждаемому объекту (*отличный*), либо передается через выражаемую эмоцию (*грустно*);

– слова (выражения), не передающие оценочное отношения автора, но имеющие положительную или отрицательную коннотацию [Feng et al., 2013], например, *безработица, терроризм, болезнь, спам* и др.;

– сленговые и ругательные слова из Твиттера.

Все лексические единицы, описанные в РуСентиЛекс и их значения, рассматриваются с трех точек зрения. Во-первых, указывается полярность слова: позитивная или нейтральная; возможны также приписывание пар полярностей. Во-вторых, проставляется источник тональности: прямо выраженная оценка, эмоция или коннотация.

В-третьих, представлены тональные различия между значениями многозначного слова. Если все значения многозначного слова имеют одну и ту же тональность во всех значениях, то указывается просто тональность слова. Если слово имеет различные характеристики тональности в своих разных значениях, то описываются особенности каждого значения. Для идентификации значений устанавливается ссылка на понятия тезауруса русского языка РуТез.

Тезаурус РуТез² представляет собой лингвистическую онтологию для автоматической обработки текстов, т.е. онтологию, в которой большинство понятий введены на основе значений реально существующих языковых выражений. Опубликованная версия тезауруса РуТез содержит около 100 тысяч русскоязычных слов и выражений.

Если сравнивать с лингвистическими ресурсами типа WordNet, то можно отметить, что тезаурус РуТез представлен как единая сеть понятий, не разделяемая по отдельным частям речи. В качестве текстовых входов понятия могут быть слова разных частей речи, а также словосочетания различной синтаксической структуры (именные группы, глагольные группы, группы прилагательного и др.). Каждое понятие имеет уникальное имя. Понятия соединяются между собой несколькими типами концептуальных отношений [Loukachevitch, Dobrov, 2014].

При подготовке словаря РуСентиЛекс было замечено, что в русском языке имеется значимое количество слов, которые во всех своих зафиксированных в тезаурусе значениях имеют одну и ту же тональность (например, *грязный*). Поэтому было принято решение не расписывать подробно тональность таких слов по отдельным значениям, а

указывать общую тональность слова. Таким образом, значения таких слов в тезаурусе могут пополняться, но тональность у них уже приписана.

Словарь РуСентиЛекс хранится в простом текстовом формате, подобном формату словаря МРQA [Wilson et al., 2005]. Каждой единице словаря, которая может быть словом, словосочетанием или лексической единицей (т.е. парой слово-понятие тезауруса РуТез) приписываются следующие атрибуты:

- слово или фраза,
- часть речи,
- слово или фраза, в которой каждое слово стоит в лемматизированной форме, что необходимо для сопоставления фраз с текстами, в которых фраза может стоять в разных словоизменительных формах,
- тональность. Тональность может быть позитивная (*positive*), негативная (*negative*), нейтральная (*neutral*) или двойная, например, *positive/negative*. В последнем случае такая отметка означает, что слово (фраза) обычно употребляется с какой-то оценкой, но эта оценка в значительной степени зависит от контекста употребления слова;
- источник тональности (явно выраженная оценка, эмоция, или факт);
- отсылки к понятиям тезауруса РуТез для значений тех слов, которые имеют различающуюся тональность в разных значениях. Для этого производится указание на имя соответствующего понятия в тезаурусе. Отметим, что в таких случаях описывается тональность для всех значений многозначного слова.

Например, слово *пресный* имеет три различных значения в тезаурусе РуТез. Два из них (значение как безвкусный о еде и значения неинтересный) имеют негативную тональность. Еще одно значение слова, связанное с пресной водой, имеет позитивную коннотацию, поскольку обладание пресной водой – это хорошо, ее истощение – это плохо и т.п.

Таким образом, описание значений слова *пресный* выглядит следующим образом (метки в кавычках соответствуют именам понятий в тезаурусе РуТез):

пресный, Adj, *пресный*, negative, emotion, "НЕВКУСНЫЙ"
пресный, Adj, *пресный*, negative, opinion, "НЕИНТЕРЕСНЫЙ"
пресный, Adj, *пресный*, positive, fact, "ПРЕСНАЯ ВОДА"

Другое оценочное русское слово *грязный* имеет два значения в тезаурусе РуТез, но оно описывается в РуСентиЛекс без ссылок на значения, поскольку оба этих значения являются негативными:

грязный, Adj, *грязный*, negative, opinion.

Слова-кандидаты на включение в лексикон были извлечены автоматически (см. п. 3). Далее эксперты-

² <http://www.labinform.ru/pub/ruthes/index.htm>

лингвисты анализировали употребление каждого слова в современных новостных текстах. Новостные тексты выбраны потому, что они адресованы максимальной аудитории современных русскоязычных людей и поэтому в среднем передают норму современного русского языка.

Например, при анализе слова *аккуратист* (*аккуратный человек*) некоторым экспертам казалось, что употребление этого слова несет негативную оценку. Но проанализированные контексты употребления этого слова показали, что оценка скорее позитивная, например:

Страховщики заплатят штраф за отсутствие скидок водителям-аккуратистам (Вести-ФМ 29.10.2015);

Шамардин: Это интеллигентнейший человек, который показал себя с первых дней своего обучения. Аккуратист во всем, в одежде, в поведении (Спорт FM, 13.10.2015).

В Таблице 1 приведены количественные характеристики разных категорий слов и выражений в словаре РуСентиЛекс.

Таблица 1. Количественные характеристики лексикона РуСентиЛекс

Категория словарного входа	Количество
Негативный	9744
Positive/negative	241
Позитивный	3585
Нейтральные	1394
Фактические	4607
Слова из Твиттера, отсутствующие в тезаурусе РуТез	798
Словосочетаний	2545
Всего разных текстовых входов	10467
Всего значений	14492

3. Источники для порождения лексикона

Словарные входы лексикона РуСентиЛекс были получены автоматизированно, на основе извлечения оценочной лексики из нескольких источников.

3.1. Использование существующих списков оценочных слов в конкретных предметных областях

Для получения начального списка оценочных слов и соответствующего списка понятий из тезауруса РуТез были использованы списки оценочных слов, которые были составлены в конкретных предметных областях в рамках лингвистико-инженерного подхода к анализу тональности.

Лингвистико-инженерный подход включает в себя создание словарей оценочных слов и выражений с проставленными весами оценочности, которые обычно обозначают положительными числами – положительную тональность, а отрицательными числами – негативную

тональность, а также составления списка правил, которые комбинируют эти оценки между собой и со словами-операторами, усиливающими (*очень*) или модифицирующими (*не, нет*) исходную оценку слов [Kuznetsova et al., 2013].

Слова и выражения, упомянутые в этих списках были сопоставлены с тезаурусом РуТез, и все понятия, текстовые входы которых сопоставились, были извлечены для дальнейшего анализа экспертом. Также была проставлена средняя оценка понятия по его текстовым входам.

Задачей эксперта было проверить тональную оценку слова, проставленную в исходном списке, уточнить оценку значений слова, если они отличаются, а также проверить полноту создаваемого словаря за счет синонимов и текстовых входов близких понятий тезауруса.

3.2. Извлечение слов с положительными и отрицательными коннотациями

Неоценочные слова с оценочными коннотациями, упомянутые в материалах прессы, обычно ссылаются на негативные или позитивные явления в общественной или личной жизни человека. Оценочность употребления таких слов сильно зависит от контекста. Например, рост безработицы, упомянутый в статистическом отчете, является просто фактографической информацией. Если же это же выражение упомянуто в аналитическом документе, оценивающим последствия некоторых решений, то выражение будет оценочным, отрицательным.

Для автоматического извлечения слов-фактов было сделано предположение, что эти слова могут упоминаться в характерных контекстах, поскольку с негативными фактами обычно борются, уничтожают, а позитивные явления поддерживаются и защищаются. Поэтому были созданы списки шаблонов для выявления слов с негативными (35 шаблонов) и позитивными (20 шаблонов) коннотациями. Эти шаблоны были применены к новостной коллекции размером два миллиона статей.

Извлеченные слова и выражения были разделены на три класса: слова с негативными коннотациями (большинство вхождений относилось к негативным шаблонам), слова с позитивными коннотациями (большинство найденных шаблонов были позитивными), и нейтральные слова, которые встречались в обоих типах шаблонов. Например, наиболее часто встречались в негативных шаблонах такие слова, как: *коррупция, терроризм, преступление, экстремизм, наркотик, инфляция, барьер, угроза, кризис, безработица* и др. В результате этой процедуры было извлечено 4879 негативных слов и выражений, 3249 позитивных и 596 нейтральных с частотой появления в шаблонах более пяти раз.

Очевидно, что нейтральных слов должно быть значительно больше. Было сделано предположение,

что верхние уровни иерархии тезауруса нейтральны. Поэтому, двигаясь по связям с самого верхнего уровня понятий, соответствующие им текстовые входы, добавлялись в список нейтральных слов, до тех пор, пока не было достигнуто понятия, с негативным или позитивным текстовым входом. В результате список нейтральных слов стал включать 4434 элемента.

После этого все размеченные слова и словосочетания были использованы в качестве исходных слов для так называемой процедуры распространения меток (*label propagation*) [Zhu, Ghahramani, 2002] по отношениям тезауруса RuТез. Данная процедура использовала предположения о структуре тезауруса. Так, распространение оценки вверх (по отношениям гипероним, целое) имело меньший вес, чем по отношениям вниз.

Полученные слова и фразы были проанализированы экспертом для пополнения лексикона RuСентиЛекс.

3.3. Использование модели машинного обучения с учителем для извлечения оценочных слов из Твиттера.

Для анализа текстов социальных сетей недостаточно использовать оценочные слова, употребляемые в литературном русском языке. Поэтому были предприняты специальные усилия, чтобы извлечь наиболее вероятные оценочные слова из Твиттера. Для этого была применен метод машинного обучения, модель извлечения оценочных слов для которого была построена на размеченных данных отзывов о фильмах [Chetviorkin, Loukachevitch, 2012]. Модель хорошо показала себя при переносе на другие предметные области. Размер коллекции русскоязычных твитов, использовавшейся в этой процедуре, составлял более миллиона неразмеченных твитов.

Применяемая модель основана на использовании статистических данных, полученных из трех коллекций: коллекции с высокой концентрацией оценочных слов (А), контрастной коллекции той же предметной области с низкой концентрацией оценочных слов (В), и с контрастной общей коллекцией (использовалась новостная коллекция). В данном случае в качестве коллекции (А) использовался набор твитов, в котором нашлось хотя бы одно слово из автоматически собранного словаря ProductSentiRus [Chetviorkin, Loukachevitch, 2014]. Остальные твиты использовались в качестве коллекции (В) с низким содержанием оценочных слов.

В результате из твитов были извлечены слова, упорядоченные по мере снижения их вероятности быть оценочными словами [Chetviorkin, Loukachevitch, 2014]. Точность данного списка на уровне первой 1000 слов была оценена как 79.9%. Из первых пяти тысяч слов этого списка были исключены уже ранее известные оценочные слова, а

остальные были рассмотрены лингвистом для включения в лексикон RuСентиЛекс.

Заключение

В данной статье был описан новый лексикон оценочных слов и выражений русского языка RuСентиЛекс. Данный лексикон был собран из нескольких источников: оценочные слова из тезауруса русского языка RuТез, сленговые слова из Твиттера и слова с позитивными или негативными ассоциациями (коннотациями) из корпуса новостей.

Для многозначных слов, имеющих различную оценочную направленность при использовании в разных значениях, установлены связи значений с соответствующими понятиями в тезаурусе русского языка RuТез, что может облегчить выбор соответствующего значения слова в конкретной предметной области или конкретном контексте. Все единицы лексикона расклассифицированы по четырем категориям тональности и трем источникам тональности (мнение, эмоция, факт).

Созданный лексикон может служить основой для создания оценочных словарей в конкретной предметной области или использоваться для порождения признаков в подходах к анализу тональности, основанных на машинном обучении.

Работа выполнена при частичной поддержке фонда РФФИ, грант 14-07-00682.

Библиографический список

- [Baccianella, 2010] Baccianella, S., Esuli, A., Sebastiani F. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. Proceedings of LREC-2010, Vol. 10, P. 2200-2204.
- [Blitzer, 2007] Blitzer, J., Dredze, M., Pereira, F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. Proceedings of ACL-2007, P. 440-447.
- [Cambria, 2012] Cambria E., Livingstone A., and A. Hussain. The hourglass of emotions. Cognitive Behavioural Systems, Lecture Notes in Computer Science, vol. 7403, Springer, P. 144-157
- [Cambria, 2010] Cambria E., Hussain A., Havasi C., Eckl C. Sentic computing: Exploitation of common sense for the development of emotion-sensitive systems. Springer, LNCS, vol. 5967, P. 148-156.
- [Chetviorkin, 2012] Chetviorkin, I., Loukachevitch N. Extraction of Russian Sentiment Lexicon for Product Meta-Domain. Proceedings of COLING-2012, P. 593-610.
- [Chetviorkin, 2014] Chetviorkin, I., Loukachevitch, N. Two-Step Model for Sentiment Lexicon Extraction from Twitter Streams. Proceedings of WASSA workshop in conjunction with ACL-2014, P. 67-72.
- [Choi, 2009] Choi Y., Cardie C. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. Proceedings of EMNLP '09, P. 590-598.
- [Feng, 2013] Feng S., Jun S. Kang, Polina Kuznetsova, Yejin Choi. Connotation lexicon: a dash of sentiment beneath the surface meaning. Proceedings of ACL-2013, P. 1774-1784.
- [Kuznetsova, 2013] Kuznetsova, E. S., Loukachevitch, N. V., Chetviorkin, I. I. Testing rules for a sentiment analysis system. Proceedings of International Conference Dialog-2013, P. 71-80.
- [Lau, 2011] Lau R., Lai C., Bruza P., Wong K. Pseudo Labeling for Scalable Semi-supervised Learning of Domain-specific Sentiment Lexicons. Proceedings of 20th ACM Conference on Information and Knowledge Management.
- [Loukachevitch, 2014] Loukachevitch, N., Dobrov, B. RuThes Linguistic Ontology vs. Russian Wordnets. Proceedings of Global WordNet Conference GWC-2014, Tartu.

[Mansour, 2013] Mansour, R., Refaei, N., Gamon, M., Abdul-Hamid, A., Sami, K. Revisiting The Old Kitchen Sink: Do We Need Sentiment Domain Adaptation? Proceedings of RANLP-2013, P. 420-427.

[Mohammad, 2013a] Mohammad, S. M., Turney, P. D. Crowdsourcing a word-emotion association lexicon. Computational Intelligence. – 2013. – 29(3). – P. 436-465.

[Mohammad, 2013b] Mohammad, S. M., Kiritchenko, S., Zhu, X. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. Proceedings of Second Joint Conference on Lexical and Computational Semantics (SEM), Vol. 2, P. 321-327.

[Nielsen, 2011] Nielsen F. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, P. 93-98.

[Perez-Rosas, 2012] Perez-Rosas, V., Banea, C., Mihalcea, R. Learning Sentiment Lexicons in Spanish. Proceedings LREC-2012, P. 3077-3081.

[San Vicente, 2014] San Vicente, I., Agerri, R., Rigau, G., Sebastián, D. S. Simple, Robust and (almost) Unsupervised Generation of Polarity Lexicons for Multiple Languages. Proceedings of EACL-2014, 88.

[Severyn, 2015] Severyn, A., Moschitti, A. On the automatic learning of sentiment lexicons. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2015).

[Taboada, 2011] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M. Lexicon-based methods for sentiment analysis. Computational linguistics. – 2011. – 37(2). – P. 267-307.

[Wilson, 2005] Wilson, T., Wiebe, J., Hoffmann, P. Recognizing contextual polarity in phrase-level sentiment analysis. Proceedings of the conference on human language technology and empirical methods in natural language processing, P. 347-354.

[Yang, 2013] Yang, A. M., Lin, J. H., Zhou, Y. M., Chen, J. Research on building a Chinese sentiment lexicon based on SO-PMI. Applied Mechanics and Materials. – 2013. – Vol. 263. – P. 1688-1693.

[Zasko-Zielinska, 2015] Zasko-Zielinska M., Piasecki M., Szpakowicz S. A Large Wordnet-based Sentiment Lexicon for Polish. Proceedings Recent Advances in Natural Language Processing Conference (RANLP-2015), P. 721-728.

[Zhu, 2002] Zhu X., Ghahramani, Z. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Ca№ 22(4).

CREATING RUSSIAN SENTIMENT LEXICON

Loukachevitch N.V., Levchik A.V.

*Lomonosov Moscow State University,
Moscow, Russia*

louk_nat@mail.ru

RJ Games, Moscow, Russia

endight@gmail.com

The paper describes the new Russian sentiment lexicon - RuSentiLex. The lexicon was gathered from several sources: opinionated words from domain-oriented Russian sentiment vocabularies, slang and curse words extracted from Twitter, objective words with positive or negative connotations from a news collection. The words in the lexicon having different sentiment orientations in specific senses are linked to appropriate concepts of the thesaurus of Russian language RuThes. All lexicon entries are classified according to four sentiment categories and three sources of sentiment (opinion, emotion, and fact). The lexicon can serve as the first version for the construction of domain-specific sentiment lexicons and be used for feature generation in machine-learning approaches.

Introduction

Automatic sentiment analysis is useful in many practical applications, such as analysis of users' reviews, posts in social networks, newspaper articles, etc. Sentiment lexicons are important components of sentiment analysis systems. They can be applied in lexicon-based approaches or be sources of features in the machine-learning framework

In this paper we present a new manually created general Russian Sentiment Lexicon – RuSentiLex. The lexicon contains about 10 thousand Russian sentiment-related words and expressions. Ambiguous words that have different sentiment polarity in different senses are provided with links to appropriate concepts of the Thesaurus of Russian language, which can help disambiguate sentiment ambiguity in specific domains or contexts.

Main Part

The RuSentiLex lexicon is an alphabet-ordered Russian sentiment vocabulary. It contains the following types of Russian sentiment-related words:

- words from general Russian for that at least one sense has a positive or negative polarity what means that it conveys negative/ positive attitude (*excellent*) or negative/positive emotion, (*sadness*);
- non-opinionated words with negative or positive connotations such as *unemployment, terrorism, disease, cancer, explosion, etc.*;
- slang and curse words from Twitter.

All words and their senses are considered from three points of views: polarity (negative, positive, or neutral); source (attitude, emotion, or non-opinionated fact); sentiment differences between word senses. If a word has different sentiment orientations or sources in its different senses then links between the senses and RuThes concepts are established.

RuSentiLex lexicon was obtained from several sources using semi-automatic techniques.

Conclusion

In the paper we describe the new Russian sentiment lexicon - RuSentiLex. The current size of the lexicon is about ten thousand words and phrases. The lexicon was gathered from several sources: opinionated words from general Russian thesaurus RuThes, slang and curse words extracted from Twitter, objective words with positive or negative connotations from news.



УДК 621.391

МИКРОСЕРВИСНЫЙ ПОДХОД К ПРОЕКТИРОВАНИЮ РЕЧЕВЫХ ИНТЕРФЕЙСОВ

Житко В.А. *, Лобанов Б.М. **

**Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

zhitko.vladimir@gmail.com

***Объединённый институт проблем информатики НАН Беларуси, Минск*

lobanov@newman.bas-net.by

В статье описывается модель построения прикладных программных систем с речевым интерфейсом, основанная на применении сервис-ориентированной архитектуры и микросервисов. Такой подход обеспечивает минимизацию расходов на интеграцию различных по реализации и подходам компонентов в единую прикладную программную систему.

Ключевые слова: речевой интерфейс; сервис-ориентированная архитектура, микросервисы.

Введение

В связи с бурным ростом рынка мобильных и встроенных приложений и систем, разработкам в сфере упрощения ввода текстового запроса информации уделяется большое внимание. Наиболее перспективными в этой ситуации являются разработки, связанные с распознаванием речи, – современная альтернатива ручному вводу с клавиатуры. Кроме того, голосовой ввод команд и вопросов даёт возможность использовать систему в качестве источника справочной информации одновременно с основной деятельностью (получение справки при выполнении технических работ, управления автомобилем и т.д.).

Использование одновременно с распознаванием речевого синтеза позволяет снизить нагрузку на зрительную систему пользователя путём восприятия результатов работы системы не через графический интерфейс, а посредством голоса.

Однако в этом направлении существует масса различных проблем:

- большая вычислительная сложность и огромный объем баз данных, необходимых для корректного решения задачи распознавания речи;
- большое разнообразие используемых мобильных устройств и интернет-платформ.

Одним из возможных решений преодоления указанных трудностей является перенос на удалённый сервер систем распознавания и синтеза

речи. В связи с этим наиболее перспективным выглядит применение активно развивающихся в настоящее время облачных интернет-технологий, позволяющих легко наращивать по мере необходимости производительность и объем используемых баз данных.

Целью данной работы является разработка моделей и средств построения естественно-языковых и речевых интерфейсов к прикладным системам, путём интеграции различных существующих компонентов в единую систему на основе применения микросервисов. В качестве основной модели была принята сервис-ориентированная архитектура программной системы [Barry, D. 2003] с применением подхода использования микросервисов [Chris, R. 2014] [Sam, N. 2015]. Использование микросервисов при проектировании позволяет без дополнительных затрат использовать компоненты построенные на базе облачных технологий. На сегодняшний день существует ряд справочных систем обладающих речевым интерфейсом (Siri, Voice Actions и др.), однако все они являются закрытыми коммерческими проектами и использовать данные наработки в других проектах не представляется возможным.

1. Сервис-ориентированная архитектура

В качестве основного подхода к проектированию приложений с естественно-языковым интерфейсом была выбрана сервис-ориентированная архитектура. Такой выбор обусловлен необходимостью

интеграции большого количества компонентов, участвующих в работе естественно-языкового интерфейса. На сегодняшний день существует множество разработок в сфере естественно-языкового общения, будь то синтез речи, распознавание, обработки и пр. Это позволяет построить из представленных компонентов системы, обладающие естественно-языковым интерфейсом, а выбранный подход обеспечивает лёгкую интеграцию компонент, построенных на отличающихся принципах за счёт слабой связанности самих компонент. Это позволяет также беспрепятственно менять компоненты на аналогичные, не перестраивая всего приложения.

Сервис-ориентированная архитектура (SOA, англ. service-oriented architecture) — модульный подход к разработке программного обеспечения, основанный на использовании распределённых, слабо связанных (англ. loose coupling) заменяемых компонентов, оснащённых стандартизированными интерфейсами для взаимодействия по стандартизированным протоколам.

2. Использование микросервисов

В качестве подхода при декомпозиции логики приложения на сервисы был выбран подход микросервисов.

Микросервисы — это архитектурный стиль, при котором сложное приложение разбивается на мелкие, независимые процессы, взаимодействующие через методы API, независимых от языка.

Архитектура приложения построенного на микросервисах обладает рядом свойств. Вот некоторые из них:

- разбиение на компоненты через сервисы;
- сервисы определяются бизнес-задачами;
- логика реализована в методах сервиса, а для передачи сообщений используются простейшие каналы передачи.

Каждый микросервис решает задачи, связанные с единственным или несколькими неразделимыми объектами домена (The Single Responsibility Principle [DeMarco, 1979], [PageJones, 1988]). При этом любой сервис, по своей природе, открыт для расширения, но его трудно модифицировать, особенно не имея доступа к исходному коду (The Open Closed Principle [Bertrand Meyer, 1988]). В силу того, что сервисы взаимодействуют между собой по контракту, то имеется возможность заменять одну реализацию сервиса, на другую (The Liskov Substitution Principle [Barbara, L. 1988]), если только он соответствует контракту. При проектировании микросервисов, также как и в объектно-ориентированном программировании, имеет смысл по возможности разделять сервисы на более мелкие, тем самым реализуя принцип разделения интерфейсов (Interface segregation principle [Martin, Robert 2002]). Ну и безусловно, используемые

микросервисы не должны зависеть от других используемых компонент (The Dependency Inversion Principle [Robert C. 1996]).

Использование микросервисов обладает следующими преимуществами:

- масштабируемость - микросервисную архитектуру очень легко масштабировать;
- отказоустойчивость - надёжность достигается за счёт запуска дополнительных процессов;
- обновляемость - малая связанность сервисов и их относительная простота делает процесс тестирования и обновления технологий довольно простой задачей.

Основная трудность при использовании архитектуры микросервисов — разбиение бизнес-логики на независимые сервисы и организация взаимодействия между ними.

2.1. Распознавание речи

Для распознавания речевого сигнала в системе используется компонент, в основе которого лежит сервис распознавания речи, разработанный компанией Google.

На сегодняшний день компания Google является лидером по предоставлению облачных технологий распознавания речи [Manjoo F., 2011]. В течение последних пяти лет активно развивалась облачная технология распознавания речи Google Voice, и к настоящему времени существуют технологии распознавания речи для большинства европейских языков, включая русский, а также японский и китайский языки. Одним из немаловажных компонентов системы распознавания речи Google Voice является обучающая выборка звукозаписей человеческого голоса. Для системы Google Voice источником таких записей являются различные сервисы, предоставляемые Google, использующие речевые технологии, к ним относятся система распознавания речи и команд в системе Android, сервис диктовки писем Google Mail, телефонная справочная система Goog411 и др. [Singhal, A., 2011]. Таким образом, обучающая выборка постоянно пополняется новыми образцами голосов с их особенностями, как произношения, так и эффектами, вносимыми техническими особенностями записи и передачи голоса на различных устройствах. К примеру, в 2011 г. обучающая выборка для английского языка составляла примерно 230 млрд записей слов извлеченных из реальных запросов [Enge, E. 2011]. Для обработки таких объемов информации требуется около 70 лет процессорного времени, однако с использованием облачной технологии Google время сокращается до одного дня [Singhal, A., 2011].

Используемая модель распознавания включает в себя три части: акустическую, лексическую и языковую модели [Enge, E. 2011]. Акустическая

модель ответственна за распознавание фонем, она учитывает все возможные варианты произношения, а также другие особенности, такие как тип используемого микрофона (качество записи), фоновые шумы, возраст и пол говорящего и многое другое. Наиболее важным в данном случае является объем обучающей выборки, чем он больше, тем лучше будет результат распознавания. В лексической модели фонемы объединяются в слова на основе словарей, в которых указаны различные варианты произношения слов. Языковая модель объединяет слова, используя статистический подход. На основе анализа поисковых запросов, текстов Интернета выделяются вероятности взаимного расположения слов в предложении.

В составе разработанной прикладной программы в процессе распознавания речи задействованы три микросервиса:

- сервис записи и детектор речи;
- сервис сжатия и обработки речевого сигнала;
- сервис по работе с распознаванием речи Google;

Детектор речи является необходимым дополнительным компонентом, обусловленным используемой архитектурой удаленного распознавания речи. В этом случае канал передачи данных (в нашем случае Интернет-соединение) является «бутылочным горлышком», ограничивающим максимально возможный объем передачи данных в заданный промежуток времени. При передаче по каналу слишком длинного отрезка речевого сигнала происходит недопустимо длительная задержка ответной реакции системы распознавания. Кроме того, при этом возникает значительный риск получения ошибочных результатов распознавания. Ввод коротких отрезков речи возможен при использовании ручного стартстопного режима, предоставляемого системой Google Voice. Однако для решения поставленной здесь задачи стенографирования устной речи этот режим оказывается крайне неудобным. Обычно диктовка осуществляется короткими фразами, заканчивающимися паузами. После произнесения каждой из них пользователь, как правило, желает убедиться в правильности распознавания и при необходимости повторить её более чётко.

Задача детектора речи заключается в автоматической локализации полезного сигнала, т. е. в определении начала и конца произнесённой фразы. Это обеспечивает автоматический пофразовый ввод речи. В экспериментальной программной системе использована библиотека SPTK для реализации механизма определения начала и конца фразы. Для этого производится запись небольшого отрезка сигнала в кольцевой буфер, далее этот отрезок проверяется на наличие частоты основного тона, и если таковая находится, то начинается запись сигнала в файл. Аналогичным образом определяется конец фразы.

Полученный полезный сигнал в модуле сжатия и обработки сигнала подготавливается для отправки на удаленный сервер распознавания речи Google Voice. Для этого сигнал кодируется в открытом формате FLAC (Free Lossless Audio Codec). При этом происходит сжатие сигнала кодеком, что уменьшает объем передаваемых данных и, как следствие, сокращает время ожидания результата распознавания. Использование open-source-кодека имеет и другие преимущества: простота использования сторонними разработчиками, единообразие получаемого сервером формата данных. Каждый компонент программы работает асинхронно и имеет свои пулы данных для нивелирования эффектов, связанных с задержками в работе каждого компонента. Такие задержки появляются как на этапе записи и кодирования речевого сигнала в файл, так и отправки его на удаленный сервер.

3. Экспериментальная прикладная программа

Для оценки предлагаемой модели был разработан программный комплекс, объединивший в себе ряд режимов (приложений) демонстрирующих различные аспекты использования речевых интерфейсов.

В качестве входных требований к проекту были взяты следующие положения:

- программное средство реализуется в виде приложения под ОС Windows, Linux;
- работа с программным средством должна вестись не только в ручном стартстопном режиме (как это предусмотрено системой Google Voice), но и в автоматическом режиме путём дополнительного включения в его состав детектора речи, определяющего начало и конец голосового сигнала;
- программное средство должно обладать простым и понятным графическим интерфейсом;
- программное средство должно характеризоваться минимальными требованиями по установке и настройке.

В связи с этим были выбраны следующие средства реализации:

- язык программирования C++ обеспечивает максимальную производительность при захвате и обработке речевого сигнала;
- библиотека построения пользовательских интерфейсов Qt Quick позволяет в кратчайшие сроки разрабатывать кроссплатформенные пользовательские интерфейсы;
- библиотека OpenAL для реализации однопипной работы с аудио устройствами под различными платформами.

В экспериментальной прикладной системе выделены следующие микросервисы:

- сервис записи и детектор речи;

- сервис работы с Google Voice;
- сервис работы с Google Translate;
- сервис работы с синтезаторами речи (Windows SpeechAPI, CMU Sphinx);
- сервис загрузки и обработки графического пользовательского интерфейса;
- сервис динамической загрузки логики работы режимов;
- и др.

В приложении присутствуют следующие режимы:

- телеграф
- сказка
- переводчик
- калькулятор

Согласно микросервисному подходу каждый логический элемент системы должен представлять собою независимую сущность, общающуюся с системой посредством асинхронных сообщений или событий, в реализуемом приложении это набор сервисов, перечисленный ранее. Режимы в приложении представляют собою компоненты на стороне клиента, использующие функциональность сервисов. Режимы реализованы в виде отдельных модулей (написанных на QML) и подгружаемых динамически после старта и инициализации основного приложения. Таким образом, перечень и логика работы режимов может быть изменена без необходимости повторной компиляции или сборки приложения. Это возможно благодаря динамической загрузке модулей режимов и интерпретации языка QML и JavaScript используемых для описания модулей.

При таком подходе приложение можно логически строго разделить на три основных слоя:

- слой поддержки микросервисной архитектуры (инфраструктура микросервисов);
- слой микросервисов;
- слой клиентских приложений (в рамках реализуемого приложения это режимы).

Ниже приведена общая схема приложения с реализацией микросервисной архитектуры.

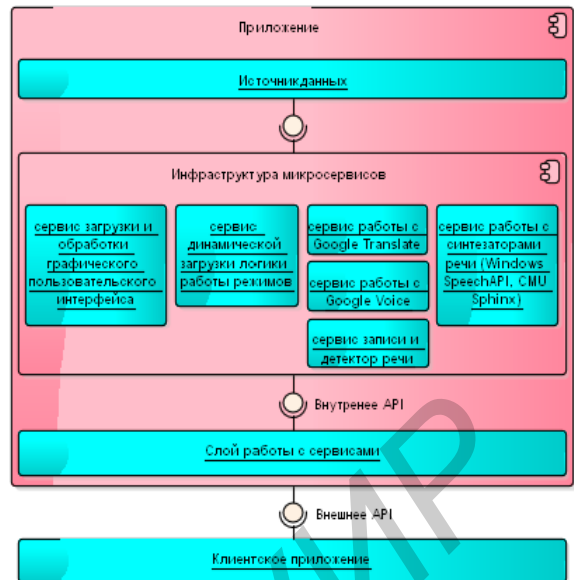


Рисунок 1 – Общая схема структуры приложения

Ниже представлен пользовательский интерфейс приложения:

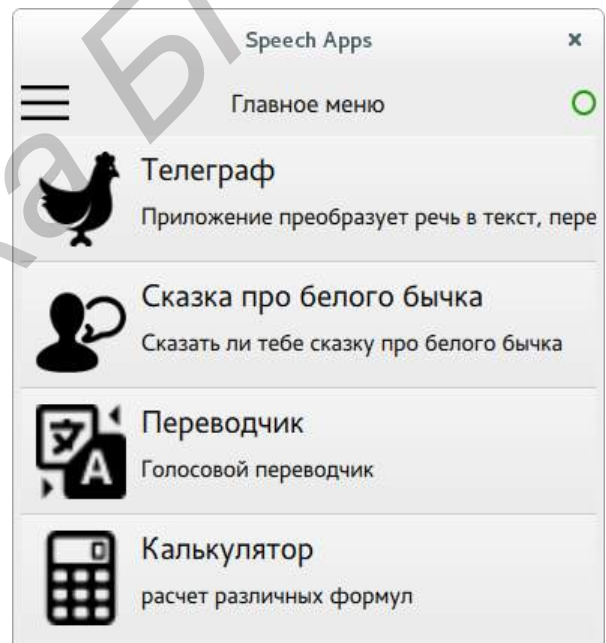


Рисунок 2 – Основное меню приложения

В приложении реализованы базовые настройки аудио входа и выхода, языка интерфейса и языка пользователя:

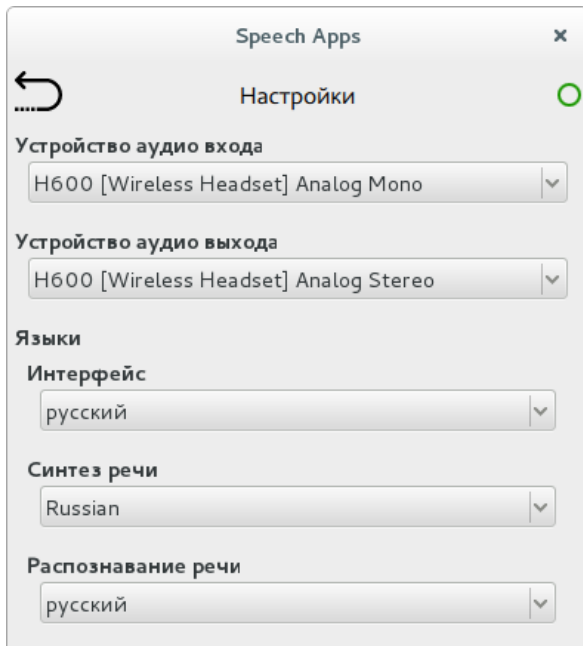


Рисунок 3 – Базовые настройки приложения

В режиме «Телеграф» приложение будет повторять любые сказанные пользователем фразы. При этом происходит их автоматическая запись, распознавание и отправка распознанного текста на вход синтезатора речи:

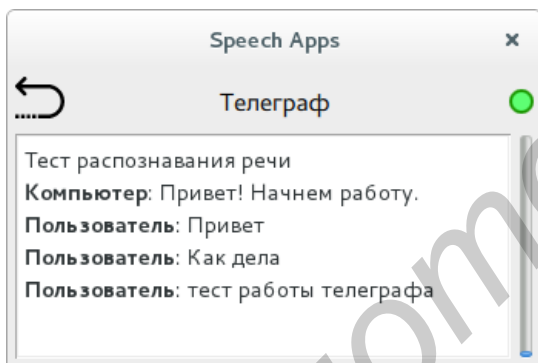


Рисунок 4 – Режим «Телеграф»

В режиме «Сказка» приложение демонстрирует элементарный диалог в стиле докучной сказки:

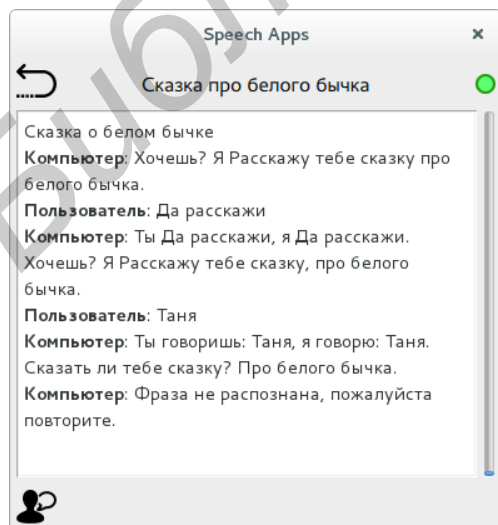


Рисунок 5 – Режим «Сказка»

В режиме «Переводчик» приложение отправляет распознанный текст на вход переводчика, а результат выводит в форме синтезированного текста:

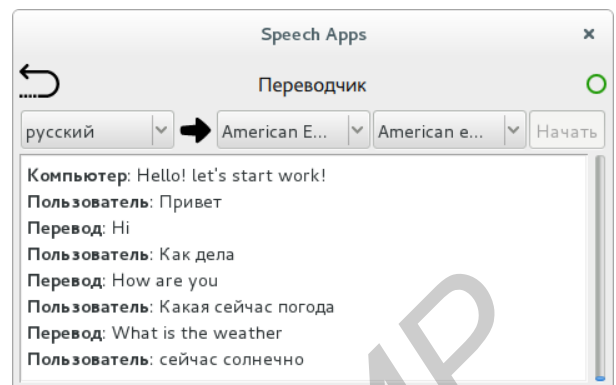


Рисунок 6 – Режим «Переводчик»

В режиме «Калькулятор» реализован ввод математических формул и дальнейший их расчёт с выводом результата в виде синтезированного текста. В приложении реализованы базовые математические операции:



Рисунок 7 – Режим «Калькулятор»

Заключение

Представленная модель речевого интерфейса с разбиением на отдельные компоненты со строго заданным функционалом, позволяет упростить разработку, а в дальнейшем – сопровождение, различных систем с речевым интерфейсом.

Компоненты системы распознавания и синтеза речи по тексту предоставляют конечному пользователю возможность устно задавать вопрос и слышать ответ на него от системы, а не просто вводить вопрос через клавиатуру и читать ответ с экрана компьютера. Это делает естественно-языковой интерфейс еще более привлекательным для пользователя.

Разбиение системы на отдельные независимые компоненты дает возможность интеграции сторонних разработок и проектов, производить интеграцию различных подходов и методов в рамках одного проекта, что позволяет эффективно использовать их лучшие стороны.

Библиографический список

[Barry, D. 2003] D. K. Barry, Web Services and Service-Oriented Architectures: The Savvy Manager's Guide: Morgan Kaufmann, 2003.

[Chris, R. 2014] Chris Richardson. Pattern: Microservices architecture. <http://microservices.io/patterns/microservices.html>, 2014

[Sam, N. 2015] Sam Newman. Building Microservices. O'Reilly Media, 2015

[Public Cloud Service Definition, 2010] Public Cloud Service Definition. Public Version 1.5 // VMware, Inc. [Electronic resource]. – 2010. Mode of access : <http://www.vmware.com/files/pdf/VMware-Public-Cloud-Service-Definition.pdf>.

[Mell, P. 2011] Mell, P. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology / P. Mell, T. Grance // U.S. Department of Commerce [Electronic resource]. – NIST Special Publication, 2011.

[DeMarco, 1979] Structured Analysis and System Specification, Tom DeMarco, Yourdon Press Computing Series, 1979

[PageJones, 1988] The Practical Guide to Structured Systems Design, 2d. ed., Meilir PageJones, Yourdon Press Computing Series, 1988

[Bertrand Meyer, 1988] Object Oriented Software Construction, Bertrand Meyer, Prentice Hall, 1988, p 23

[Barbara, L. 1988] Barbara Liskov, "Data Abstraction and Hierarchy," SIGPLAN Notices, 23,5 (May, 1988).

[Martin, Robert 2002] Martin, Robert (2002). Agile Software Development: Principles, Patterns and Practices. Pearson Education.

[Robert C. 1996] The Dependency Inversion Principle, Robert C. Martin, C++ Report, May 1996

[Manjoo F., 2011] Manjoo, F. Now You're Talking / Farhad Manjoo // The Slate Group. – Washington Post Company, 2012. – [Electronic resource] – Mode of access : http://www.slate.com/articles/technology/technology/2011/04/now_youre_talking.single.html. – Date of access : 01.08.2012.

[Singhal, A., 2011] Singhal, A. Knocking down barriers to knowledge / Amit Singhal // Google Official Blog [Electronic resource]. – 2011. – Mode of access : <http://googleblog.blogspot.com/2011/06/knocking-down-barriers-to-knowledge.html>. – Date of access : 01.08.2012.

[Enge, E. 2011] Enge, E. Search Algorithms with Google Director of Research Peter Norvig / E. Enge // Stone Temple Consulting [Electronic resource]. – 2011. – Mode of access : <http://www.stonetemple.com/search-algorithms-with-google-director-of-research-peter-norvig>. – Date of access : 01.08.2012.

MICROSERVICE DESIGN APPROACH FOR DEVELOPING SPEECH USER INTERFACES

B.M. Lobanov*, V.A. Zhitko**

**United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Republic of Belarus*

lobanov@newman.bas-net.by

***Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

zhitko.vladimir@gmail.com

In work describes design approaches to make system with speech user interface. Main flow is to use service oriented architecture and micro services business logic decompositions. Also describes components for text-to-speech and voice recognition. Primary goal of this work is making easy building applications with speech user interfaces.

Main Part

Microservices is modern approach to build scalable

flexible applications and computer services. This allow to build application as ecosystem with independent small services, called microservices. Microservice is application component with weak relations and performs only one business function.

There are three main layout in such applications. They are: microservice infrastructure, list of microservices and client applications. All this layout communicate with each other via unified protocol. It could be REST api, json api, xml api or something else.

Describe experimental system with a number of examples of user speech interfaces. It include text-to-speech, speech-to-text transformations.

Additional third-party component is natural Russian voice input. For this component used Google voice recognition service with made some changes. So component can recognize non-stop user speech. Also used as third-party component Google translate service.

Another important third-party component is voice output. This component is top required for making natural language interface friendlier for users. Also, as previous component, it built in a traditional technology. But some tasks are hard to solve in traditional ways, for example correct detect the stress in words, sometimes for this need understand meaning of text content, and this could be solve by semantic approaches.

Conclusion

In the given paper short description of model and methods for designing and prototyping natural language interfaces with voice input and output.



УДК 004.822:514

КОНТЕКСТНЫЕ ПРАВИЛА ДЛЯ РАЗРЕШЕНИЯ МОРФОЛОГИЧЕСКОЙ МНОГОЗНАЧНОСТИ В КОРПУСЕ ТАТАРСКОГО ЯЗЫКА

Гильмуллин Р.А., Гатауллин Р.Р.,

*Казанский Федеральный университет,
Институт «Прикладной семиотики» Академии Наук Республики Татарстан,
г. Казань, Российская Федерация*

rinatgilmullin@gmail.com

ramil.gata@gmail.com

Данная работа является продолжением работ по подготовке размеченного корпуса татарского языка. Ранее был представлен проект (<http://tatcorp.antat.ru>) по ручному снятию морфологической многозначности в корпусе татарского языка, основной целью которого является использование краудсорсингового подхода для накопления данных. Следующим шагом стала разработка инструментария для создания, тестирования и апробации контекстных правил разрешения морфологической многозначности.

Ключевые слова: корпус татарского языка, разрешение морфологической многозначности, контекстный метод.

Введение

Татарский язык относится к агглютинативному типу языков, т.е. словоформы в татарском языке образуются последовательным присоединением аффиксов к основе слова. Аффиксы имеют жесткий порядок следования, но некоторые последовательности могут повторяться, усложняя смысл словоформы и образуя таким образом теоретически бесконечно длинные словоформы (например, «урманнардагылардагылар» - те, что (кто) у тех, кто в лесах) [Сулейманов и др., 1997]. Но на практике, по статистическим данным корпуса, длина аффиксальной цепочки в среднем не превышает 5-6 аффиксов, а максимальная длина аффиксальной цепочки равна 12. Но и такая ситуация приводит к большому разнообразию типов морфологической многозначности.

Из предварительного анализа корпуса татарского языка выявлено, что почти треть всех словоформ (~31%) корпуса в той или иной мере имеют более одного разбора [Хакимов и др., 2014]. Количество типов морфологической многозначности превышает 10000 для корпусной выборки в 21 млн. словоупотреблений [Гатауллин, 2014], что вместе с агглютинативностью татарского языка теоретически приводит к бесконечному многообразию таких типов. Практически же должна быть возможность свести их к конечным классам, для которых правила разрешения будут одними и те же. На данный момент было выявлено порядка 400 таких классов,

но данная гипотеза еще требует подтверждения и дальнейших исследований.

Для автоматического разрешения морфологической многозначности предлагается применять гибридный метод, включающий метод, основанный на контекстных правилах, и статистико-вероятностный метод. Такой выбор обусловливается тем, что, несмотря на то, что метод контекстных правил показывает достаточно хорошую точность при разрешении, сама по себе разработка таких правил достаточно трудоемкая задача, требующая тщательного лингвистического анализа контекстных ограничений для каждого типа многозначности. Статистико-вероятностные методы и методы машинного обучения хорошо выявляют и успешно используют при работе скрытые закономерности и взаимосвязи в контексте, но минусом их является плохая обучаемость на разреженных данных. Для некоторых типов, действительно, в корпусе имеется мало примеров.

Таким образом, предлагается перед использованием статистико-вероятностного метода, как первоначальный этап, применять контекстный метод. Также кроме случаев с разреженными данными, контекстный метод подходит для случаев, когда многозначность возникает вследствие избыточности словаря основ морфоанализатора, и других исключительных случаев, когда легче прописать контекстные ограничения, чем готовить обучающую выборку по данному типу или случаю [Хакимов и др., 2014].

В настоящее время подготовлено веб-приложение для разметки корпуса татарского языка. Основной упор делается на «кроудсорсинговый» аспект, т.е. использование усилий большого количества людей. С помощью данного инструмента, во-первых, будет подготовлен вручную размеченный подкорпус. Во-вторых, будет получено достаточное количество данных для обучения статистико-вероятностного метода. Следующим шагом в развитии этого проекта стал инструментарий разработки контекстных правил для разрешения, использующий ранее полученные данные для тестирования и апробации разрабатываемых правил.

1. Ключевые понятия и архитектура инструментария для разработки контекстных правил разрешения морфологической многозначности

В ранних публикациях [Гатауллин и др., 2014] описывались идея и архитектура инструментария, представлен прототип, который показал работоспособность. В текущей работе инструментарий был доработан и реализован в виде веб-приложения. В работе [Зинькина и др., 2005] подробно описаны основные достоинства и недостатки метода контекстных правил, приведены конкретные структуры обобщенных правил для разрешения функциональной омонимии некоторых типов.

Обобщенный метод контекстного разрешения функциональной омонимии для татарского языка включает несколько этапов [Гатауллин и др., 2014]:

1. построение полной классификации типов функциональных омонимов;
2. выделение минимального множества разрешающих контекстов для каждого типа. Минимальность множества означает, что для каждого типа функционального омонима следует оценить сложность распознавания каждой части речи, принадлежащей данному типу. Затем необходимо построить множество разрешающих контекстов (МРК), имеющих минимальную сложность распознавания. В алгоритмической записи данное требование выражается следующим правилом: если для функционального омонима X, имеющего тип T1 или T2, подошло правило из МРК, то тип омонима X определяется примененным правилом, иначе приписывается альтернативный тип;
3. построение управляющей структуры обобщенного правила, обеспечивающего максимальную точность распознавания.

Для решения поставленных задач разработана соответствующая архитектура программного инструментария, включающая следующие базовые объекты и понятия:

- Омоформа (или функциональный омоним) – слова, совпадающие в своем звучании лишь в

отдельных формах (той же части речи или разных частях речи);

- База типов омоформ (или База контекстных правил) – иерархически упорядоченный список типов омоформ; для каждого типа определено множество разрешающих контекстов. На основе этих правил происходит разрешение многозначности для отдельно взятого типа омоформ;

- Обобщенное правило разрешения (ОПР) – правило, на основе контекстной информации определяющее актуальный вариант структуры омоформы. Для каждого типа функционального омонима следует оценить сложность распознавания каждой части речи, принадлежащей данному типу;

- Множество разрешающих контекстов (МРК) – совокупность минимальных разрешающих контекстов, достаточных для распознавания функционального омонима как определенного варианта структуры омоформы;

- Управляющая структура обобщенного правила обеспечивает и контролирует порядок применения правил;

- Минимальный разрешающий контекст – неделимое в данном контексте простое условие, имеющее минимальную сложность распознавания.

Процесс распознавания омонимии происходит следующим образом:

1. У анализируемого слова определяется тип функциональной омонимии, и в соответствии с этим типом из Базы контекстных правил находится обобщенное правило разрешения;
2. Управляющая структура задает порядок применения правил;
3. При применении каждого правила, проверяется каждый минимальный контекст разрешения этого правила;
4. Если при проверке правила получили подтверждение о его истинности, то функциональная омонимия распознается в соответствии с этим вариантом структуры омоформы;
5. Иначе, если есть другое правило, осуществляется переход к следующему правилу и выполняется то же самое;
6. И если нет другого правила, то в качестве структуры выбирается тип по умолчанию;
7. Если нет такого типа, то многозначность помечается как неразрешенная.

2. Кроудсорсинговый аспект приложения

В некоторых источниках «краудсорсинг» (англ. crowd – толпа, народ; source - ресурс) трактуется как мобилизация ресурсов людей посредством информационных технологий с целью решения задач, стоящих перед бизнесом, государством и обществом в целом [5]. Действительно, для решения некоторых задач такой подход полностью оправдывает себя. Примером служат всемирно-известный ресурс Википедия, программистский

ресурс <http://stackoverflow.com/> и другие разного рода форумы, где сбором информации и наполнением сайта занимаются обычные пользователи. В сфере NLP можно отметить Открытый корпус русского языка <http://opencorpora.org/>, где с помощью пользователей происходит разметка корпуса [Бочаров и др., 2013]. Как уже отмечалось ранее [Гатауллин, 2014], у нас также имеется опыт в таком роде проекте: с помощью пользователей ресурса <http://tatcorp.antat.ru> разрешается морфологическую многозначность в корпусе татарского языка.

Основная идея подхода состоит в разбиении задачи на мелкие подзадачи, которые достаточно легко решаются и не сильно затрудняют пользователя. Другой важной частью является мотивация пользователей. Для одних это всевозможные “ачивки” (англ. achievement - достижение), для других развитие open-source проектов (англ. open - открытый; source - ресурс).

Применения данного подхода для разработки контекстных правил вполне возможно, но в отличие от случая ручного снятия морфологической многозначности, где требуется простое знание языка, разработка контекстных правил требует определенных знаний в области языкознания и лингвистики, что сильно ограничивает круг возможных пользователей. Но несмотря на это, есть возможность привлечения для работы студентов-лингвистов и учителей, которые занимаются данной проблематикой.

3. Разработка контекстных правил

Для того, чтобы начать разработку контекстных правил необходимо зарегистрироваться на сайте <http://tatcorp.antat.ru> и перейти на вкладку <http://tatcorp.antat.ru/disam/rules/>, где представлен список всех оформов, для которых уже имеются правила разрешения. Имеется возможность как улучшать уже имеющиеся правила, так и создавать новые правила для не имеющих в списке типов оформов (см. Рис.1).

Так как в разработке участвует не один пользователь, появляется необходимость разграничения доступа пользователей, а также необходимость своего рода “песочницы”, где происходит разработка и тестирование правил. После этого правило может быть добавлено в основную базу правил.

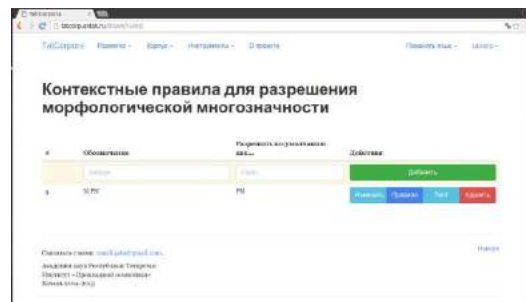


Рисунок 1 – Основная страница веб-приложения по разработке контекстных правил.

Исходя из этих соображений, были приняты простые правила:

- нельзя удалять и редактировать правила других пользователей. При желании доработать правило, нужное правило дублируется с припиской к текущему пользователю и все улучшения делаются там;
- пока правило не добавлено в основную базу правил, она находится в зоне “песочницы”, где она может редактироваться и тестироваться;
- перед добавлением (либо обновлением) в основную базу, правило тестируется на корпусных данных, и при условии успешного прохождения тестов, правило добавляется в основную базу и может быть применено в процессе разрешения многозначности.

Как уже было описано ранее, разработка правил состоит из нескольких этапов:

- сначала выбирается тип оформов, для разрешения которых разрабатываются правила;
- Потом выбирается управляющая структура, т.е. в процессе лингвистического анализа выявляются множества минимальные разрешающие контексты и упорядочиваются по частотности так, что самые частые случаи будут рассматриваться первыми;
- для каждого множества минимальных контекстов определяется порядок минимальных контекстов (см. Рис.2).

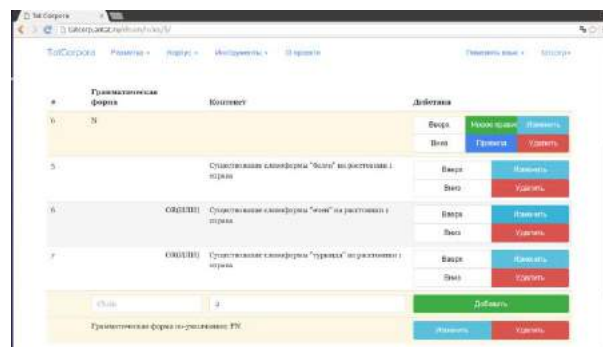


Рисунок 2 – Страница управляющей структуры правила с определенным множеством минимальных контекстов.

Кроме самой разработки имеется возможность тестирования правил на корпусных данных (см. Рис. №3). Очевидно, таким образом, легко выявляются исключительные и ошибочные случаи

распознавания, что намного облегчает процесс разработки.

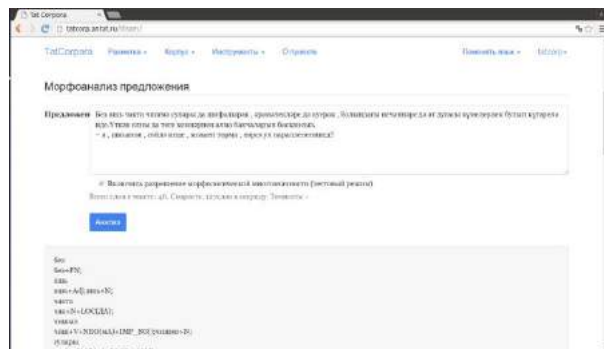


Рисунок 3 – Страница тестирования контекстных правил.

Заключение

В данной работе представлен инструментарий разработки контекстных правил для разрешения морфологической многозначности в корпусе татарского языка. При разработке особый упор делается на “краудсорсинговый” аспект приложения. На данный момент приложение на этапе тестирования. В разработке участвуют 4 человека, было разработано 9 тестовых правил, которые по большей части покрывают исключительные случаи морфологической многозначности.

Следующим шагом планируется реализация статистико-вероятностных методов для разрешения и компоновка их с методом контекстных правил.

Библиографический список

[Сулейманов и др., 1997] Сулейманов, Д.Ш. Двухуровневое описание морфологии татарского языка / Д.Ш. Сулейманов, Р.А. Гильмуллин // Тезисы Международной научной конференции "Языковая семантика и образ мира". Казань: Изд-во Казан. гос. ун-та., 1997. Книга 2. С. 65-67.

[Хакимов и др., 2014] Разрешение грамматической многозначности в корпусе татарского языка / Б.Э.Хакимов, Р.А.Гильмуллин, Р.Р.Гатауллин // Учен. зап. Казан. ун-та. Сер. Гуманит. науки. - 2014. - Т. 156, кн. 5. - С. 236-244.

[Гатауллин, 2014] Веб-инструментарий для снятия морфологической многозначности в текстовом корпусе татарского языка / Р. Р. Гатауллин // Сохранение и развитие родных языков в условиях многонационального государства: проблемы и перспективы: материалы V Международной научно-практической конференции (Казань, 19-22 ноября 2014 г.). – Казань: Отечество, 2014. - С. 71-73

[Гатауллин и др., 2014] Программный инструментарий для разрешения морфологической многозначности в татарском языке / Р. Р. Гатауллин, Д. Ш. Сулейманов, Р. А. Гильмуллин // Открытые семантические технологии проектирования интеллектуальных систем OSTIS-2014 Open Semantic Technologies for Intelligent Systems МАТЕРИАЛЫ IV МЕЖДУНАРОДНОЙ НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ (Минск, 20-22 февраля 2014 года), - Минск. : БГУИР, 2014. - С. 503-508.

[Зинькина и др., 2005] Ю.В. Зинькина, Н.В. Пяткин, О.А. Невзорова, Разрешение функциональной омонимии в русском языке на основе контекстных правил. // Труды междунар. конф. Диалог'2005.– М.: Наука, 2005. С. 198-202.

[Бочаров и др., 2013] Crowdsourcing morphological annotation / Bocharov V.V., Alexeeva S.V., Granovsky D.V., Protopopova E.V., Stepanova M.E., Surikov A.V. // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной

Международной конференции «Диалог» (Бекасово, 29 мая–2 июня 2013 г.). Вып. 12 (19). — М.: РГГУ, 2013.

CONTEXTUAL RULE METHOD FOR MORPHOLOGICAL DISAMBIGUATION IN THE TATAR LANGUAGE

Gilmullin R.R., Gataullin R.R.

Research Institute of Applied Semiotics of the Tatarstan Academy of Sciences, Kazan Federal University, Kazan, Russia

rinatgilmullin@gmail.com

ramil.gata@gmail.com

The article describes a software tool for creating, editing, and testing contextual rules for the automatically resolve of morphological ambiguity in the Tatar language.

Introduction

For last years, scientists from Research Institute of Applied Semiotics of the Tatarstan Academy of Sciences have been developing Tatar language corpus, which contains by these days more than 40 million word usages. Morphological features are automatically annotated, but the problem of morphological ambiguity has not been solved yet. Since Tatar language is one of agglutinative languages, types of morphological ambiguities are theoretically infinite. It means that machine learning algorithms will not cover all cases of them, due to data sparseness. So it is necessary to combine them with rule base methods for these cases. And this work introduces such tool for Tatar language.

Main Part

Actually, the rule development tool was constructed as web service. It is available at <http://tatcorp.antat.ru>. To get more efficiency, “crowdsourcing” approach is used. It means, that rules are created with help of many users of systems. Of course, as production rules will be used only successfully tested rules.

Conclusion

For now the tool is in testing phase. By January 2016, nine testing rules were created and tested on corpus data. Next step will be the development of machine learning algorithms, which will be combined with this tool.



УДК 004.822:514

СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР КАЗАХСКОГО ЯЗЫКА НА ОСНОВЕ ГРАММАТИКИ СВЯЗЕЙ - LINKGRAMMARPARSER

Бегимтай У.Х.

*Евразийский Национальный Университет имени Л.Н.Гумилева,
г. Астана, Республика Казахстан*

ulugbek_begimtai@mail.ru

В работе описан принцип работы синтаксического анализатора на основе грамматики связей – linkgrammarparser для казахского языка. Парсер грамматики связей позволяет размечать слова тегами частей речи и определять тип связей между ними. Также в данной работе показан принцип проверки сходства двух предложений и принцип вычисления числа на семантическом графе соответствующих и несоответствующих дуг.

Ключевые слова: linkgrammarparser, синтаксический анализатор, парсер грамматики связей, семантический анализ казахского языка.

Введение

Информатизация общества приводит к изменению ситуации на трудовом рынке, требуя все новые квалификации и компетентности. Это заставляет большинство людей в корне менять свою специализацию или обучаться новым знаниям. Меняются традиционные формы обучения и виды услуг в области образования. В мире наступает эра электронного образования, в котором количество студентов стало более 150 млн. человек. Что касается Казахстана, то сектор электронного образования почти отсутствует.

Если сравнивать традиционное образование с электронным образованием, то доказано, что в последнем скорость образования на 30-60% быстрее, но качество низкое. Причиной низости качества электронного образования является пассивность и статичность электронных образовательных ресурсов (как правило, простой текстовый или графический материал) и отсутствие диалога с обучающимися в реальном масштабе времени. Ведь при изучении этих ресурсов у обучающегося может появиться масса вопросов к ним, а у них просто нет возможности ответить на эти вопросы, которые с продолжением изучения могут только возрастать. Кроме того, современные методы контроля и оценки знаний не всегда выдают объективную оценку знаний. Эти проблемы можно решить, если статические электронные образовательные ресурсы заменить

интеллектуальными анализаторами текстов для проверки качества знания.

В этой статье написана часть технологий проверки знания путем интеллектуального сравнения двух предложений - "предложение запрос" и "предложение претендент". Данная технология позволит автоматизировать процесс проверки знания обучающегося путем интеллектуального анализа текста и интеллектуальной оценки знания на основе грамматики связей - linkgrammarparser.

Общий принцип работы

1.1. Сравнение семантических графов двух предложений

Предположим, даны два предложения \bar{x}_1 и \bar{x}_2 , и второе предложение необходимо сравнить с первым. Будем называть предложение \bar{x}_1 запросом, а предложение \bar{x}_2 – претендентом. Предположим есть семантический граф для \bar{x}_1

$$G_1 = \langle V_1, E_1, s_1, r_1 \rangle \quad (1)$$

и соответствующий семантический граф для \bar{x}_2

$$G_2 = \langle V_2, E_2, s_2, r_2 \rangle \quad (2)$$

Каждой дуге семантического графа запроса сопоставляется «равная» ей дуга в семантическом графе претендента.

- V – множество вершин графов

- E – множество дуг графов(таблица 1)
- $s:V \rightarrow L$, L – подмножество графов
- $r:E \rightarrow M$, M – множество семантико-синтаксических отношений.

После этого задается отображение двух графов:

$$F : G_1 \rightarrow G_2 \quad (3)$$

Чтобы определить сходство двух предложений \bar{x}_1 и \bar{x}_2 нужно вычислить число на семантическом графе как соответствующих, так и несоответствующих дуг. Вес каждой дуги соответствует семантико-синтаксическому отношению.

Таблица 1 – Пометки дуг семантического графа

Обозн-е	Семантико-синтак-ое отношение	Пример	Вес
<i>Пример: Кеше(вчера) өте(очень) ашулы(злая) ит(собака) адамға(на человека) қатты(сильно) үрді(лаяла) де(и) қолындағы затты(вещь которую держал в руке) ұрлап кетті(своровала)</i>			
SUB (PRED_A CT_SUB)	Соответствует связи между действием и действующим субъектом	$\begin{array}{c} \text{Үрді} \\ \xrightarrow{\text{SUB}} \\ \text{ит} \\ \text{ұрлады} \\ \xrightarrow{\text{SUB}} \\ \text{ит} \end{array}$	1
dirOBJ (PRED_A CT_DIR_ OBJ)	Соответствует связи между действием и «прямым» объектом действия	$\begin{array}{c} \text{ұрлады} \\ \xrightarrow{\text{dirOBJ}} \\ \text{затты} \end{array}$	0,2
indirOBJ	Соответствует связи между действием и «косвенным» объектом действия	$\begin{array}{c} \text{Үрді} \\ \xrightarrow{\text{indirOBJ}} \\ \text{адамға} \end{array}$	0,2
ATTR (N_ATTR)	Соответствует связи между объектом и его признаком объект ның нышаны арасындағы байланыс	$\begin{array}{c} \text{ит} \\ \xrightarrow{\text{ATTR}} \\ \text{ашулы} \end{array}$	0,1
mannCIR (ADV_M AN_ADV)	Соответствует связи между действием и признаком образа действия или признаком и его степенью	$\begin{array}{c} \text{Үрді} \\ \xrightarrow{\text{mannCIR}} \\ \text{қатты} \\ \text{ашулы} \\ \xrightarrow{\text{mannCIR}} \\ \text{өте} \end{array}$	0,1
timeCIR	Соответствует связи между действием и временной характеристикой действия	$\begin{array}{c} \text{Үрді} \\ \xrightarrow{\text{timeCIR}} \\ \text{кеше} \\ \text{ұрлады} \\ \xrightarrow{\text{timeCIR}} \\ \text{кеше} \end{array}$	0,1

1.2. Принцип вычисления совпадения двух предложений

Ниже представлена формула вычисления степени совпадения предложений, подходящая для ранжирования претендентов и отвечающая вышеизложенным принципам:

$$y = \frac{\sum_{i=1}^N p_i - \left(\frac{\sum_{i=1}^M q_i}{\tilde{M}} \right)}{\sum_{i=1}^K r_i} \quad (4)$$

где:

У – коэффициент совпадения претендента предложения с запросом;

K, N – число дуг в семантическом графе запроса и подграфе претендента, состоящем из совпадающих дуг;

\tilde{M} – общее число дуг в семантическом графе претендента;

M – число несовпадающих дуг в семантическом графе претендента,

$$M = \tilde{M} - \sum_{i=1}^N N_i \quad (5)$$

r_i, t_i – веса дуг семантических графов запроса и претендента соответственно;

p_i – вес совпадающей дуги в семантическом графе претендента;

q_i – вес несовпадающей дуги в семантическом графе претендента.

Таким образом, чем больше в семантическом графе претендента имеется совпадающих дуг, и чем больше их веса, тем большую оценку он должен получить. Кроме того, в формуле присутствует корректирующее слагаемое.

$$-\frac{\left(\sum_{i=1}^M q_i \right)}{\left(\sum_{i=1}^{\tilde{M}} t_i \right)} \quad (6)$$

Оно по абсолютной величине не превышает единицу и служит для ранжирования претендентов, имеющих одинаковое число равнозначных совпадений с запросом. При этом чем легче несовпадающие дуги, тем меньше снижается оценка претендента.

Пример работы парсера

1.3. Обработка предложения претендента

Для примера возьмем предложение на казахском языке:

X₁ - "Қызыл(красная) түлкі(лиса) ақ(белого)

қоянды(зайца) өте(очень) тез(быстро) жеді(съела)".

После обработки получаем следующий синтаксический вид:

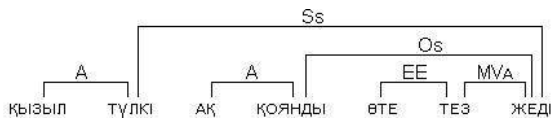


Рисунок 1 - Связь коннекторов

V - {қызыл, түлкі, ақ, қоянды, өте, тез, жеді}

E - {A, Ss, MVA, Os, A, EE}

Каждое слова приводим в "нормализованный вид" и указываем к какой части речи принадлежит каждое из них. Вершины этих графов являются "базовыми метасловами".

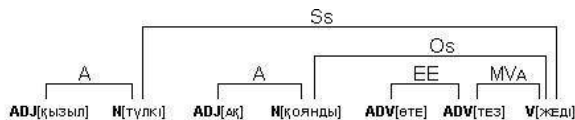


Рисунок 2 - Определение базовых метаслов

Затем происходит построение производных метаслов и формируется конечный набор метаслов.

В нашем случае "базовое метаслово" V[жеді](был съеден) превращается в "производное метаслово" O_SH_3[же](есть), а "базовое метаслово" N[қоянды](зайца) превращается в "производное метаслово" TBS_3[қоян](заяц). Таким образом, граф, состоящий из конечного набора метаслов, будет иметь следующий вид:

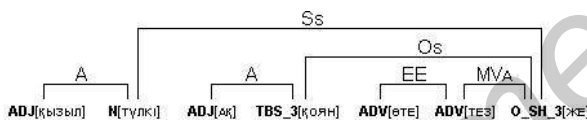


Рисунок 3 - Определение производных метаслов

После выбора главной связи и главного метаслова предложения, которые в данном случае равны Ss O_SH_3[же] соответственно, получится дерево метаслов:

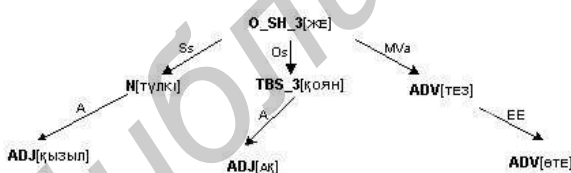


Рисунок 4 - Дерево метаслов

Теперь каждый лист дерева будет втягиваться в своего родителя для построения метасвязей.

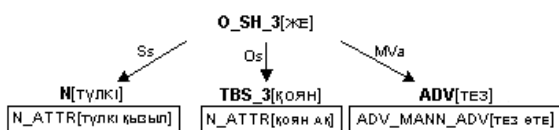


Рисунок 5 - Построение метасвязей

Таким образом, получается конечный набор метасвязей:

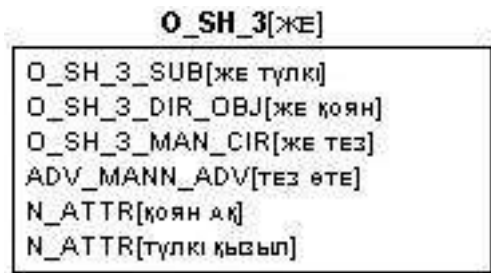


Рисунок 6 - Конечный набор метасвязей

На основе этого набора будет построен следующий семантический граф:

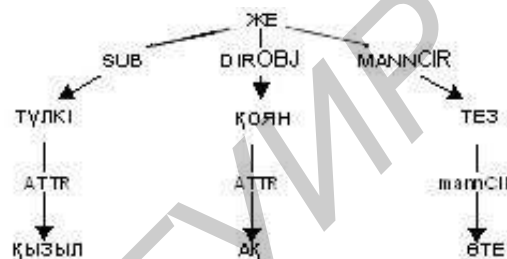


Рисунок 7 - Семантический граф

1.4. Обработка предложения запроса

Пусть теперь имеется второе предложение: X₂ - "Ақ(белый) қоянды(заяц) жеген(съеденный) түлкі(лисой)". Синтаксическая структура, порожденная системой LinkGrammarParser, будет иметь следующий вид:

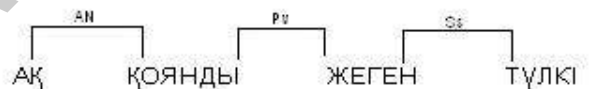


Рисунок 8 - Связь коннекторов

Этот граф обрабатывается аналогично первому. После построения финального набора метаслов и выбора корня получится следующее дерево метаслов:

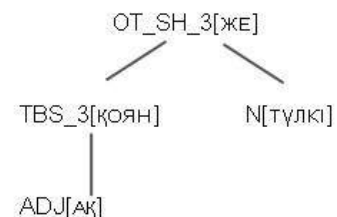


Рисунок 9 - Дерево метаслов

После стягивания графа получится набор метасвязей, указанный ниже:

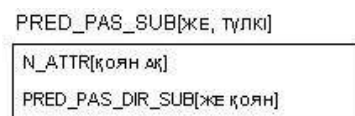


Рисунок 10 - Набор метасвязей

По данным метасвязям будет построен следующий семантический граф:

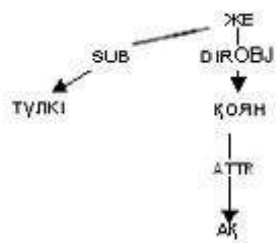


Рисунок 11 - Семантический граф

Сопоставление двух графов и оценка степени совпадения предложений

Далее производится сопоставление семантических графов. Предположим, что первое предложение – это запрос, а второе – претендент.

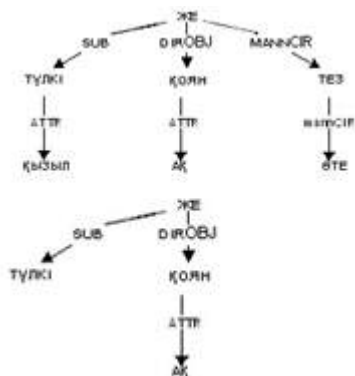


Рисунок 12 - Сопоставление двух графов

Таким образом, получается, что в подграфе второго графа, состоящем из «совпадающих» дуг, имеется одна компонента связности, состоящая из трех дуг. Дуг, не помеченных как «совпадающие», во втором графе нет. Поэтому получается следующая оценка степени совпадения второго предложения с первым

$$y = \frac{3 \cdot (P_{SUB} + P_{ATTR} + P_{DIR OBJ})}{6 \cdot (P_{SUB} + P_{DIR OBJ} + 2 \cdot (P_{ATTR} + P_{MAN+CR}))} = \frac{3 \cdot (100 + 5 + 25)}{6 \cdot (100 + 25 + 2 \cdot (5 + 5))} = 0,45$$

Заключение

Предложенный же метод основан на предположении, что на вход ему подается правильная диаграмма, в которой все связи расставлены так, как их расставил бы человек, поэтому если на вход будет подана некорректная диаграмма, то и семантическое дерево будет отображать связи между понятиями неверно, то есть так, как они отражены в разборе.

На данный момент проработано относительно небольшое количество синтаксических конструкций, поэтому только это небольшое число конструкций может анализироваться. Дальнейшее расширение словарей повлечет расширение множества анализируемых конструкций.

Библиографический список

[Бениаминов, 2008] Бениаминов, Е.М. О построении Web-сервера в стиле SemanticWiki с открытым контекстным языком представления и запросов/Е. М. Бениаминов// КИИ-2008. Труды конференции. Т 2, С. 15-21

[Temperley D., Sleator D., Lafferty J.] Temperley D., Sleator D., Lafferty J. Link Grammar Documentation. – 1998./Electronic resource//: <http://www.link.cs.cmu.edu/link/dict/index.html>

[Murzin F., Perfliev A., Shmanina T.] Murzin F., Perfliev A., Shmanina T. Methods of syntactic analysis and comparison of constructions of a natural language oriented to use in search systems/Murzin F., Perfliev A., Shmanina T.//Bull. Nov. Comp. Center, Comp. Science. – 2010. – Iss. 31. – pp. 91-109.

[Батура Т.В., Мурзин Ф.А.] Батура Т.В., Мурзин Ф.А. Машинно-ориентированные логические методы отображения семантики текста на естественном языке: моногр. Институт систем информатики им. А.П. Ершова СО РАН/Батура Т.В., Мурзин Ф.А.// – Новосибирск: Изд. НГТУ, 2008. – 248 с.

SYNTACTICAL ANALYZER OF KAZAKH LANGUAGE BASED BY "LINK GRAMMAR PARSER"

Begimtay U.H.

*Eurasian National University named L.N.Gumilev,
Astana, Kazakhstan*

ulugbek_begimtai@mail.ru

In work the main concepts of semantic model for parsing sentences in kazakh language by link grammar parser. And in main part this work show one example of parsing kazakh proposal.

Introduction

In this part describes the general condition of the intellectualization of the world and particularly in Kazakhstan.

Main Part

In main part it covers the general principle of operation link grammar parser for kazakh language. Step by step visually demonstrate how parses sentences

Conclusion

In conclusion shows the result of processing offers a specific example.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК [004.522+004.934+004.91]:004.89

АЛГАРЫТМ І ЛІНГВІСТЫЧНЫЯ РЭСУРСЫ ДЛЯ НАРМАЛІЗАЦЫІ ТЭКСТАЎ ГЕАГРАФІЧНАГА ДАМЕНА

Гецэвіч Ю.С., Качан Я.С., Лысы С.І., Маракулiна П.А., Крывальцэвіч А.В.

Аб'яднаны інстытут праблем інфарматыкі НАН Беларусі, Мiнск, Рэспубліка Беларусь

yury.hetsevich@gmail.com

evgeniakacan@gmail.com

stanislau.lysy@gmail.com

marakulina.polina@gmail.com

elena.krivaltsevich@gmail.com

У дадзеным артыкуле апісваецца алгарытм лінгвістычнай апрацоўкі і нармалізацыі тэкстаў геаграфічнага дамена на прыкладзе вучэбнага дапаможніка “Геаграфія Беларусі”. Прыведзена паслядоўнасць крокаў вылучэння ўсіх катэгорый сімвалаў, лікаў і іншых ужыванняў, неабходных для апрацоўкі.

Ключавыя словы: лінгвістычная апрацоўка, нармалізацыя, адзінка вымярэння, семантычная катэгорыя.

Уводзіны

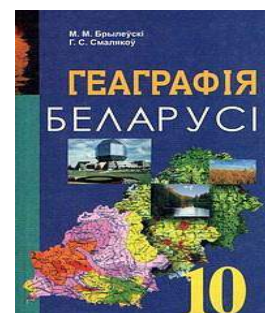
Адной з задач сістэм сінтэзу і распазнавання маўлення з’яўляецца агучванне апрацаванага тэксту. На сённяшні дзень падобныя сістэмы дастаткова якасна ажыццяўляюць усе пастаўленыя перад імі мэты. Аднак, нягледзячы на яскравыя поспехі, у працэсе сінтэзу маўлення застаецца яшчэ адна нявырашаная праблема: успрыманне і агучванне незразумелых для машыны сімвалаў (лікі, скарачэнні, даты і г.д.) [Гецэвіч, 2014]. Ужо існуюць артыкулы паводле апісанай праблемы [Барадзіна, 2015]. Таму аўтары дадзенага артыкула, азнаёміўшыся з даследаваннямі папярэдніх навукоўцаў, лічаць важным распрацаваць правяраныя нармалізаваныя мноствы фраз і сказаў на прыкладзе геаграфічнага дамена для далейшага тэставання сістэм сінтэзу і распазнавання маўлення. У будучыні атрыманы спіс будзе пакладзены ў аснову аўтаматызаванага працэсу нармалізацыі ўсіх невядомых сімвалаў.

1. Збор матэрыялаў і метадыка іх апрацоўкі

У якасці матэрыялаў даследавання выкарыстоўваліся вучэбныя дапаможнікі па геаграфіі з пятага па адзінаццаты клас. За аснову быў выбраны вучэбны дапаможнік “Геаграфія Беларусі” для дзясятага класа ўстаноў агульнай сярэдняй адукацыі з беларускай мовай навучання (малюнак 1). Дапаможнік складаецца з дзесяці

частак: геаграфічнае становішча і даследаванні Беларусі; прыродныя ўмовы і рэсурсы Беларусі; геаграфічныя ландшафты, экалагічныя праблемы; прыроднае раяніраванне Беларусі; насельніцтва; агульная характарыстыка гаспадаркі; геаграфія вытворчай сферы; геаграфія невытворчай сферы; Беларусь у сусветнай супольнасці; вобласці Беларусі. Такім чынам, дадзеная кніга дае магчымасць сабраць матэрыял не толькі геаграфічнага дамена, але і скарачэнні і сімвалы іншых даменаў.

Увесь матэрыял перагледжаны і былі абраны па тры прыклады на кожную семантычную падгрупу.



Малюнак 1 – Вучэбны дапаможнік “Геаграфія Беларусі” для дзясятага класа ўстаноў агульнай сярэдняй адукацыі з беларускай мовай навучання

2. Класіфікацыя сабраных матэрыялаў

Сабраны матэрыял быў аформлены як спіс сказаў з ненармалізаванымі дадзенымі з наступнымі семантычнымі катэгорыямі:

- 1) Адзінкі вымярэння:
 - а) Маса (т, кг, г, мг).
 - б) Тэмпература (градусаў па Цэльсію).
 - в) Плошча (га, км², м², см²).
 - г) Аб'ём (м³, см³).
 - д) Вага (кг, г, мг).
 - е) Час (г,ст,г, хв).
 - ж) Даўжыня і адлегласць (км, м, см).
 - з) Магутнасць (Вт, кВт, мВт).
 - і) Ападкі (мм, см, м, м,км/с).
 - к) Геаграфічныя каардынаты (пд/пн ш, у/з д, градусаў °, хвілін ').
 - л) Шчыльнасць насельніцтва (чал/км², чал/м², тыс.чал/км²,млн. чал., тыс. чал.).
- 2) Працэнты, праміле.
- 3) Скарачэнні (інш., г.д., г., ст., стст.).
- 4) Лікі.
- 5) Матэматычныя знакі (-,+/,=).
- 6) Абрэвіятуры.
- 7) Ініцыялы.
- 8) Даты.
- 9) Выпадковыя дадзеныя.
- 10) Статыстычныя дадзеныя.

Кожная з катэгорый таксама падзяляецца на падгрупы ў залежнасці ад кантэксту, у якім сустракаецца той ці іншы выраз. Напрыклад, *Больш за 20 прадпрыемстваў выкарыстоўваюць натуральныя і штучныя скуру*. Выраз *больш* за патрабуе ад залежнага слова вінавальнага склона.

Таму словаспалучэнне *20 прадпрыемстваў* пішацца ў вінавальным склоне.

Такім чынам, на гэтым этапе атрымаўся наступны спіс (табліца 1):

Табліца 1 – Першапачатковы спіс сабранага матэрыялу па семантычных катэгорыях (фрагмент)

Адзінкі вымярэння	Разрад	Прыклад
Плошча	км/м ²	Яна займае плошчу 207,6 тыс. км ² .
Аб'ём	Тыс/млн м ³	Сумарны памер лесакарыстання можа скласці больш за 19 млн м ³ драўніны.
Маса	Тыс/млн т	У 2010 г. у рэспубліцы было выраблена 4,5 млн цэменту (для параўнання: у 1913 г.33 тыс. т).

3. Экспертная нармалізацыя прадстаўленых семантычных класаў

Большасць ненармалізаваных дадзеных складаюць спалучэнні лікаў з адзінкамі вымярэння. Таму першапачаткова неабходна надаць увагу лікам: перавесці іх у колькасныя ці парадкавыя лічэбнікі (табліца 2). Ужыванне ўсіх лічэбнікаў залежыць ад папярэдняга прыназоўніка ці спалучэння слоў. Так, напрыклад, такія словы як *амаль, прыкладна, складае і інш.* патрабуюць пасля сябе лічэбнікі ў назоўным ці вінавальным склонах. Ніжэй прыведзена табліца прыкладаў прыназоўнікаў з улікам склонаў ужывання лічэбнікаў з іх залежнымі словамі (табліца 3).

Табліца 2 – Пераўтварэнне лікаў у колькасныя лічэбнікі

Запіс лікаў (шаблонамі)	Колькасныя лічэбнікі ў наз. скл.	Канчаткі адзінак вымярэння			
		Мужчынскага роду		Жаночага роду	
1	Адзін/адна	Нулявы канчатак	мільён, працэнт	-а	тысяча
2-4	два, тры, чатыры	-а	мільёна, працэнта	-ы	тысячы
5-20	пяць...дваццаць	-аў	мільёнаў, працэнтаў	нулявы канчатак	тысяч
[2-9]1	дваццаць адзін/адна – дзевяноста адзін/адна	Нулявы канчатак	мільён, працэнт	-а	тысяча
[2-9][2-4]	дваццаць дзве – дзевяноста чатыры	-а	мільёна, працэнта	-ы	тысячы
[2-9][5-9]	Дваццаць пяць тысяч – дзевяноста тысяч	-аў	мільёнаў, працэнтаў	нулявы канчатак	тысяч
[1-9]01	Сто адзін/адна – дзевяцьсот адзін/адна	Нулявы канчатак	мільён, працэнт	-а	тысяча
[1-9]0[2-4]	Сто два/дзве – дзевяцьсот чатыры	-а	мільёна, працэнта	-ы	тысячы
[1-9]0[5-9]	Сто пяць – дзевяцьсот дзевяноста дзевяць	-аў	мільёнаў, працэнтаў	нулявы канчатак	тысяч

Табліца 3 – Спіс канчаткаў адзінак вымярэння ў спалучэнні з колькаснымі лічэбнікамі

Прыназоўнікі	Склон ужывання лічэбнікаў	Канчаткі адзінак вымярэння			
		Канчаткі мужчынскага роду		Канчаткі жаночага роду (тысячы)	
		мн. л.	адз. л.	мн. л.	адз. л.
Empty	N, V (назоўны, вінавальны)	нул. к.	-аў/оў	-у	нул. к.
у/ў	P (месны)	-е	-ах	-ы	-ах
Па/праз	V (вінавальны)	-ы	-оў	-ы	нул. к.
на	T (творны)	-е	-ах	-ы	-ах
3-за/з/каля/да/ад	R (родны)	-а	-аў/оў	-ы	нул. к.

Акрамя лічэбнікаў у працэсе нармалізацыі прыкладаў аўтары таксама сустрэліся з наступнымі пытаннямі:

1. Вызначэнне прынцыпу нармалізацыі абрэвіатур. Усе абрэвіатуры падзелены на дзве групы: тыя абрэвіатуры, што ўтвораны з пачатковых літар элементаў зыходнага словазлучэння і чытаюцца не па алфавітных назвах літар, а як звычайнае слова, называюцца акронімамі (*ЮНЭСКА, НАТА, ААН і інш.*); абрэвіатуры, якія ўтвораны часткова з назваў пачатковых літар, часткова з пачатковых гукаў слоў зыходнага словазлучэння, называюцца літарна-гукавымі (*СССР, ВУП, ВКЛ і інш.*). Складанаскарочаныя абрэвіатуры нармалізуюцца як звычайныя словы (*ЛітБел, БелАз і інш.*).

2. Нармалізацыя дробных лікаў. Такія лікі нармалізуюцца ў залежнасці ад кантэксту, а менавіта ад папярэдняга прыназоўніка (*каля 1/3 (адной трэцяй) насельніцтва, на 1/3 (адну трэцюю) частку насельніцтва*).

3. Вызначэнне прынцыпу нармалізацыі скарачэнняў. З тымі скарачэннямі, які маюць зафіксаваны выгляд, не ўзнікла пытанняў. Усе яны маюць адну словаформу: *г. д. (гэтак далей), т. п. (таму падобнае)* і інш. Цяжэй апрацаваць скарачэнні, форма якіх залежыць ад азначаемага слова ці спалучэння слоў (*гг., ст., стст., р. і інш.*). Яшчэ больш пытанняў узнікае, калі скарачэнне можа прымаць розныя значэнні. Напрыклад, скарачэнне *г.* можа перакладацца як *горад, гара і год*. У такіх выпадках усё залежыць ад кантэксту.

4. Нармалізацыя адзінак вымярэння пасля дробных лікаў. Усе адзінкі вымярэння апрацоўваюцца ў залежнасці ад дробнай часткі лічэбніка (*55,3% - пяцьдзесят пяць цэлых тры дзясяты працэнта*).

5. Нармалізацыя дадзеных з прамежкамі. Яшчэ адной складанасцю з’яўляюцца адзінкі вымярэння ў нейкі пэўны перыяд. Напрыклад, тыя дадзеныя, перад якімі ўжываецца прыназоўнік, расшыфроўваюцца ў неабходным склоне праз коску: *у 2-3 (два, тры) разы*. Тыя лікі, што ўжываюцца ў назоўным ці вінавальным склоне, таксама ў круглых дужках, нармалізуюцца з дапамогай спалучэння прыназоўнікаў *з...на, ад ...да (складаюць 200-1000 чалавек – складаюць ад ста да тысячы чалавек)*.

Такім чынам, атрымаўся спіс нармалізаваных выказаў геаграфічнага дамена з вызначэннем семантычных катэгорый і іх груп.

4. Распрацоўка алгарытму лінгвістычнай апрацоўкі і нармалізацыі тэкстаў на прыкладзе семантычнай групы “Насельніцтва”

У выніку праведзеных даследаванняў, аўтары артыкула распрацавалі наступны алгарытм для нармалізацыі дадзеных семантычнай катэгорыі “Шчыльнасць насельніцтва”. Алгарытм складаецца з наступных крокаў:

1. Ідэнтыфікаваць адзінкі вымярэння шчыльнасці насельніцтва [Гецэвіч, 2012]. Першапачатковыя паказчыкі семантычнай групы: чалавек, людзі ў спалучэнні з такімі адзінкамі вымярэння, як *тысячы, мільёны, тысяч(ы) на квадратны метр/кіламетр*.

2. Атрымаць структурныя часткі, якія складаюцца з бліжэйшага папярэдняга кантэксту (адзін-два словы), ліка(ў) і адзінак вымярэння насельніцтва ў спалучэнні са словамі чалавек(а) ці людзі(ей).

3. Ідэнтыфікаваць вызначаны кантэкст: ці з’яўляецца гэта прыназоўнікам, дзеясловам, ці гэта круглыя дужкі:

а) калі гэта прыназоўнік (на, каля, з, у і г.д), лік пераўтвараецца ў колькасны лічэбнік у залежнасці ад склона, які патрабуе дадзены прыназоўнік (глядзіце табліцу 3);

б) калі гэта дзеяслоў (*складае, налічвае*), прыслоўе (*амаль, прыблізна*), то лік пераўтвараецца ў колькасны лічэбнік у назоўным ці вінавальным склоне (глядзіце табліцу 3);

в) калі структурная частка знаходзіцца ў круглых дужках, лік пераўтвараецца ў колькасны лічэбнік у назоўным склоне;

г) калі ў дужках знаходзяцца два лікі, напісаныя праз працяжнік без папярэдняга кантэксту, то ў працэсе нармалізацыі трэба выкарыстоўваць такія спалучэнні прыназоўнікаў, як *ад... да...* (лічэбнікі апрацоўваюцца ў родным

склоне), з... на... (адпаведна ў родным і вінавальным склонам) [Hetsevich, 2013].

4. Згенераваць колькасныя лічэбнікі ў адпаведнасці з неабходным склонам (глядзіце табліцу 3).

5. Вызначыць, да якога разраду адносяцца колькасныя лічэбнікі: да цэлых ці дробных састаўных лічэбнікаў. Калі гэта цэлы лічэбнік, адзінкі вымярэння апрацоўваюцца ў залежнасці ад склону, у якім ужываецца лічэбнік. Калі гэта дробны лічэбнік, адзінкі вымярэння апрацоўваюцца ў залежнасці ад дробнай часткі лічэбніка ў неабходным склоне (глядзіце табліцу 2).

6. Згенераваць атрыманыя вынікі і ўнесці ў сказ нармалізаваныя выразы.

Заклучэнне

У дадзеным артыкуле аўтары апісалі падрабязны алгарытм ручнога пошуку і апрацоўкі выказаў геаграфічнага дамена, якія патрабуюць нармалізацыі. На дадзены момант спіс неабходны для тэставання якасці працы сістэм распазнавання і сінтэзу маўлення, але ў далейшым будзе пакладзены ў аснову вырашэння задачы аўтаматызаванага папаўнення базы дадзеных для любога дамена.

Бібліяграфічны спіс

[**Барадзіна, 2015**] Барадзіна, Ю.С. Апрацоўка колькасных выказаў з адзінкамі вымярэння: ад навукова-тэхнічнага тэксту да тэлеметраў / Ю.С. Барадзіна, Ю.С. Гецэвіч // Мова і літаратура ў XXI стагоддзі: актуальныя аспекты даследавання: матэрыялы III Рэсп. навуц.-практ. канф. маладых навукоўцаў / БДУ; пад рэд. П. І. Навойчык. — Мінск: Бел. дзярж. ун-т., 2015. — С. 7-12.

[**Гецэвіч, 2014**] Гецэвіч, Ю.С. Лінгвістычныя рэсурсы для пераўтварэння колькасных выказаў з адзінкамі вымярэння тыпу “лічба-сімвал” у словазлучэнні для беларускай і рускай моў / Ю.С. Гецэвіч, А.М. Скопінава // Карповские научные чтения, выпуск 8: сб. науч. ст.: в 2 ч. / Бел. гос. ун-т; редкол.: А.И. Головня (отв. ред.) [и др.]. — Мінск: “ИВЦ Минфина”, 2014. — Ч. 1. — С. 236-240.

[**Hetsevich, 2013**] Hetsevich, Yu. Identification of Expressions with Units of Measurement in Scientific, Technical & Legal Texts in Belarusian and Russian // Yu. Hetsevich, A. Skopinava // Proceedings of the Workshop on Integrating IR technologies for Professional Search [Electronic resource]. — 2013. Mode of access: http://ceur-ws.org/Vol-968/irps_6.pdf. — Date of access: 24.03.2013

[**Гецэвіч, 2012**] Гецэвіч, Ю.С. Ідэнтыфікацыя выказаў з адзінкамі вымярэння ў навукова-тэхнічных і прававых тэкстах на беларускай і рускай мовах / Ю.С. Гецэвіч, А.М. Скопінава // Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2012): доклады XI Международной конференции (Минск, 15 ноября 2012 г.). — Мінск: ОИПИ НАН Беларуси, 2012. — С. 260–265.

ALGORITHM AND LINGUISTIC RESOURCES FOR TEXT NORMALIZATION OF GEOGRAPHIC DOMAIN

Hetsevich Y.S.*, Lysy S.I.*, Kachan E.S.*, Marakulina P.A*, Krivaltsevich A.V.*

* *United Institute of Informatics Problems, National Academy of Sciences, Minsk, Belarus*

yury.hetsevich@gmail.com

evgeniakacan@gmail.com

stanislau.lysy@gmail.com

marakulina.polina@gmail.com

elena.krivaltsevich@gmail.com

Introduction

This article covers the problem of linguistic processing and text normalization of geographic domain. It introduces steps of symbols categorization, numbers and other cases for text processing.

The problem is that not all characters in text can be perceived and vocalized with automatic algorithms.

Main Part

Materials of Geography course books in the Belarusian language, in terms of “Geography of Belarus” for 10th form of secondary school were studied and analyzed. All materials were collected in a non-systematic fashion with three examples on each semantic subgroup.

Next categories were extracted: measurement units, percent, permille, acronyms, numbers, mathematical characters, abbreviations, initials, date, random data, statistical data.

Also the algorithm of categorization of characters, numbers and other cases for text processing was proposed.

Conclusion

In this article, authors explained in detail the algorithm of manual search and processing of expressions in geographical domain for the purpose of normalization.

All this materials should be used as a basis for automatic appending of database for any domain in future.



УДК 004.822

СЕМАНТИЧЕСКИЙ АНАЛИЗ ТЕКСТА ДЛЯ РАСПОЗНАВАНИЯ ЭЛЕМЕНТОВ ВНЕШНЕГО ВИДА ЧЕЛОВЕКА

Долбин А.В. *, Розалиев В.Л. *, Орлова Ю.А. *, Заболеева-Зотова А.В. **

**Волгоградский государственный технический университет,
г. Волгоград, Россия*

sizeof.void34@gmail.com

vladimir.rozaliiev@gmail.com

yulia.orlova@gmail.com

***Российский фонд фундаментальных исследований, г. Москва, Россия*

zabzot@gmail.com

В работе приводится описание различных методов обработки текста на естественном языке. Большинство описанных методов относятся к методам, основанным на машинном обучении. Рассматривается метод распознавания именованных сущностей с разрешением кореференции местоимений в третьей лице. Затем приводится использование латентного размещения Дирихле и латентно-семантического анализа.

Ключевые слова: метод опорных векторов; ЛСА; ЛДА; распознавание именованных сущностей.

Введение

Распознавание элементов внешнего вида человека относится к категории задач информационного поиска. В данной статье рассматривается задача распознавания именованных сущностей и извлечения фактов и неструктурированного текста, составленного на естественном языке.

Задача семантической обработки текста появилась относительно недавно. Но несмотря на это уже является чрезвычайно актуальной.

Результаты семантического анализа текста могут быть использованы в чрезвычайно большом спектре научных областей. Несмотря на свою востребованность, семантический анализ текста является одной из сложнейших математических задач. Это заключение вытекает из следующих факторов:

- естественный язык не формализован и этим обуславливается целый ряд сложностей в понимании текста;
- возможно неоднозначное толкование один и тех же слов. Так как одно и то же слово может менять свой смысл в зависимости от контекста, то и программа «понять», как интерпретировать данное слово;

- производительность алгоритмов анализа текста.

Главная цель статьи – извлечение из текстов фактов о внешнем виде человека. Однако используемые методы извлечения фактов из текста могут быть использованы и для других областей.

1. Поиск именованной сущности «человек» в тексте.

1.1. Описание метода поиска личности.

Извлечение объектов и фактов из текста является задачей NLP и непосредственно Textmining. В данном разделе описывается способ нахождения личностей в тексте на русском языке. Данный способ также можно адаптировать под большинство современных языков.

Основная идея для поиска сущностей в тексте – использование контекстных правил и регулярных выражений. Данный метод относится к категории обучения с учителем. Обучение с учителем – один из способов машинного обучения, в ходе которого система обучается с использованием заранее составленной выборки. На основе этой выборки требуется установить зависимость между данными и на выходе получить точный ответ.

В качестве обучающей выборки используется обычный текст, составленный на русском языке. Программе требуется считать входные данные и из каждого образца обучающих данных (абзац, предложение) составить регулярное выражение особого типа.

1.2. Применение контекстных правил для поиска личности.

Первое, что требуется сделать – заменить искомую сущность (в данном случае упоминание о личности) на специальный символ (например, {P}). В рамках решения задачи нахождения человека в тексте это единственная необходимая ручная операция.

Следующим этапом является применение над обучающей выборкой алгоритмов графематического анализа для разделения текста на отдельные предложения и слова [Солошенко и др., 2014]. Как только все слова разделены, то при помощи доступного словаря или корпуса русского языка нужно получить часть речи слова.

Основная часть формирования регулярных выражений – часть речи слова. К примеру, рассмотрим предложение:

«У {P}, сидящей напротив, очень выразительный взгляд».

Из данного предложения можно сформировать следующее правило:

«PREP? PERSON PRTF ADVB+ выразительный взгляд».

Знак вопроса означает, что в данном случае предлог можно опустить, т.к. он употребляется на первом месте в предложении. Слово PERSON – потенциальная личность в тексте. Знак «+» означает, что слово с данной частью речи употребляется один или более раз подряд. Ключевые слова, которые относятся к тематике поиска (в данном случае рассматривается внешний вид человека) заносятся в отдельный список и никак не интерпретируются. Дополнительно, ключевые слова могут сопровождаться в выражении логической операцией или «|» Под специальной группой символов и находится искомая сущность. Очевидным плюсом данного подхода является то, что он не зависит от каких-либо грамем искомой сущности. В приведенном выше примере под личностью может подразумеваться как и слово «девушка», так и чье-либо имя. [Mikheev, 1999]

Для решения задачи данным способом требуется решить две основные проблемы:

- необходим достаточно большой объем обучающих данных, иначе система не сможет составить достаточное количество регулярных выражений;

- обучающая выборка должна быть обработана вручную.

1.3. Использование метода опорных векторов для разрешения кореференции в третьем лице.

В рамках данной работы рассматривается только разрешение кореференции местоимений в третьем лице, т.к. это один из наиболее простых случаев. Кореференция – связь нескольких отсылок в тексте к одному объекту. Для разрешения кореференции применяется метод опорных векторов. Метод опорных векторов относится к методам обучения с учителем. Следует рассматривать задачу бинарной классификации, т.к. пространство можно разделить на 2 класса: «является кореференцией»/«не является кореференцией».

В методе опорных векторов необходимо выбрать прямую, максимально удаленную от группы точек. Расстояние от этой прямой до каждой точки – максимально. Если такая прямая существует, то ее называют гиперплоскостью. Опорные вектора – это точки, расстояние до которых от гиперплоскости.

$$\frac{1}{\|w\|} \quad (1)$$

Метод опорных векторов строит классифицирующую функцию:

$$F(x) = \text{sign}(\langle w, x \rangle + b). \quad (2)$$

w – нормальный вектор к разделяющей гиперплоскости, b – вспомогательный параметр, треугольные скобки – скалярное произведение. Необходимо выбрать такое w и b , которые максимизируют расстояние до каждого класса. Таким образом, необходимо решить задачи оптимизации.

Для реализации SVM также требуется обучающая выборка, размеченная вручную. Для выборки необходимо специальными символами разметить антецедент и потенциальную анафору. А также, к какому из двух классов относится каждый обучающий набор данных.

Были выделены следующие параметры для метода опорных векторов:

- количество предложений, разделяющих анафору и антецедент;
- стоит ли антецедент в именительном падеже;
- расположение анафоры в предложении (ближе к началу или концу предложения);
- расположение антецедента в предложении (ближе к началу или концу предложения);
- количество существительных и местоимений, расположенных в предложениях;
- совпадает ли падеж анафоры и антецедента;
- совпадает ли род анафоры и антецедента;
- совпадает ли число анафоры и антецедента.[Толпегин, 2006]

2. Извлечение фактов о внешнем виде человека.

2.1. Латентно-семантический анализ

Латентно-семантический анализ – метод обработки текстовой информации, анализирующий взаимосвязь между коллекцией терминов и документов. Основная задача данного метода – нахождение документов, которые наиболее близки в векторном пространстве к поисковому слову. ЛСА применяется для индексирования текста на естественном языке. Особенность ЛСА – частичное снятие омонимии с индексируемых слов. [Орлова и др., 2015]

Алгоритм латентно-семантического анализа:

- составить частотную матрицу термины на документы;
- стоит ли антецедент в именительном падеже;
- над частотной матрицей применить метод оценки релевантности TF-IDF для получения более правдоподобных результатов; [Маннинг и др., 2011]
- использование сингулярного разложения над частотной матрицей на матрицы U , S , Vt ;
- Сократить количество строк в матрице Vt до 2. Для матрицы Усократить количество столбцов до 2;
- По матрицам Vt , U определить координаты ключевого параметра.

Входной текст для латентно-семантического анализа:

«У Ольги светлые волосы и голубые глаза (1). Ногти у нее покрашены красным лаком (2). Она носит туфли на высоком каблуке (3). В её сумке всегда найдется шоколадка (4). У нее есть любимый кот по кличке Порш (5). В понедельник утром ей снова на работу (6).». Ключевое слово для поиска – «волосы».

Таблица 1 – Результат работы латентно-семантического анализа

Номер предложения	Координаты	Индекс
0	(-1; 0)	0.0
2	(0; 0)	1.0
3	(0; 0)	1.0
4	(0; 0)	1.0
5	(0; 0)	1.0
1	(0; -1)	1.4142

Согласно таблице 1, ЛСА точно нашел искомый документ, максимально релевантный к заданному запросу.

2.2. Латентное размещение Дирихле

Латентное размещение Дирихле – порождающая модель, позволяющая объяснять результаты обработки данных с помощью неявных групп. В ЛДА каждый документ рассматривается как набор

различных тематик. Количество тематик является один из выходных параметров данного метода.

Входной текст для модели латентного размещения Дирихле в текущем примере тот же, что и при рассмотрении ЛСА. Набор ключевых терминов: «светлые, волосы, покрашены, красным, лаком, высоким, найдется, шоколадка, глаза, носит, голубые, кот, суббота, работа, дел, идти».

В итоге на выходе метод определяет четыре тематики по входному тексту:

- кот, шоколадка, лаком, красным;
- найдется, волосы, идти, дел;
- голубые, глаза, светлые, волосы;
- идти, носит, высоким, дел.

В рамках одного предложения или абзаца может быть упомянуто несколько тем. Но методы кластеризации документов по темам это не могут учесть. В связи с этим и применяется метода латентного размещения Дирихле. В отличии от обычной кластеризации, в методе для каждого заданного слова по распределению выбирается тема. Латентное размещение Дирихле относится к методам машинного обучения.

В результате обучения модели получаются векторы, отображающие как распределены в каждом документе заданные темы и векторы, отображающие, какие ключевые термины наиболее вероятны в той или иной теме. В итоге можно получить информацию о темах в рамках документа и о списке слов, характерных для данной темы.

Заключение

Таким образом, на данном этапе все описанные методы используется для общей цели – распознавания элементов внешнего вида человека по тексту на естественном языке. В качестве основного механизма будет использоваться латентно-семантический анализ, т.к. он обладает более точными результатами, по сравнению с ЛДА [Коляда и др., 2014]. Преимущество латентного размещения Дирихле заключается в том, что с его помощью можно определить неявные элементы именованной сущности. К примеру, для распознавания внешнего вида возможно нахождение фразеологизмов. Метод разрешения кореференции в дальнейшем планируется расширить и не ограничиться только местоимениями в третьей лице.

Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов №15-07-05440, 15-07-07519, 15-37-70014, 16-07-00453.

Библиографический список

- [Mikheev, 1999]Named Entity Recognition Without Gazetteers / Andrei Mikheev [and etc.]; - 9th Conference on the European Chapter of the Association for Computational Linguistics, Stroudsburg, PA, 1999, pp. 1-8.
- [Толпегин, 2006]Толпегин, П. В. Алгоритм автоматизированного разрешения анафоры местоимений

третьего лица на основе методов машинного обучения. [Электронный ресурс] // Режим доступа : <http://www.dialog-21.ru/digests/dialog2006/materials/html/Tolpegin.html>, свободный. — Загл. с экрана. (18.06.2014).

[**Орлова и др., 2015**] Орлова, Ю. А. Автоматизация составления портретных изображений по естественно-языковому описанию / Орлова Ю. А., Долбин А. В., Кипаева Е. В., Розалиев В. Л. // Известия ВолгГТУ. Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. – Волгоград, 2015. - № 2 (157). – с 71-76.

[**Маннинг и др., 2011**] Маннинг, К. Д. Введение в информационный поиск / К. Д. Маннинг, П. Рагхаван, Х. Шютце; пер. с англ. под ред. П. И. Браславского, Д. А. Ключина, И. В. Сегаловича. - Москва.: И. Д. Вильямс, 2011. — 528 с.

[**Коляда и др., 2014**] Коляда, А. С. Применение латентного разложения Дирихле для анализа публикаций в наукометрических базах данных / А. С. Коляда, В. А. Яковенко, В. Д. Гогунский // Одесский национальный политехнический университет. – 2014. – Вып. 1. – с 186 – 191.

[**Солошенко и др., 2014**] Thematic Clustering Methods Applied to News Texts Analysis / Солошенко А.Н., Орлова Ю.А., Розалиев В.Л., Заболеева-Зотова А.В. // Knowledge-Based Software Engineering : Proceedings of 11th Joint Conference, JCKBSE 2014 (Volograd, Russia, September 17-20, 2014) / ed. by A. Kravets, M. Shcherbakov, M. Kultsova, Tadashi Iijima ; Volgograd State Technical University [etc.]. – [Б/м] : Springer International Publishing, 2014. – P. 294-310. – (Series: Communications in Computer and Information Science ; Vol. 466).

SEMANTIC ANALYSIS OF TEXT FOR RECOGNITION OF THE ELEMENTS OF HUMAN APPEARANCE

Dolbin A.V. *, Rozaliev V.L. *, Orlova Y.A.
*, Zabolzeva-Zotova A.V. **

**Volograd State Technical University,
Volograd, Russia
sizeof.void34@gmail.com
vladimir.rozaliev@gmail.com
yulia.orlova@gmail.com*

*** Russian Foundation for Basic Research,
Moscow, Russian Federation
zabzot@gmail.com*

The paper describes the various methods of processing natural language. Most of these methods are based on machine learning. The method of recognition of named entities with coreference resolution of pronouns in the third person is described. Then, given the use of the latent Dirichlet allocation and latent semantic analysis.

Introduction

Recognition of elements of the appearance of a man falls into the category of information search. This article discusses the problem of named entity recognition and fact extraction from unstructured text in a natural language.

The problem of semantic text processing is a relatively new. But in spite of this is an extremely urgent.

The results of the semantic analysis of text can be used in a very large range of scientific fields. Despite its relevance, semantic analysis of the text is one of the

most complex mathematical problems. This conclusion stems from the following factors:

- natural language is not formalized, and this caused a number of difficulties in understanding the text;
- possibly ambiguous the same words. Since the same word can change its meaning depending on the context, and the program should know how to "understand" the word;
- performance issue.

The main purpose of the article - the extraction of facts about the appearance of a person. However, methods that were used for facts extraction from the text can be used for any other theme.

Main Part

This article describes the mathematical methods for the identifying the elements of human appearance. Before processing natural language with these methods, every word must be reduced to the normal form. An alternative solution is the use of the Porter stemmer.

First, you must determine the persons for further processing. To this end, based on the training sample generated contextual rules are then applied to the processed text. This approach has a number of positive sides. The disadvantage of this method is that it is impossible to distinguish a person from any other entity.

Then you need to allow the coreference pronouns. This problem was solved by using the method of support vector machine. Support Vector Machines is classified as machine learning method. In feature extraction space is divided into 2 parts by hyper-plane. Then the model can conclude whether anaphora and antecedent are referenced to the same person.

The final step is the search for elements of the appearance of a man with the key words using the method of latent semantic analysis. After the LSA was used, the latent Dirichlet allocation should be used to determine possible phraseology, that can describe some aspect of human appearance.

Conclusion

Thus, at this stage, all methods used for the common goal - recognition of elements of the appearance of the human in natural language. As a main mechanism latent semantic analysis should be used, as it has more accurate results, compared with the LDA [Kolyada et al., 2014]. The advantage of placing the latent Dirichlet allocation is that it can be used to determine the implicit elements of a named entity. For example, to detect the possible presence of phraseology in human appearance. Coreference resolution method in the future is planned to expand and not be limited only in the third person pronouns.



УДК 534.87; 534.4

О ФИЗИЧЕСКОЙ СТРУКТУРЕ ПРОСТЫХ ГЛАСНЫХ ЗВУКОВ РЕЧИ ЧЕЛОВЕКА

Митянок В.В.

*Полесский государственный университет
г. Пинск, Республика Беларусь*

Mitsianok@mail.ru

В статье описываются результаты численных экспериментов по разложению гласных звуков речи человека на составляющие их звуковые моды с дрейфующими амплитудами. Полученные моды вновь суммировались, при этом предварительно подвергались сознательным искажениям с целью выявления факторов, как имеющих значение так и не имеющих значение для распознаваемости звуков.

Ключевые слова: распознавание речи, синтез речи, фазовый анализ звуков.

Введение

В последнее время появились сведения о том, что дельфины обладают разумной речью. [Janik и др., 2006; Kassewitz, 2015]. В частности, дельфины имеют имена, которые сородичи дают им при рождении, и по которым они обращаются друг к другу. Некоторые из звуков, издаваемых дельфинами, уже расшифрованы. Таким образом, в перспективе люди рано или поздно займутся масштабным изучением этой речи и, как следствие, ее семантикой. Однако в отношении дельфинов проблема семантики представляется намного более сложной, нежели в отношении человека, поскольку совершенно неизвестны ни мировоззрение дельфинов, ни их мышление. А оно, в свою очередь может быть понято лишь после установления контакта с ними. То есть возникает как бы замкнутый круг проблем. Проблемы усугубляются огромной разницей в частотных диапазонах речи человека и звуковых сигналов дельфинов – эти диапазоны перекрываются лишь частично, звуки дельфинов в основном лежат в ультразвуковой (по мнению человека) области частот.

Автор настоящей статьи разработал и демонстрирует на конференциях и научных семинарах компьютерную программу «перековки» голоса человека на другой частотный диапазон. Так, мужской голос может быть «перекован» на женский и даже детский, и наоборот.

Причем перековка происходит безо всякого ущерба для семантики. Это может быть шагом к унификации частотных диапазонов людей и дельфинов. Но унификация частотных диапазонов –

это лишь часть дела. Необходимо выявить структуру основных звуковых единиц голосов дельфинов. А таковые в настоящее время не вполне ясны даже в отношении людей. Так, например, считается, что поскольку ухо человека не реагирует на фазы различных составляющих мод, то и в составе звуков, издаваемых человеком нет никаких фазовых закономерностей. Но это мнение было оспорено [Митянок и др., 2013].

Поэтому начинать нужно именно с выявления математических особенностей различных звуковых единиц как человека так и дельфинов. Представляет интерес вопрос о том, что именно делает звук «А» звуком «А», звук «О» звуком «О» и т.д. Какие именно математические характеристики звуков здесь существенны, какие привнесены несовершенством аппарата речеобразования человека, какие позволяют отличать одного диктора от другого, а какие вообще ни за что не несут ответственности, и попали в состав звуков случайно.

Расшифровка математических особенностей различных звуков речи человека – это ключ к расшифровке математических особенностей звуков речи дельфинов и, в более отдаленной перспективе, к пониманию их семантики.

1. Метод аппроксимации

Как известно, метод преобразований Фурье, используемый для нахождения спектра звуков, обладает рядом недостатков. В частности, в спектре присутствуют фальшивые линии, линии спектра даже в случае идеального гармонического сигнала, но рассмотренного на ограниченном

интервале времени, размыты. (В квантовой механике это обстоятельство является математической подоплекой соотношения неопределенностей). Спектр сигнала существенно зависит от его длительности. Если в исследуемом сигнале присутствуют малоинтенсивные моды, то они могут оказаться скрытыми под фальшивыми линиями. Поэтому в [Митянок, 2008, 2009] была поставлена задача нахождения спектра звуковых сигналов методом аппроксимации, с учетом того, что звуковые сигналы, соответствующие отдельным звукам речи человека представляют собой сумму мод, параметры которых (амплитуды, частоты, фазы) могут слегка меняться в процессе звучания, дрейфовать, дрожать, то есть зависеть от времени. Метод основан на функционале

$$S = \sum_{i=1}^n [y(t_i) - y_1(t_i)]^2 + \alpha \sum_{k=1}^{n-1} (b_{0,i} - b_{0,i+1})^2 + \alpha \sum_{k=1}^l \sum_{i=1}^{n-1} (a_{k,i} - a_{k,i+1})^2 + \alpha \sum_{k=1}^l \sum_{i=1}^{n-1} (b_{k,i} - b_{k,i+1})^2, \quad (1)$$

где $y(t_i)$ — зависящая от времени аппроксимируемая функция, описывающая сигнал, заданная своими значениями в n последовательных моментах времени от t_1 до t_n , a

$$y_1(t_i) = b_{0,i} + \sum_{k=1}^l [a_{k,i} \sin(\omega_k t_i) + b_{k,i} \cos(\omega_k t_i)], \quad i=1..n \quad (2)$$

— аппроксимирующая функция, $b_{0,i}$ — дрейфующее начало отсчета, $a_{k,i}$, $b_{k,i}$ — дрейфующие амплитуды синус- и косинус- волн (параметры аппроксимирующей функции), ω_k — их несущие частоты, l — количество волн (мод) в аппроксимирующей функции. В (1) и (2) для простоты можно принять $t_i = i$, хотя это и не обязательно. Параметр α в (1) позволяет сглаживать изменения амплитуд волн при переходе от точки к точке. Чем больше значение α , тем более гладкими являются амплитуды волн.

Вычисляя частные производные (1) по дрейфующим амплитудам и по дрейфующему началу отсчета и приравнивая результаты нулю, получим систему линейных алгебраических уравнений относительно параметров аппроксимирующей функции. Решив эту систему, найдем эти параметры и тем самым произведем разложение аппроксимируемой функции на сумму волн с медленно меняющимися амплитудами. Найденные таким путем $b_{0,i}$, $a_{k,i}$, $b_{k,i}$ можно вновь подставить в (2) и произвести численное суммирование. Полученную таким путем аппроксимирующую функцию естественно назвать

восстановленным звуком. Если затем вычесть восстановленный звук из исходного звука и подвергнуть разность преобразованиям Фурье, то выясняется, что часто существуют еще какие-то несущие частоты, которые не были замечены при первом разложении в ряд (интеграл) Фурье по причине малой интенсивности несомых ими мод. В частности, этим способом в [Митянок, 2014] было установлено, что в спектре звуков «З», «ЗЬ», «Ж», «ЖЬ» присутствуют полуцелые (по отношению к базовой) несущие частоты.

Каждую из мод, входящую в (2) можно переписать в физически более информативном виде:

$$a_{k,i} \sin(\omega_k t_i) + b_{k,i} \cos(\omega_k t_i) = c_{k,i} \sin(\omega_k t_i + \varphi_{k,i}), \quad k=1..l, i=1..n \quad (3)$$

и тогда аппроксимирующая функция выглядит так:

$$y_1(t_i) = b_{0,i} + \sum_{k=1}^l c_{k,i} \sin(\omega_k t_i + \varphi_{k,i}). \quad i=1..n \quad (4)$$

Здесь $c_{k,i}$ — дрейфующая общая амплитуда моды, $\varphi_{k,i}$ — дрейфующая фаза моды.

2. Анализ простых гласных речи человека.

Изучались звуки «А», «О», «У», «Э», «Ы», «И». Эти звуки были отобраны для изучения по той причине, что их можно произносить достаточно долго и от этого они не теряют свою индивидуальность в отличие от звуков «Я», «Е» и других, которые при длительном звучании преобразуются соответственно в звуки «А», «Э» и т.д. Для изучения вышеуказанных звуков, методом преобразований Фурье определялась в нулевом приближении система несущих частот, нижняя из которых назначалась базовой. Во – вторых, эта система несущих частот дополнялась теми частотами, которые остались незамеченными методом Фурье при первом его использовании. Для этого использовался вышеописанный способ. В третьих, эта система частот дополнялась полуцелыми частотами, составляющими 0.5, 1.5, 2.5, 3.5 от базовой. В результате получалась ловящая сеть, которая и использовалась для окончательного разложения звуков на моды. При анализе разложенных звуков выявилось следующее:

Общие амплитуды различных мод заметно нестабильны в процессе звучания (рисунок.1). Их нестабильность носит хаотический характер, между дрейфующими (плавающими, болтающимися) амплитудами не прослеживается никакой связи.

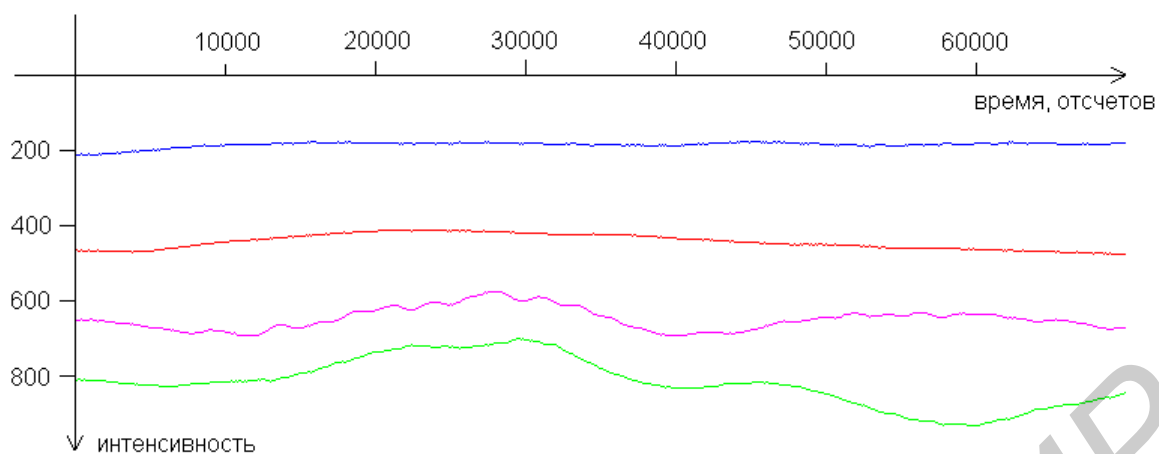


Рисунок 1. Амплитуды нижних мод, несомых целыми частотами. Звук «О», респондент Митянок. Частота дискретизации 44100 Гц. Базовая мода – красный цвет, вторая мода – фиолетовый, третья – зеленый, четвертая – синий. Образец N5

Напротив, между фазами отдельных мод такие связи прослеживаются достаточно хорошо (рисунки 2,3)

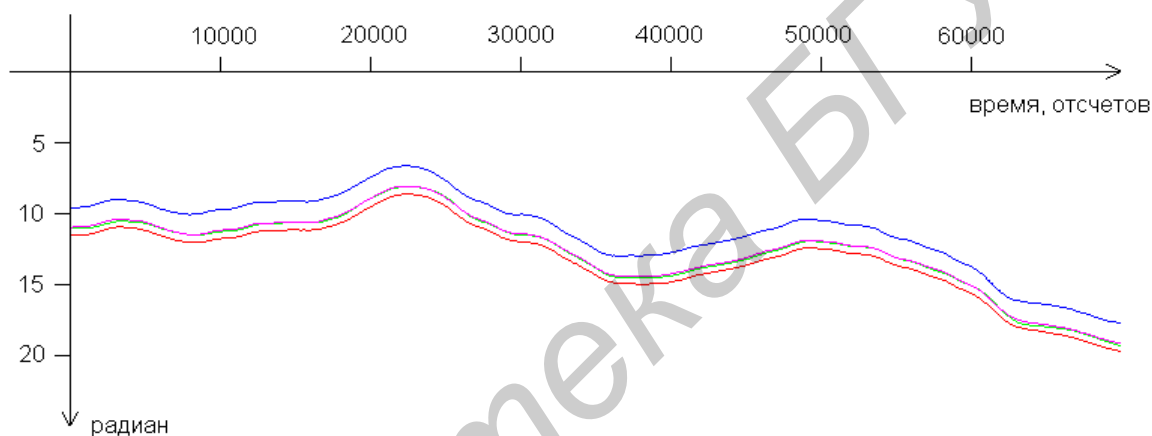


Рисунок 2. Фазы нижних мод, несомых целыми частотами, деленные на номер частоты (нормированные фазы). Звук «О», респондент Митянок. Частота дискретизации 44100 Гц. Базовая мода – красный цвет, вторая мода – фиолетовый, третья – зеленый, четвертая – синий. Образец N5.

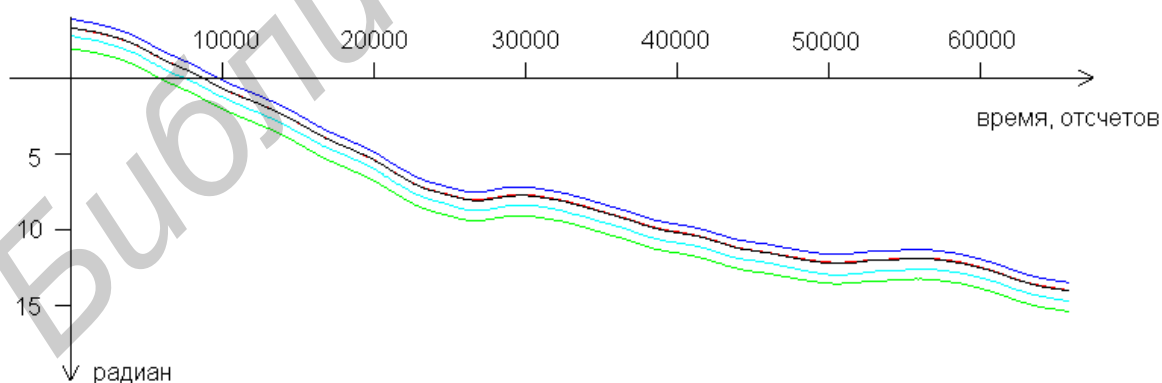


Рисунок 3. Фазы мод, несомых целыми частотами, деленные на номер частоты (нормированные фазы). Звук «Э», респондент Янковский. Частота дискретизации 44100 Гц. Базовая мода – красный цвет, вторая мода – фиолетовый, третья – зеленый, четвертая – синий, пятая – бирюзовый.

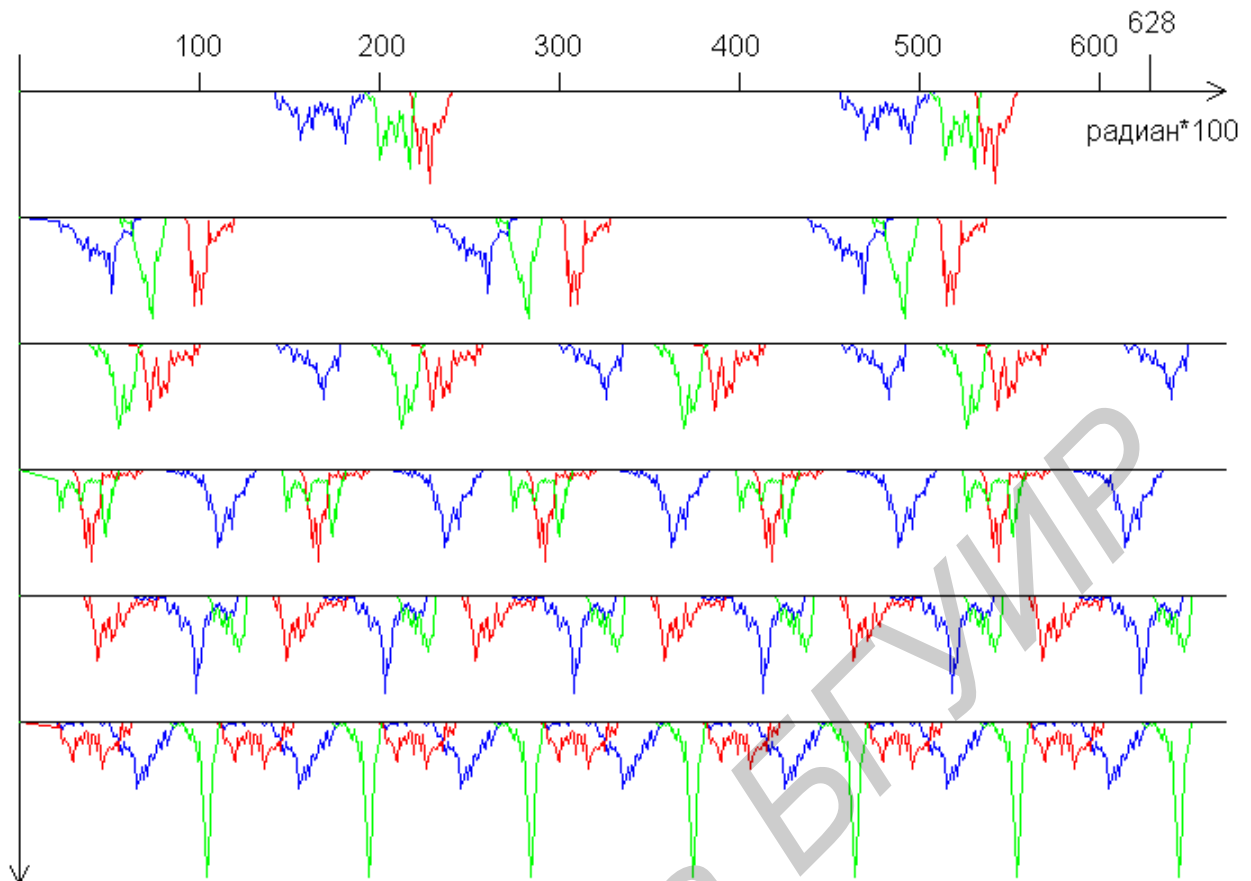


Рисунок 4. Гистограммы фазовых критериев звука А в исполнении респондентов: Коваленко – синие линии, Коновалова – красные, Романова – зеленые. Критерии разнесены по вертикали сверху вниз, начиная от K_2 и кончая K_7 ; Усреднение в (5) произведено при $m=500$. Частота дискретизации 44100 Гц

Как видно из рисунков 2 и 3 расстояние между различными фазами, нормированными на номер моды, практически не зависит от времени. Такое поведение фаз имеет место для всех исследованных звуков и для всех респондентов. Фаза моды, соответствующей базовой частоте может слегка зависеть от времени, нормированные же фазы остальных мод послушно повторяют ее изгибы. Однако расстояния между фазой базовой моды и нормированными фазами остальных мод, хотя и не зависят от времени, но зависят от того, какой именно звук звучит и от того, кто именно этот звук произносит. Это может быть использовано для создания систем идентификации и верификации человека по голосу (рисунок 4). Для этого введем фазовые критерии

$$K_i = \sum_{j=1}^m (\varphi_{1,j} - \frac{\varphi_{i,j}}{i}), i=1..l \quad (5)$$

Здесь m – длина отрезка (в отсчетах дискретизации) избранная для проведения усреднения

При формировании гистограмм результат каждого усреднения рассматривался как одна точка. Периодичность критериев K_i из (5) составляет $2*\pi/i$, что объясняет тот факт, что в верхней строчке рисунка 4 на интервале $[0,2\pi]$ имеется 2

группировки точек, во второй строчке – 3, в третьей – 4 и т.д.

Уже только первая строка рисунка 4 позволяет по нескольким образцам звучания уверенно различать вышеуказанных трех респондентов. Пользуясь аналогичными гистограммами для других звуков речи, можно создать систему, позволяющими различать и намного большее число респондентов.

3. Синтез искусственных звуков

Если по формуле (4) произвести суммирование, используя найденные дрейфующие амплитуды волн, то, как и ожидается, получается звук, который при прослушивании звучит неотличимо от исходного звука. Для того, чтобы ответить на вопрос, что именно делает звук «А» звуком «А», звук «О» звуком «О» и т.д., перед суммированием были проведены математические эксперименты по сознательному искажению общих мод и фаз. Во-первых, фазы всех целых мод, кроме базовой, заменялись на искусственно вычисленные, привязанные к фазе базовой моды:

$$\varphi_{k,i} = k\varphi_{1,i}, \quad k=1..l, i=1..n \quad (6)$$

где k – номер целочисленной моды, $\varphi_{1,i}$ – зависящая от времени (номера отсчета) фаза базовой моды. Звук от такой замены не меняется. Во-вторых, к новой фазе каждой из мод прибавлялось любое случайное число, постоянное, однако, на всем отрезке звучания. От этого звук также не менялся. В третьих, сам факт хаотичного поведения общих амплитуд различных мод различных звуков, полученных от различных респондентов и различных их образцов наводит на мысль о том, что хаотичность – это нечто привнесенное, не имеющее отношения к индивидуальности звуков. Так и оказалось. Оказалось, что дрейфующие общие амплитуды можно усреднить по отрезку звучания, и затем заменить фактические дрейфующие амплитуды их усредненными значениями. Звуковая функция, полученная после такого искажения амплитуд,

звучала так же, как и исходный звук. Звук сохранял свою индивидуальность. В четвертых оказалось, что при суммировании мод можно опустить дрейфующий нуль и полуцелые частоты. И от такого отбрасывания звук не менялся. А вот если фазу каждой из мод на всем отрезке звучания заменить на постоянное, но случайное число, то звук портился. Вместо четкого звука слышалось то, что скорее можно назвать звучанием зуммера. В поисках объяснений были проделаны следующие математические эксперименты. 1. Усредненный амплитудный спектр любого из звуков соединялся в формуле (4) с фазами от любого из других звуков и от любого из других респондентов. От такой операции звук не менялся, звучал четко и соответствовал амплитудному спектру. 2.

Таблица 1. - Значения амплитуд различных мод простых гласных звуков.

Номер моды	А	О	У	Э	Ы	И
1	637	613	1060	566	1757	914
2	375	744	814	540	354	112
3	674	836	303	1007	65	22
4	794	495	0	61	0	0
5	753	51	0	114	25	0
6	180	0	0	123	51	0
7	49	0	0	90	140	0
8	19	0	0	97	32	0
9	15	0	0	183	54	16
10	17	0	28	93	111	49
11	17	10	0	114	14	30
12	21	17	0	120	10	35
13	8	0	0	44	22	71
14	16	0	34	31	92	135
15	16	0	7	42	26	147
16	16	15	17	54	26	110
17	30	30	8	79	8	35
18	34	12	0	45	0	6
19	13	0	0	37	0	8
20	0	0	0	18	7	5
21	0	0	0	45	0	14
22	0	0	0	25	9	21
23	0	12	0	0	6	14
24	0	0	0	0	12	5
25	0	0	10	0	10	9
26	0	0	10	0	8	22
27	0	0	5	0	7	16
28	0	0	3	0	0	24
29	0	11	4	0	0	42
30	0	15	8	0	0	18
31	0	18	13	0	0	14
32	0	18	5	0	0	13
Базовая частота	0.0269	0.0262	0.0305	0.0268	0.0302	0.0291

Примечание: допустимы небольшие (в пределах 10-30 процентов) изменения амплитуд, не влияющие на звук. Возможно также одновременное пропорциональное изменение всех амплитуд некоторого звука – этому соответствует изменение громкости. Данные получены усреднением по 20 образцам длительностью по 2-3 сек.

Перед пересадкой фаз с одного амплитудного спектра на другой, фаза базовой моды аппроксимировалась различными степенными многочленами степени от 15 до 30. (при длительности звучания от 1 до 3 секунд). Результат был тот же – изменений в звучании нет. Так чем же объясняется «порча» звука при замене дрейфующих фаз на константы? Из рисунков 2,3 (и аналогичных для других звуков и для других респондентов) видно, что реальные фазы не являются строгими константами, а как бы дрейфуют (плавают) вокруг неких средних значений с неустойчивым периодом от 1.5 до 2.5 Гц и с неустойчивой амплитудой 0.5-2 радиан. В связи с этим возникло предположение, что именно так и должно быть. Что мозг слушателя уже готов к тому, что диктор будет производить сигнал с испорченной фазой, а звук с неиспорченной фазой мозгом за звук не воспринимается. Когда же в качестве фазы принималась испорченная величина, то звук вновь звучал четко и распознаваемо. Если подытожить все вышесказанное, то получаем, что для синтеза вышеуказанных звуков, вместо (4), как один из вариантов, можно принять формулу:

$$y_i = \sum_{j=1}^{32} C_j \sin[\omega_0 i j + 1.0 j \sin(i / 3300) + r_j], \quad (7)$$

$$i = 1..n$$

где усредненные значения общих амплитуд C_k приведены в нижеследующей таблице, ω_0 – базовая частота, ее значение приведено в последней строчке таблицы, r_k – массив произвольных чисел, n – длина отрезка звучания (в отсчетах дискретизации). За основу получения усредненных общих амплитуд в таблице 1 был взят голос автора. Изменению i в (7) на единицу соответствует изменение реального времени на 1/44100 долю секунды.

Заключение

В результате проведенных математических экспериментов установлено: 1. Звук определяется именно набором амплитуд, которые могут иметь постоянные значения по всему промежутку звучания. Возможные номинальные значения амплитуд приведены в таблице 1. 2. Моды частот, полуволн по отношению к базовой, и дрейфующее начало отсчета несущественны для звуков. 3. Фазы мод, нормированных на номер моды, отличаются от фазы базовой моды на величину, постоянную по всему промежутку звучания, но зависящую от диктора и от произносимого им звука. 4. Прибавка к фазам мод произвольных постоянных чисел не влияет на звучание звуков. Заинтересованный читатель, пользуясь формулой (7) и данными для нее из таблицы (1) сможет самостоятельно подготовить любые из гласных звуков, исследованных в настоящей статье

Библиографический список

- [Janik и др., 2006] Janik, V. Signature whistle conveys identity information of Bottlenose Dolphins/ V.M Janik, L.S. Sayigh, R.S. Wells // Proc. of the Nat. Acad. of Sci. of the USA 103, 21, 2006, pp 8293 – 8297.
- [Kassewitz, 2015] Kassewitz, J. Speak Dolphin, 2015 -28 p.
- [Митянок и др., 2013] Митянок, В.В. Применение фазового анализа звуков речи для распознавания человека по его голосу. [Электронный ресурс] / В.В. Митянок, Н.В. Коновалова //Техническая акустика. – Электрон. журн.- 2013.-4.- Режим доступа: <http://www.ejta.org>, свободный.
- [Митянок, 2014] Митянок, В.В. О физической структуре звуков З, Зь, Ж, Жь [Электронный ресурс] /В.В. Митянок.// Техническая акустика. – Электрон. журн.- 2014.-9.- Режим доступа: <http://www.ejta.org>, свободный
- [Митянок, 2008] Митянок, В.В. О числовых характеристиках некоторых низкочастотных звуков человеческой речи [Электронный ресурс] /В.В. Митянок // Техническая акустика. – Электрон. журн.- 2008.-15.- Режим доступа: <http://www.ejta.org>, свободный
- [Митянок, 2009] Митянок, В.В. Определение числовых характеристик высокочастотных звуков речи на основе аппроксимации гармоническими функциями. / В.В. Митянок // Известия НАН Беларуси, сер. ф.-м.н.-2009.-, №2- с.111-118.

ABOUT THE PHYSICAL STRUCTURE OF SIMPLE VOWEL SOUNDS OF HUMAN SPEECH

Mitsianok V.V.

Polessian St. Univ
Pinsk, Belarus

Mitsianok@mail.ru

Approximation method is used for decomposition of the simple vowel sounds of human speech onto the set of the different frequencies partial waves, and for creating the artificial sounds. It is found, that before summation all the modes, their amplitudes may be averaged over the entire duration of sounds, so they may be constant values. The table of appropriate amplitude figures is presented. But appropriate phases should not be constants, but a bit spoiled constants. For regular phases sound sounds unnatural, but for slightly damaged phases the sounds are natural. The formula for synthesis of sounds is given. An interested reader can prepare artificial sounds by using this formula and table of averaged amplitudes.



УДК 004.822:514

МЕТОДИКА ГОЛОСОВОЙ ИДЕНТИФИКАЦИИ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ

Меньшаков П.А., Мурашко И.А.

*Гомельский государственный технический университет имени П.О. Сухого
Гомель, Республика Беларусь*

pmenshakov@gmail.com

iamurashko@tut.by

На данный момент основным средством контроля доступа являются пропускные пункты, оборудованные различными средствами контроля доступа. Но большинство из средств контроля доступа имеют высокую цену. Причем большая часть расходов приходится на выделение персонального средства идентификации каждому пользователю. Решением данной проблемы может стать голосовая идентификация. Использование биометрии позволяет отказаться от чипов и карт доступа, исключить потерю средства идентификации и его кражу. А использование голоса позволит отказаться от дорогостоящего оборудования для считывания данных.

Ключевые слова: голосовая идентификация; биометрия; контроль доступа.

Введение

В настоящее время, голосовая идентификация, как и биометрия в целом, уже получила широкое распространение. Самый простой пример – сканеры отпечатков пальцев, установленные почти на каждом ноутбуке.

Биометрия предполагает систему распознавания людей по одной или более физических или поведенческих черт. В области информационных технологий биометрические данные используются в качестве формы управления идентификаторами доступа и контроля доступа. Также биометрический анализ используется для выявления людей, которые находятся под наблюдением [1]. Так же биометрия предусматривает и поведенческий анализ объекта. К ним относятся ходьба, жесты и т.п.

Довольно длительное время отпечатки пальцев используются для идентификации преступников и предотвращения воровства или мошенничества. Некоторые люди умеют имитировать голоса, но, это требует особых навыков, которые не часто встречаешь в обыденной жизни [2].

1. Аппаратная реализация

Первоначальным этапом голосовой идентификации является получение голоса пользователя. Для этого необходим микрофон, фильтр и аналого-цифровой преобразователь, для дальнейшей работы с цифровой записью голоса.

В общем виде процесс ввода речевых сообщений приведен на рисунке 1.

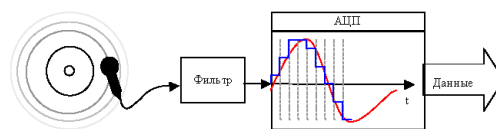


Рисунок 1 – Схема ввода записи голоса

С выхода микрофона сигнал подается на вход блока фильтрации. Следующим этапом является прохождение АЦП. Когда АЦП используется для амплитудного анализа, число, получаемое на выходе АЦП, используется для адресации памяти и называется номером канала, а V - шириной канала. Номер канала несет информацию об амплитудном значении сигнала.

Далее оцифрованный сигнал попадает в блок цифровой обработки. В блоке цифровой обработки сигнал фильтруется и преобразуется в вектор, с которым в дальнейшем будет работать микропроцессор и нейросетевой обработчик.

Так же, полученный вектор заносится в энергонезависимую память. Это необходимо для последующего сравнения с полученным отпечатком.

После сравнения отпечатка в памяти с полученным отпечатком, микроконтроллер подает команду на блок управления внешним устройством, к примеру, на магнитный дверной замок. Общая схема устройства представлена на рисунке 2.

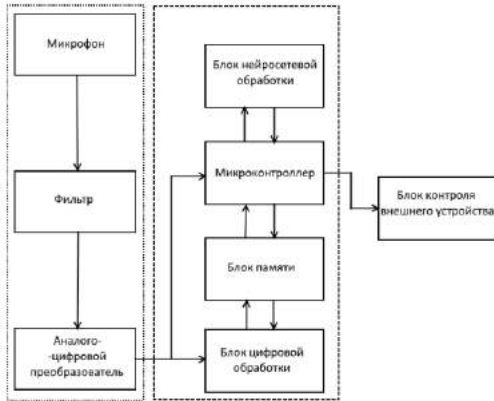


Рисунок 2 – Схема устройства

2. Принцип голосовой идентификации

Сам процесс голосовой идентификации не требователен к ресурсам, и состоит из двух этапов. Сперва, необходимо получить голосовой отпечаток пользователя и преобразовать к виду, в котором его можно будет сравнить с другими. Вторым шагом является сравнение голосовых отпечатков при помощи обученной нейронной сети. Для реализации процесса преобразования необходимо произвести определенный порядок действий.

При помощи микрофона получается запись голоса идентифицируемого и отправляется на ЭВМ. Наиболее оптимальным является получение WAV файла, в виду простоты работы с ним.

Полученную запись голоса необходимо разделить на кадры. Разделение на кадры представлено на рисунке 3. Данное действие необходимо для более простой работы с записанной звуковой дорожкой.

Далее все вычисления будут производиться с каждым кадром в отдельности.

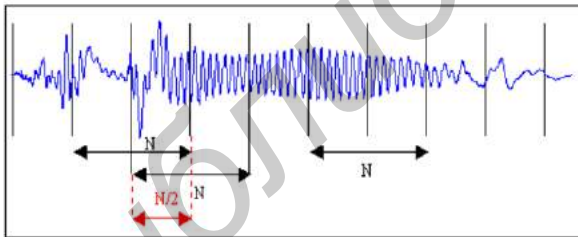


Рисунок 3 – График звуковой волны

Следующим этапом является устранение нежелательных эффектов и шумов. Это необходимо для того, чтобы записи, полученные в разное время соответствовали друг другу независимо от сторонних факторов. Существует множество способов, при помощи которых можно уменьшить шумовые эффекты. Мною использовалось умножение каждого кадра на особую весовую функцию "Окно Хемминга":

$$\omega(n) = 0.53836 - 0.46164 * \cos\left(\frac{2\pi n}{N-1}\right). \quad (1)$$

где n – порядковый номер элемента в кадре, для которого вычисляется новое значение амплитуды,
 N – длина кадра (количество значений сигнала, измеренных за период).

Полученные кадры преобразуется в их частотную характеристику при помощи прогонки через "Быстрое Преобразование Фурье":

$$X_k = \sum_{i=0}^{N-1} x_n e^{-\frac{2\pi i k n}{N}}. \quad (2)$$

где N – длина кадра (количество значений сигнала, измеренных за период),

x_n – амплитуда n -го сигнала,

X_k – N -комплексных амплитуд синусоидальных сигналов, слагающих исходный сигнал.

На сегодняшний день наиболее успешными являются системы распознавания голоса, использующие знания об устройстве слухового аппарата. В виду данных особенностей необходимо привести частотную характеристику каждого кадра к «мелам». Зависимость представлена на рисунке 4.

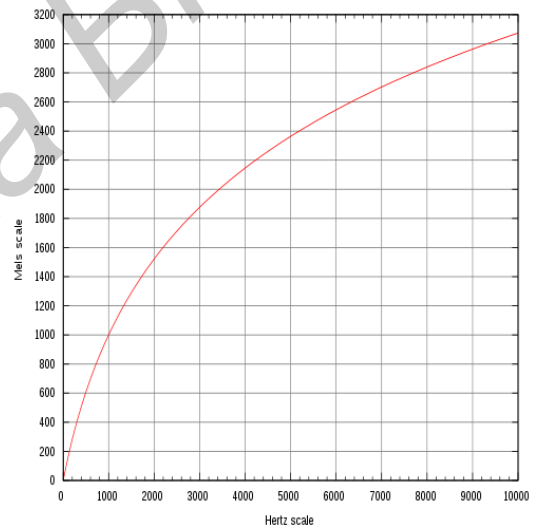


Рисунок 4 – График зависимости высоты звука (в мелах) от его частоты

Для перехода к «мел» характеристике используется следующая зависимость:

$$m = 1127 \log_e \left(1 + \frac{f}{700}\right). \quad (3)$$

где m – частота в мелах,

f – частота в герцах.

Это последнее действие, необходимое для последующего преобразование в вектор характеристики, который, впоследствии, сравнивается с базой голосовых записей. Вектор будет состоять из мел-кепстральных коэффициентов, получить которые можно по следующей формуле:

$$c_n = \sum_{k=1}^K (\log S_k) \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right]. \quad (4)$$

где c_n – мел-кепстральный коэффициент под номером n ,

S_k – амплитуда k -го значения в кадре в мелах,

K – наперед заданное количество мел-кепстральных коэффициентов $n \in [1, K]$.

Полученный вектор характеристик добавляется в базу данных, для последующего сравнения с ним.

Однако более оптимальным вариантом является использование нескольких записей одного и того же голоса. Заранее определенное количество образцов голоса можно использовать для обучения нейронной сети.

В работе использовалось обучение без учителя, так как оно является намного более правдоподобной моделью обучения в биологической системе. Развита Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с predeterminedными идеальными ответами, а обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т.е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор [3]. Схематически данная сеть изображена на рисунке 5.

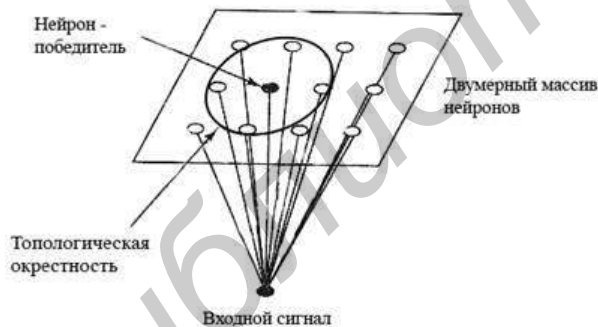


Рисунок 5 – Схема сети Кохонена

Распространение сигнала в такой сети происходит следующим образом: входной вектор нормируется на 1.0 и подается на вход, который распределяет его дальше через матрицу весов W . Каждый нейрон в слое Кохонена вычисляет сумму на своем входе и в зависимости от состояния окружающих нейронов этого слоя становится активным или неактивным (1.0 и 0.0). Нейроны этого слоя функционируют по принципу конкуренции, т.е. в результате определенного количества итераций активным остается один нейрон или небольшая группа. Этот механизм называется латеральным. Так как обработка этого

механизма требует значительных вычислительных ресурсов, в моей модели он заменен нахождением нейрона с максимальной активностью и присвоением ему активности 1.0, а всем остальным нейронам 0.0. Таким образом, срабатывает нейрон, для которого вектор входа ближе всего к вектору весов связей.

Если сеть находится в режиме обучения, то для выигравшего нейрона происходит коррекция весов матрицы связи по формуле:

$$w_n = w_n + \alpha (x - w_n). \quad (5)$$

где w_n – новое значение веса;

w_m – старое значение;

α – скорость обучения;

x – величина входа.

Геометрически это правило иллюстрирует рисунок 6.

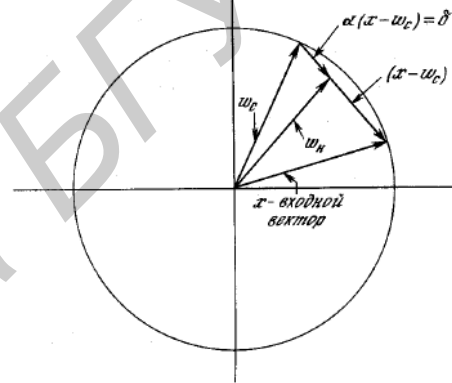


Рисунок 6 – Коррекция весов нейрона Кохонена

Так как входной вектор x нормирован, т.е. расположен на гиперсфере единичного радиуса в пространстве весов, то при коррекции весов по этому правилу происходит поворот вектора весов в сторону входного сигнала. Постепенное уменьшение скорости поворота позволяет произвести статистическое усреднение входных векторов, на которые реагирует данный нейрон.

Заключение

Итогом данного исследования стало модульное приложение, осуществляющее голосовую идентификацию пользователя, использующее модернизированный алгоритм вычисления нейронов в слое Кохонена. Программа состоит из трех основных частей. Первая выполняет добавление пользователей, вторая выполняет идентификацию и третья хранение голосовых записей.

Как показало исследование, полученный алгоритм позволяет значительно ускорить работу программы голосовой идентификации. Данная модернизация позволяет использовать программу на предприятиях с большим потоком пользователей.

Так же программный комплекс очень гибок и имеет большое пространство для дальнейшего усовершенствования и добавления новых функций, что делает его не только выгодным программным продуктом, но и перспективным проектом для развития и получения прибыли.

Библиографический список

[Bosi M., 2003] Introduction to digital audio coding and standards / M. Bosi, R.E. Goldberg - Springer Science+Business, Media USA. - 2003. - 434 p.

[You Y., 2010] AudioCoding: Theory and Applications / Y. You - NY: Springer, 2010 - 349 p.

[Загуменнов, А. П., 1999] Компьютерная обработка звука./ А. П. Загуменнов - М. : ДМК, 1999. - 384 с

VOICE USER IDENTIFICATION IN ACCESS CONTROL SYSTEMS

Menshakov P.A., Murashko I.A.

*Department of Information Technology
Gomel State Technical University named by P.O.*

*Suhoi
Gomel, Belarus*

pmenshakov@gmail.com

iamurashko@tut.by

At the moment, the primary means of access control checkpoints are equipped with a variety of means of access control. But most of the access controls are expensive. And most of the costs fall on the allocation of personal identification means for each user. The solution to this problem is to voice recognition. The use of biometrics eliminates the chips and access cards, identification means to eliminate loss and theft. And the use of the voice will eliminate the expensive equipment to read the data.

Introduction

Currently, voice recognition, biometrics as a whole, already widespread. The simplest example - fingerprint scanners installed on almost every laptop. Biometrics recognition system involves people on one or more physical or behavioral traits. In the field of information technology, biometric data is used as a form of identity management and access control access. Also, biometric analysis is used to identify people who are under the supervision of. Just provide biometrics and behavioral analysis of the object. These include walking, gestures, etc. Authorization process, the use of biometrics, is quite simple. Using an apparatus for varying the characteristics of an identified current data scanned and compared with previous data. Biometric systems have a number of important advantages. Biometrics uses the properties of the human body and its behavior, which make them unique. Unlike paper identifiers from a password or personal identification number (PIN), biometric characteristics can not be subjected to theft, can not be lost or forgotten. Quite a long time fingerprints are used to identify criminals and prevent theft or fraud. Some people are able to mimic the voice,

but it requires special skills that are not often met in everyday life.

Main Part

To implement voice recognition is necessary to make a specific course of action. With a microphone turns voice recording identified and sent to the computer. The optimal reception is WAV file, since handling ease. The resulting voice recording should be divided into frames.

The next step is to eliminate the undesirable effects and noises. It is necessary to record obtained at different time correspond to each other, regardless of external factors. There are many ways in which to reduce the effects of noise.

To date, the most successful are the voice recognition system, using the knowledge of the hearing aid device. They are based on the fact that the ear interprets sounds not linearly but in a logarithmic scale. In view of these features is necessary to bring the frequency response for each frame of mels.

This is the last step required for the subsequent conversion to vector features, which, compared to the base of voice recordings. The vector will comprise mel-spectral coefficients.

The resulting feature vector is added to the database for later comparison. But a more accurate alternative is to use multiple entries of the same voice. A predetermined number of voice samples may be used to train the neural network. We used learning without a teacher, because it is much more plausible model of learning in the biological system. Kohonen developed and many others, it does not need to output the target vector and therefore.

Conclusion

The result of this study is a modular application that voice recognition by using an upgraded algorithm for computing the neurons in the Kohonen layer. The program consists of three main parts. The first performs the addition of users, the second and third authenticates voice recordings.

The study showed that the resulting algorithm can significantly speed up the program of voice identification. This upgrade allows you to use the program at the enterprises with the influx of users.

The same software system is very flexible and has plenty of room for further improvements and adding new features, making it not only profitable software product, but promising project for development and profit.



УДК 681.3

ОНТОЛОГИЧЕСКИЙ ПОДХОД К ФОРМИРОВАНИЮ КОНТЕКСТНЫХ ЗАПРОСОВ В ЭЛЕКТРОННОМ АРХИВЕ ТЕХНИЧЕСКИХ ДОКУМЕНТОВ

Наместников А.М.

Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация
nam@ulstu.ru

В работе приводятся модели и методы выполнения контекстно-ориентированных запросов к электронным архивам технических документов крупной проектной организации. Онтология рассматривается в контексте решения задачи уточнения пользовательских запросов с целью повышения точности и полноты информационного поиска. В статье содержатся результаты вычислительных экспериментов на базе современного предметно-ориентированного электронного архива.

Ключевые слова: онтология; контекстно-ориентированный запрос; электронный архив.

Введение

Основными задачами электронного архива крупной проектной организации является обеспечение коллективной работы проектно-конструкторских отделов над проектом, добавление, хранение и поиск технических документов (ТД). Поиск часто осуществляется по заранее определенным реквизитам документов и по ключевым словам [Маннинг и др., 2011]. Однако данные модели поиска не имеют представления об информационных потребностях пользователя и, тем самым, всегда присутствует вероятность того, что документы, которые были отобраны, не позволят сократить информационную неопределенность проектировщика. Современные системы информационной поддержки используют механизмы интеллектуального поиска. Интеллектуальный поиск – это ключевая тенденция в современном информационном поиске, которая предполагает способность поисковой системы к самоорганизации, осуществление независимого общения с пользователем, эффективный поиск текстовых документов, реагирующий на изменения информационной потребности пользователя. Знания могут быть представлены в виде онтологии предметной области [Добров Б.В. и др., 2006; Гаврилова Т.А. и др., 2000].

В данной статье представлена модель формирования контекстно-ориентированных поисковых запросов, основанная на использовании знаний о предпочтениях проектировщиков в процессе поиска ТД. Фактически, речь идет об

использовании индивидуального профиля проектировщика, который может применяться в задачах онтологически-ориентированного информационного поиска текстовых документов и позволит максимально полно удовлетворить информационную потребность пользователя.

1. Модель профиля проектировщика

Формализация *профиля проектировщика* осуществляется на основе предположения о том, что имеется возможность фиксировать результаты проектных запросов к электронному архиву в виде множества документов, удовлетворяющих информационной потребности, и множества документов, которые текущей информационной потребности не удовлетворяют.

Каждой информационной потребности In_j^i будем ставить в соответствие пару классов понятий онтологии предметной области $C^+ = \{c_1^+, c_2^+, \dots, c_n^+\}$, $C^- = \{c_1^-, c_2^-, \dots, c_m^-\}$, определяющие положительные и отрицательные подмножества понятий онтологии, соответственно (рисунок 1).

В процессе выполнения конкретным проектировщиком информационных запросов к электронному архиву определяется набор ТД, которые соответствуют его информационной потребности (D^+) и ТД, не соответствующие ей (D^-), с учетом текущей стадии (этапа) проектирования.

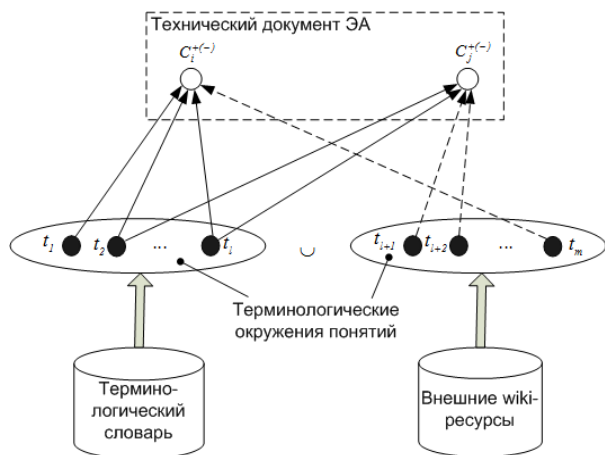


Рисунок 1 – Источники формирования ресурсов профиля проектировщика

Для каждого документа определяется его концептуальное представление. Другими словами, производится нечеткое онтологическое индексирование. Запишем нечеткое соответствие между множеством $C^{+(-)}$ и множеством $T^D = T_{in}^D \cup T_{ext}^D$, как $\tilde{F}_{CT} = (C^{+(-)}, T^D, \tilde{F}_{CT})$, где \tilde{F}_{CT} – нечеткое множество в $C^{+(-)} \times T^D$. Определим нечеткое соответствие \tilde{F}_{CT} в виде ориентированного двудольного графа с множеством вершин $C^{+(-)} \cup T^D$, каждой дуге $\langle c_i^+, t_j \rangle$ ($\langle c_i^-, t_j \rangle$) которого приписываем значение функции принадлежности $\mu_{F_{CT}} \langle c_i^+, t_j \rangle$ ($\mu_{F_{CT}} \langle c_i^-, t_j \rangle$). Указанное значение функции принадлежности вычисляется на основе нормализованной частоты встречаемости термина в терминологическом окружении понятия.

Образ множества T^D , при соответствии \tilde{F}_{CT} , фактически представляет собой нечеткое множество, элементами которого являются концепты с соответствующими степенями выраженности:

$$\tilde{F}_{CT}(T^D) = \{\mu_{F_{CT}}(c^{+(-)})/c^{+(-)}\}, \quad (1)$$

где $\mu_{F_{CT}}(c^{+(-)}) = \bigvee_{t \in T^D} \mu_{F_{CT}} \langle c_i^+, t_j \rangle$.

В положительные и отрицательные подмножества понятий онтологии предметной области включаются такие понятия из онтологических представлений, степень выраженности которых наибольшая.

Формально профиль проектировщика будем представлять в виде кортежа [Наместников и др., 2014]:

$$\begin{aligned} Ex_j^i &= \langle In_j^i, C^+, C^- \rangle, c^{+(-)} = \mu_{F_{CT}}(c^{+(-)}) = \\ &= \max_{c \in D} \left(\bigvee_{t \in T^D} \mu_{F_{CT}} \langle c_i^+, t_j \rangle \right), c^{+(-)} \in C^{+(-)}, \end{aligned}$$

где i – индекс проектировщика, j – индекс стадии жизненного цикла проектирования автоматизированной системы (АС).

2. Формирование онтологического контекста

Следовательно, контекст проекта перепишем как граф вида [Наместников и др., 2012]:

$$G^{PT} = \langle C^{PT}, R^{PT} \rangle, \quad (2)$$

где C^{PT} – множество вершин-понятий проекта, R^{PT} – множество дуг, соединяющих вершины-понятия.

Множество понятий проекта определяется как результат функции онтологического индексирования технического задания (Tz) на реализуемый проект ($F_{ol}(Tz)$) и функции онтологического доопределения множества C^{Tz} , как результата $F_{ol}(Tz)$ с применением wiki-ресурсов сети Internet ($F_{cAdd}(C^{Tz})$). Алгоритм формирования онтологического контекста проекта АС (2) представим в виде следующих шагов [Наместников и др., 2012]:

Шаг 1. Загрузка файла технического задания (Tz).

Шаг 2. Онтологическое индексирование технического задания:

$$C^{Tz} = F_{ol}(Tz). \quad (3)$$

Шаг 3. Доопределение множества C^{Tz} .

На данном шаге выполняется анализ wiki-ресурса сети Internet и определяется множество дополнительных понятий, имеющих связи с понятиями множества C^{Tz} . Идентификация связей между понятиями определяется на основе существующих гиперссылок на соответствующие страницы сети, содержащие текстовые описания понятий.

Шаг 4. Загрузка терминологического словаря Dic , который формируется на основе технической документации электронного архива, в том числе из набора основных терминов и понятий из ГОСТ, принятых к исполнению в проектной организации.

Шаг 5. Сравнение терминологических окружений $T_{sur}(\hat{C}^{Tz})$ понятий $\hat{C}^{Tz} = F_{cAdd}(C^{Tz})$ с терминами из Dic . Если $\forall \hat{w} \in T_{sur}(\hat{C}^{Tz})$ выполняется условие $\hat{w} \notin Dic$, то необходимо удалить понятие $\hat{c} \in \hat{C}^{Tz}$.

Шаг 6. Проверка очередного $\hat{c} \in \hat{C}^{Tz}$. Если сравнение терминологических окружений со словарем выполнено не для всех элементов множества \hat{C}^{Tz} , тогда выполняется переход к шагу 5.

Шаг 7. Определение множества дуг R^{PT} на основе анализа гиперссылок страниц wiki-ресурса.

Шаг 8. Сохранение графа G^{PT} .

Рассмотрим детально процесс формирования концептуальной сети, извлекаемой из «Википедии» – свободной общедоступной мультязычной универсальной интернет-энциклопедии, реализованной на принципах wiki. Концепты в данной библиотеке представлены в виде HTML-страниц. Для связи между страницами используются гиперссылки, которые символизируют семантическую связь между понятиями. Опираясь на систему гиперссылок, существует возможность в автоматическом режиме переходить от одной страницы к другой, извлекая знания о понятиях предметной области.

Рассмотрим определение множества дуг R^{PT} (шаг 7) на основе анализа гиперссылок wiki-ресурсов:

1. Извлекаются понятия из онтологии проекта.

2. Выполняется извлечение понятий из wiki. В основе данного процесса лежит модифицированный алгоритм волновой трассировки (применяемый при трассировке печатных плат радиоэлектронных устройств). Данный процесс состоит из ряда последовательных шагов:

2.1. На вход данного алгоритма поступают множества понятий полученных на этапе 1.

2.2. Выполняется поиск страниц в wiki, в которых концепты являются заголовками.

2.3. Страницы, полученные на этапе (2.2) анализируются с целью нахождения тех концептов, для которых одновременно выполняются условия:

- существует страница, которая описывает концепт;
- анализируемая страница содержит гиперссылки на страницу найденного концепта;
- страница концепта содержит обратную гиперссылку на анализируемую страницу.

2.4. Обнаруженные концепты добавляются в предварительное ядро онтологии.

2.5. Проверяется условие существования маршрута между всеми первичными концептами, которые получены на этапе (1).

2.6. Если условие пункта 2.5 выполняется, то это означает окончание модифицированного алгоритма волновой трассировки, если не выполняется, то пункты 2.2-2.5 выполняются снова для концептов, извлеченных на этапе 2.3.

Таким образом, на выходе второго этапа получаем множество понятий, между которыми существуют неидентифицированные семантические отношения. Однако может оказаться так, что это множество содержит понятия, которые выходят за границы исследуемой предметной области.

3. Полученные на втором этапе концепты

приводятся в нормальную форму, т.е. с помощью алгоритма стемминга выделяются словарные основы концептов.

4. Для каждого извлеченного концепта формируется терминологическое окружение. Терминологическое окружение создается путем извлечения терминов из страницы концепта wiki-ресурса и вычисления частоты встречаемости термина на данной странице по следующей формуле:

$$ntf_{t,d} = a + (1-a) \frac{tf_{t,d}}{tf_{\max}(d)}, \quad (4)$$

где $tf_{\max}(d) = \max_{\tau \in d} tf_{\tau,d}$ – максимальная величина tf в документе d , a – сглаживающий коэффициент, принимающий значение между нулем и единицей (экспериментально устанавливаются равной 0,4). Роль данного параметра состоит в уменьшении вклада второго члена. Нормировка частоты термина по максимуму предназначена для того, чтобы избежать следующей аномалии [Маннинг и др., 2011]: в более длинных документах наблюдаются более высокие частоты терминов, так как в более длинных документах чаще содержатся повторяющиеся слова.

3. Классификация запросов на основе байесовской модели

Рассмотрим алгоритм формирования проектных запросов для улучшения показателей точности и полноты запросов к электронным архивам технических документов проектной организации.

Пусть множество $\hat{w} = \{\hat{w}_1, \hat{w}_2, \dots, \hat{w}_n\}$ – есть множество ключевых слов проектного запроса к электронному архиву. Используя функцию $F_{DC} : \{T^D\} \rightarrow C$ для построения терминологических окружений понятий онтологии предметной области, данное множество отображается в нечеткое множество:

$$\tilde{I}_q = \{\mu_1 / c_1, \mu_2 / c_2, \dots, \mu_m / c_m\}, \quad (5)$$

как результат онтологического преобразования терминов в набор степеней выраженности понятий онтологии.

Для представления модели формирования проектных запросов на концептуальном уровне будем использовать наивный байесовский классификатор. Вероятность того, что понятие запроса c принадлежит классу $k \in \{C^+, C^-\}$, будем определять по формуле Байеса:

$$P(k | c) = \frac{P(c | k) \cdot P(k)}{P(c)}, \quad (6)$$

где $P(c | k)$ – вероятность встретить понятие c среди всех понятий класса k ; $P(k)$ – безусловная

вероятность понятия класса k в онтологии предметной области; $P(c)$ – безусловная вероятность понятия C в онтологии предметной области.

Наиболее вероятный класс для понятия запроса определяется, используя оценку апостериорного максимума:

$$k_{map} = \arg \max_{k \in K} \frac{P(c | k) \cdot P(k)}{P(c)}. \quad (7)$$

Поскольку $P(c) = const$ в рамках одной онтологии и учитывая, что

$$P(c | k) \approx P(w_1 | k) \cdot P(w_2 | k) \cdot \dots \cdot P(w_n | k) \\ = \prod_{i=1}^n P(w_i | k), \quad (8)$$

получаем:

$$k_{map} = \arg \max_{k \in K} [P(k) \cdot \prod_{i=1}^n P(w_i | k)]. \quad (9)$$

Для больших документов количество множителей $P(w_i | k)$ в выражении (9) может быть большим, а, следовательно, возникает проблема исчезновения порядка вследствие перемножения большого количества малых чисел. Перепишем выражение (9) с учетом свойств логарифма:

$$k_{map} = \arg \max_{k \in K} [\log P(k) \cdot \sum_{i=1}^n \log P(w_i | k)] \quad (10)$$

Оценка вероятностей $P(k)$ и $P(w_i | k)$ выполняется на основе обучающей выборки, сформированной для каждой информационной потребности In_j^i . Вероятность класса будем записывать как:

$$P(k) = \frac{D_k}{D}, \quad (11)$$

где D_k – количество документов, принадлежащих классу k и определяемое на основе результатов выполнения запросов проектировщика к электронному архиву; D – общее количество документов в обучающей выборке.

Величина $P(w_i | k)$ определяет вероятность встретить термин w_i среди терминов документов, принадлежащих классу k . Значение данной величины будем определять с учетом того, что термин из окружения понятия, включенного в проектный запрос, может отсутствовать в документах анализируемого класса. Применяя метод аддитивного сглаживания (сглаживания Лапласа), получаем:

$$P(w_i | k) = \frac{f_{ik} + 1}{\sum_{i' \in V} (f_{ik'} + 1)}, \quad (12)$$

где f_{ik} – частота встречаемости i -го термина в документах класса k ; V – терминологический словарь проектной организации (список всех уникальных терминов).

В результате классификации понятий запроса (5) понятия, принадлежащие к положительному классу понятий информационной потребности проектировщика, остаются в составе запроса. Понятия, которые классифицированы как отрицательные, исключаются из исходного запроса.

4. Способ редукции понятий контекстно-ориентированного запроса

В случае достаточно большой онтологии и неявной принадлежности запроса конкретному фрагменту предметной области проектирования мощность множества C^q (количество понятий, включенных в нечеткое представление проектного запроса) может быть большой. Поэтому возникает необходимость в процедуре редукции запроса (5).

Способ редукции понятий проектного запроса основывается на разделении исходного графа $G^q = \langle C^q, R^q \rangle$ на несколько подграфов, каждый из которых содержит только вершины, соединенные дугами одной семантической категории. В данной работе используются две семантические категории: «обобщение» («isA») и «часть-целое» («part_of»). На рисунке 2 представлен иллюстративный пример графа проектного запроса, в котором присутствуют вершины C_1, \dots, C_{14} – понятия онтологии предметной области и два типа дуг: дуги «isA» (штриховая стрелка) и дуги «part_of» (сплошная стрелка).

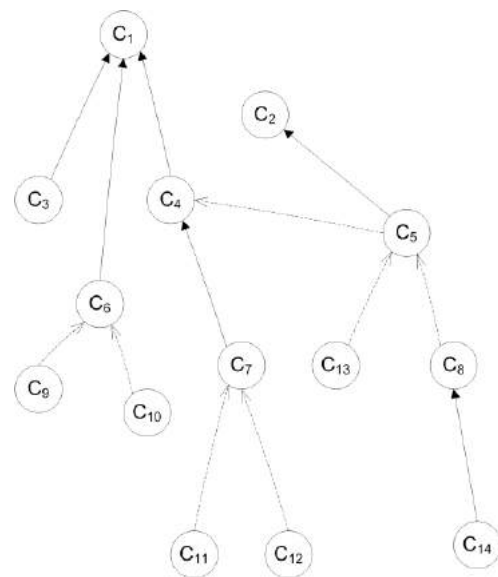


Рисунок 2 – Иллюстративное представление проектного запроса в виде графа

Редуцирование понятий графа проектного запроса предполагает выполнения ряда следующих шагов.

Шаг 1. Разбиение графа проектного запроса на несколько подграфов с учетом семантических категорий («isA» или «part_of»). На рисунке 3 показан подграф проектного запроса $G_{isA}^q = \langle C_{isA}^q, R_{isA}^q \rangle$, понятия которого связаны между собой отношениями типа «isA». Соответственно, на рисунке 4 представлен подграф $G_p^q = \langle C_p^q, R_p^q \rangle$ с отношениями «part_of».

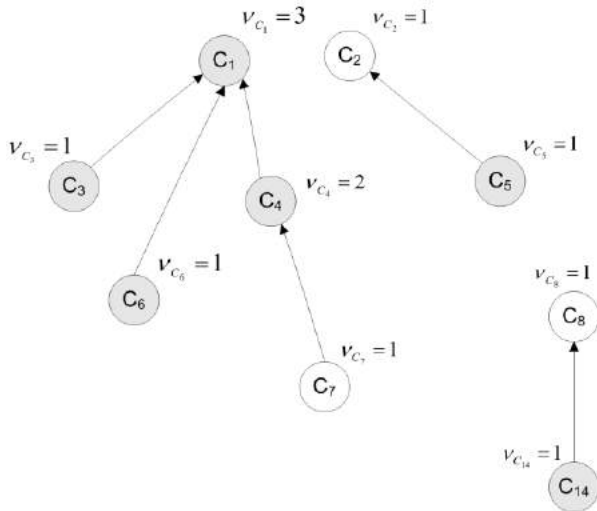


Рисунок 3 – Подграф графа проектного запроса с типом отношений «isA»

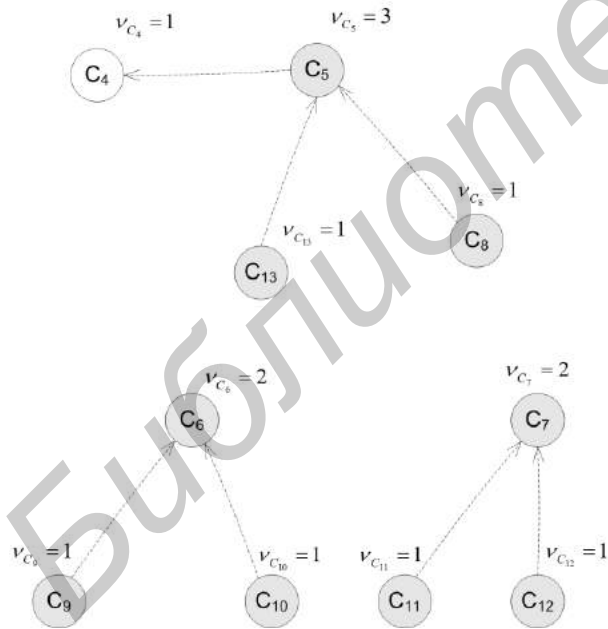


Рисунок 4 – Подграф графа проектного запроса с типом отношений «part_of»

Шаг 2. Для каждой вершины подграфов определяется ее степень V_{C_j} (количество входящих и исходящих дуг). Как видно из рисунков наибольшие степени имеют понятие C_1 в подграфе

«isA» и понятия C_5 , C_6 и C_7 в подграфе «part_of».

Шаг 3. В редуцированное множество понятий отдельно взятого подграфа проектного запроса включаются понятия согласно следующим правилам.

Правило 1: Если в подграфе существует вершина с максимальной степенью, то в результирующее множество соответствующего подграфа включается данная вершина и связанные с ней вершины, дуги от которых направлены к вершине с максимальной степенью.

Правило 2: Если подграф содержит две вершины, соединенные дугой, то в результирующее множество включается вершина с исходящей дугой.

Правило 3: Если подграф состоит из одной изолированной вершины, то данная вершина включается в результирующее множество понятий.

На рисунках 3 и 4 темным фоном отмечены те вершины подграфов, которые включены в результирующие множества понятий C_{isA}^{q+} и C_p^{q+} с использованием вышеприведенных правил.

Шаг 4. В редуцированное множество понятий проектного запроса включаются понятия, которые включены как во множество C_{isA}^{q+} , так и во множество C_p^{q+} : $C^{q+} = C_{isA}^{q+} \cap C_p^{q+}$.

5. Результаты вычислительных экспериментов

Вычислительные эксперименты проводились в центре обработки данных под управлением операционной системы Red Hat Enterprise Linux с виртуализацией Red Hat Enterprise Virtualization со следующими характеристиками: 8-ми ядерный процессор Intel Xeon MP E7-2830 Westmere-EX, размер оперативной памяти – 256 Гб.

В ходе эксперимента была построена прикладная онтология, которая включает в себя около 300 концептов и 30000 уникальных терминов.

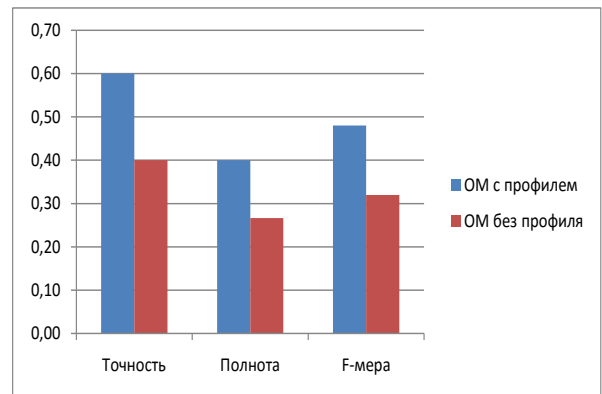


Рисунок 5 -- Сравнение результатов экспериментов

Разработанная интеллектуальная система

контекстно-ориентированного поиска документов позволила увеличить точность поиска технических документов в электронном архиве до 30%.

Заключение

В данной статье рассмотрен новый подход к анализу контекстно-ориентированных проектных запросов к электронному архиву технических документов проектной организации.

Результаты вычислительных экспериментов на реальном множестве документов из реализованных проектов по созданию автоматизированных систем демонстрируют, что формализация профилей проектировщиков и описание предметной области в виде онтологии позволяет улучшить качество человеко-машинного взаимодействия с предметно-ориентированными электронными архивами.

Исследование выполнено в рамках государственного задания №2014/232 на выполнение государственных работ в сфере научной деятельности Минобрнауки России по проекту «Разработка нового подхода к интеллектуальному анализу слабоструктурированных информационных ресурсов».

Библиографический список

[Маннинг и др., 2011] Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. М: Вильямс, 2011.

[Добров и др., 2006] Добров Б.В., Лукашевич Н.В., Лингвистическая онтология по естественным наукам и технологиям: основные принципы разработки и текущее состояние // Десятая национальная конференция по искусственному интеллекту с международным участием (Обнинск, 25-28 сентября 2006 г.) – М.: Физматлит, 2006.

[Гаврилова и др., 2000] Гаврилова Т.А., Хорошевский В.Ф., Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.

[Наместников и др., 2014] Наместников А.М., Субхангулов Р.А. Формирование информационных запросов к электронному архиву на основе концептуального индекса // Радиотехника №7 – 2014. – С. 126-129.

[Наместников и др., 2012] Наместников А.М., Субхангулов Р.А. Разработка инструмента инженерии онтологии в интеллектуальном проектном репозитории // Автоматизация процессов управления №2 (28) – 2012. – С.38 – 43.

[Наместников и др., 2013] Наместников А.М., Субхангулов Р.А., Филиппов А.А. Применение нечетких моделей в задачах кластеризации и информационного поиска текстовых проектных документов // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VII-й Международной научно-практической конференции (Коломна, 20-22 мая 2013 г.). В 3-х томах. Т3. – М.: Физматлит, 2013. – С. 1278–1289.

ONTOLOGICAL APPROACH TO FORMATION OF CONTEXTUAL QUERIES IN ELECTRONIC ARCHIVE OF TECHNICAL DOCUMENTATION

Namestnikov A.M.

*Ulyanovsk State Technical University, Ulyanovsk,
Russian Federation*

nam@ulstu.ru

In work models and methods of performance of the contextual-oriented queries are given to electronic archives of technical documentation in a large design organization. The ontology is considered in the context of the solution of a problem of specification for user queries for the purpose of increase of accuracy and completeness of information retrieval. The article contains results of computing experiments on the basis of modern subject-oriented electronic archive.

Introduction

The main objectives of electronic archive of a large design organization is ensuring collective work of design departments on the project, addition, storage and search of technical documentation. The model of formation of the contextual focused search queries based on uses of knowledge of designer preferences in the course of search of the document is presented in this article. Actually, it is about use of an individual cross-section of the designer which can be applied in problems of the ontology focused search of text documents and will allow to satisfy of user information need the most fully.

Main Part

Formalization of a designer cross-section is carried out on the basis of the assumption that there is an opportunity to fix results of design queries to electronic archive in the form of a set of documents, the satisfying information requirement, and a set of documents which don't satisfy the current information requirement.

For each document its conceptual representation is defined. In other words, indistinct ontological indexing is made.

For representation of design queries formation model at the conceptual level the naive Bayesian qualifier is used.

In case of a big ontology and implicit accessory of inquiry to a concrete fragment of domain area the set power (amount of the concepts included in indistinct representation of design query) can be big. Therefore there is a need for procedure of a reduction of query.

The mode of reduction of concepts of design inquiry is based on division of the initial count into some subgraphs, each of which contains only the tops connected by arches of one semantic category. In this work two semantic categories are used: "generalization" and "part - whole".

Conclusion

Results of computing experiments on a real set of documents from the realized projects on the automated systems creation show that formalization of designer cross-sections and the description of subject domain in the form of ontology allows to improve quality of human-machine interaction with subject-oriented electronic archives.



УДК 811.93

СЕМАНТИЧЕСКИЙ АНАЛИЗ ТЕКСТОВОЙ ЧАСТИ РУКОВОДСТВА ПОЛЬЗОВАТЕЛЯ ПО СБОРКЕ ДВИГАТЕЛЯ С ПОМОЩЬЮ ОНТОРЕДАКТОРА FLUENT EDITOR

Орлова А.А., Боргест Н.М.

*Самарский государственный аэрокосмический университет им. академика С.П. Королева
(национальный исследовательский университет), г. Самара, Россия*

a.aorlova@yandex.ru

borgest@yandex.ru

В работе рассматривается семантический анализ текстовой части руководства по сборке двигателя 2,0 DOHC-16V, подготовленной на естественном английском языке, и преобразование его в онтологию на контролируемом английском языке (Controlled Natural Languages, CNL). Изучена возможность использования редактора Fluent Editor, в котором онтологии создаются с помощью CNL, как для проверки полноты руководств пользователя, так и для их подготовки. Онтология части руководства пользователя по сборке двигателя представлена тезаурусом используемых профессиональных терминов и понятий рассматриваемой предметной области.

Ключевые слова: Controlled Natural Language, онтология, Fluent Editor, семантический анализ.

Введение

В последнее время все чаще для решения проблем неоднозначности используют естественные контролируемые языки. Создаются онтологические редакторы, использующие для построения онтологий CNL (Controlled Natural Languages), например, польский редактор Fluent Editor. Контролируемый язык – упрощенная версия естественного языка, следовательно, CNL можно использовать в качестве языка для описания технических текстов, так как они имеют более строгие структуру и систему понятий. В свою очередь, редакторы, в основу которых заложен CNL можно использовать в качестве инструмента для подготовки технических текстов (в том числе руководств по сборке).

1. Fluent Editor

1.1. Controlled Natural Language

Контролируемый язык – упрощенная версия естественного языка, созданная путем ограничения грамматики, терминологии и речевых оборотов, чтобы снизить или искоренить многозначность и сложность естественного языка [O'Brien, 2003].

Fluent Editor – редактор от польской компании

Cognitum для всеобъемлющего редактирования сложных онтологий, при создании которых используется естественный контролируемый язык. В Fluent Editor контролируемым естественным языком является английский. CNL может служить для представления знаний и в качестве языка интерфейса систем знаний высокого уровня.

2. Семантический анализ

При написании онтологии на CNL необходимо придерживаться правила контролируемого языка, а также использовать тезаурус описываемой предметной области для искоренения многозначности профессиональных терминов. Поэтому сначала рассмотрен вопрос о выявлении экземпляров, отношений и классов руководства пользователя для преобразования текста из естественного английского языка в онтологию на CNL. Приведена технология преобразования тезауруса используемых профессиональных терминов.

2.1. Алгоритм выявления экземпляров и отношений между ними в руководстве пользователя по сборке двигателя

В первоначальном варианте онтологический редактор не распознает текст, так как он не соответствует грамматике и синтаксису CNL. Окно,

визуализирующее иерархию отношения классов и экземпляров онтологии, – окно Дерева Таксономии (от англ. Taxonomy Tree) остается пустым.

Следовательно, необходимо преобразовать текст на естественном английском в CNL. (рисунок 1).

Алгоритм преобразования описания руководства по сборке двигателя из естественного английского языка в CNL состоит из следующих действий:

1. Создание зависимости экземпляр-экземпляр. Экземпляр - частный вид класса, не имеющий подвидов, но имеющий возможность состоять в отношениях с другими экземплярами. Экземпляры на CNL всегда записываются с заглавной буквы, если имя экземпляра состоит из нескольких слов, то они разделяются дефисами и тогда каждое слово в имени экземпляра записывается с заглавной буквы.

2. Отношения необходимо задать зависимостью «Экземпляр отношение Экземпляр», например, «Step-1 cleaning Engine.». Экземпляр «Step-1» вводится по причине того, что как правило все предложения в руководствах пользователя, употребляются в указательной форме «Сказуемое дополнение», а для задания предложения на CNL необходима форма «Подлежащее сказуемое

дополнение», в данном примере экземпляр «Step-1» является подлежащим. Отношение «cleaning» осталось от предыдущей формы предложения, но так как необходимо в предложении ответить на вопрос «Очистить что?», дополнительно введен экземпляр «Engine».

3. Если встречаются предложения с однородными членами «Measure the diameters of the main and big-end bearing journals using a proprietary micrometer screw.» необходимо провести преобразование типа «Step-2 measure Diameters-Of-The-Main-Bearing and measure Big-End-Bearing.», т.е. описать, что в шаге №2 проводится две подобных операции, союзом «and» обозначается принадлежность двух отношений и экземпляров к одному экземпляру-подлежащему. Произведя пошаговую доработку обычного текста в CNL, получаем онтологию, представленную на рисунке 2.

В Дереве Таксономии в пункте «relation» отображены все, описанные отношения. Для лучшего восприятия и проверки написанного, визуализируем построенную онтологию с помощью встроенного в Fluent Editor инструмента графического представления онтологий – CNL-диаграммы, представленной на рисунке 3.

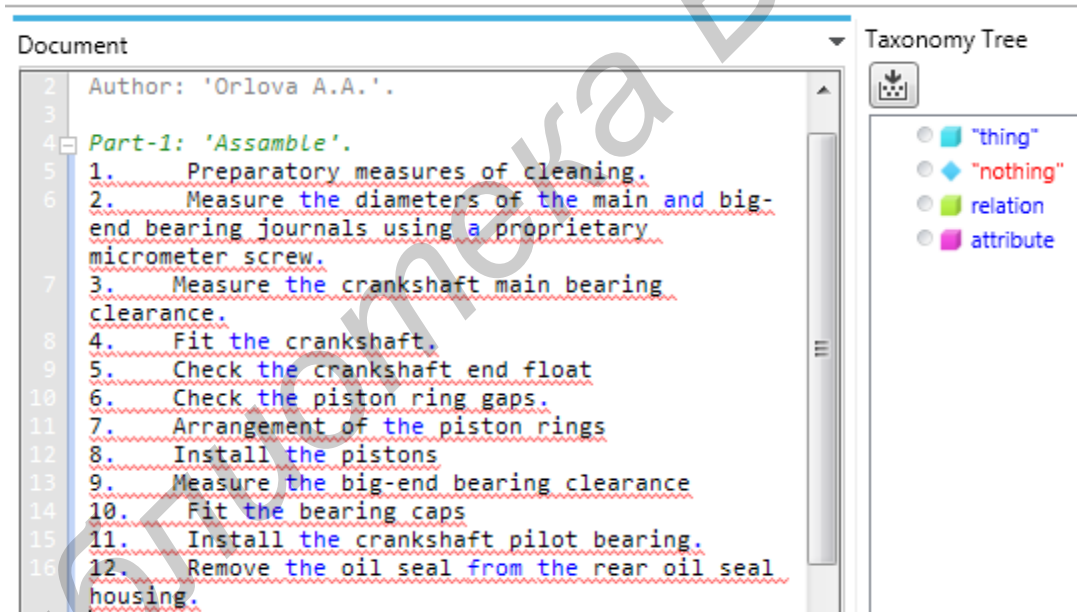


Рисунок 1 – Часть руководства пользователя по сборке двигателя на естественном английском языке

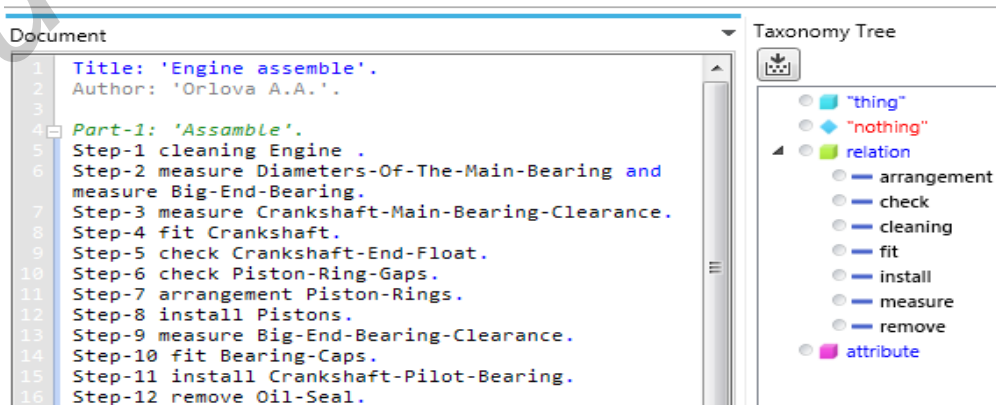


Рисунок 2 – Часть руководства пользователя по сборке двигателя, представленная на CNL, после 1,2 и 3 преобразования

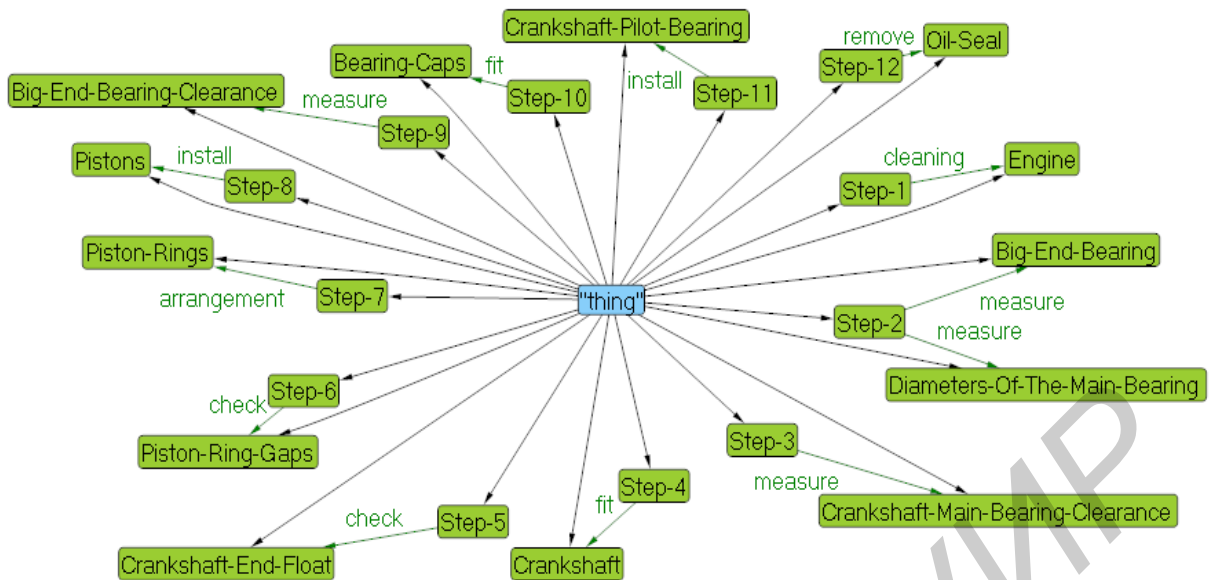


Рисунок 3 – Визуализация части руководства по сборке двигателя, представленная в виде CNL-диаграммы

2.2. Технология преобразования тезауруса к руководству пользователя по сборке двигателя в тезаурус на CNL

Продолжим алгоритм тем, что добавим в онтологию тезаурус профессиональных терминов для искоренения многозначности [Габдрахманов и др., 2012]. Основой многих прикладных онтологий является тезаурус, разработка которого позволяет упорядочить знания о предмете, «договориться» о понимании сути обозначений и понятий, определиться и разобраться с терминологией [Боргест, 2010]. При введении определений терминов в их начальном виде на естественном английском редактор не воспринимает текст, Дерево Таксономии остается пустым (рисунок 4).

4. Выявление ключевых слов в каждом определении, например, в определении понятия «engine» ключевой будет являться фраза «is a machine to convert form of energy», «crankshaft» - «is a mechanical part to perform of motion», «oil seals» -

«is a spaces for helping prevent lubricant escape». Данные ключевые фразы являются основой для написания на CNL, необходимо лишь воспользоваться правилами правописания имен экземпляров и классов. При задании ключевых фраз понятий как классов следует соблюдать несколько правил правописания CNL. Во-первых, пробелы в имени класса так же, как и в имени экземпляра, заменяются на дефисы. Во-вторых, имя класса всегда начинается с маленькой буквы, а если имя класса состоит из нескольких слов, то каждое из них начинается с маленькой буквы (рисунок 5). На рисунке 6 представлена CNL-диаграмма онтологии тезауруса руководства пользователя по сборке двигателя.

Целиком алгоритм последовательного выполнения семантического анализа руководства пользователя по сборке двигателя из текста на естественном английском языке в CNL представлен на рисунке 7 в виде блок-схемы.

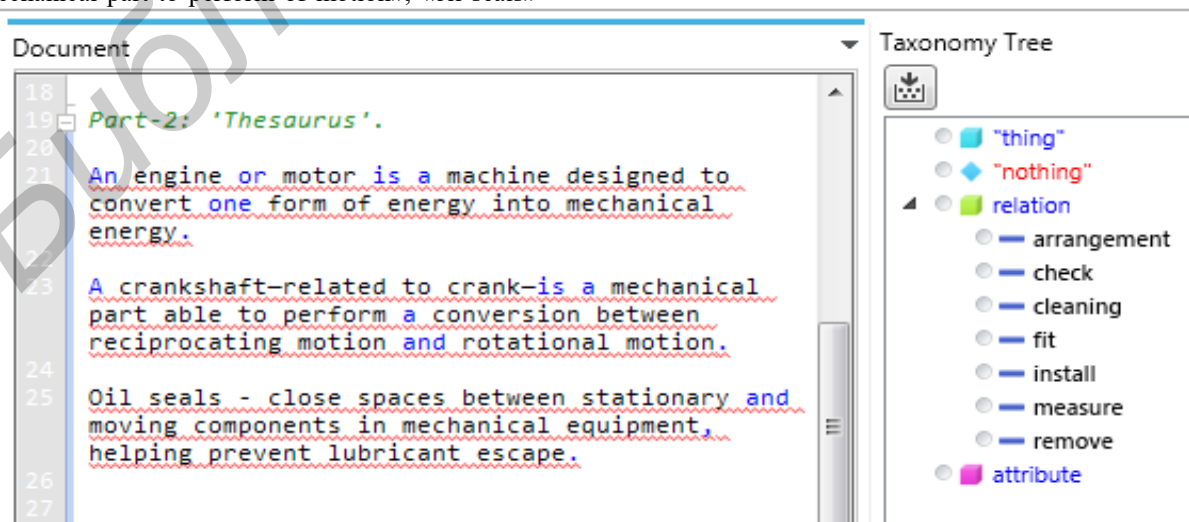


Рисунок 4 – Тезаурус к части руководства пользователя по сборке двигателя на естественном английском языке

Document

```

17
18
19 Part-2: 'Thesaurus'.
20
21 Engine is a machine-to-convert-form-of-energy.
22
23 Crankshaft is a mechanical-part-to-perform-of-motion.
24
25 Oil-Seal is a spaces-for-helping-prevent-lubricant-escape.
26
27
28
29
30
31
32
33

```

Document Document Diagram

Taxonomy Tree

- thing
 - machine-to-convert-form-of-energy
 - Engine
 - mechanical-part-to-perform-of-motion
 - Crankshaft
 - spaces-for-helping-prevent-lubricant-escape
 - Oil-Seal
 - "nothing"
 - relation
 - arrangement
 - check
 - cleaning
 - fit
 - install
 - measure
 - remove
 - attribute

Рисунок 5 – Тезаурус к части руководства пользователя по сборке двигателя , представленная на CNL, после четвертого преобразования

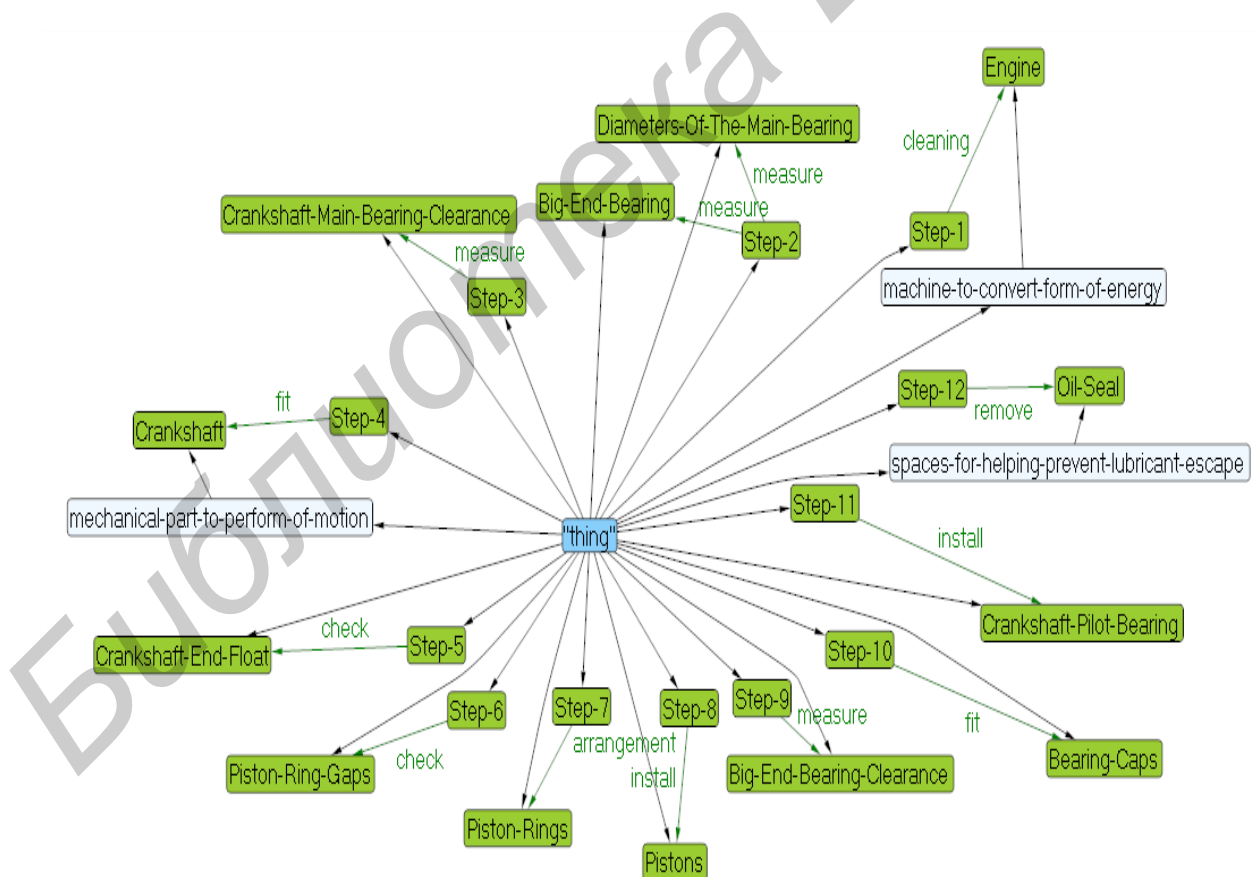


Рисунок 6 – Визуализация тезауруса к части руководства по сборке двигателя , представленная в виде CNL-диаграммы

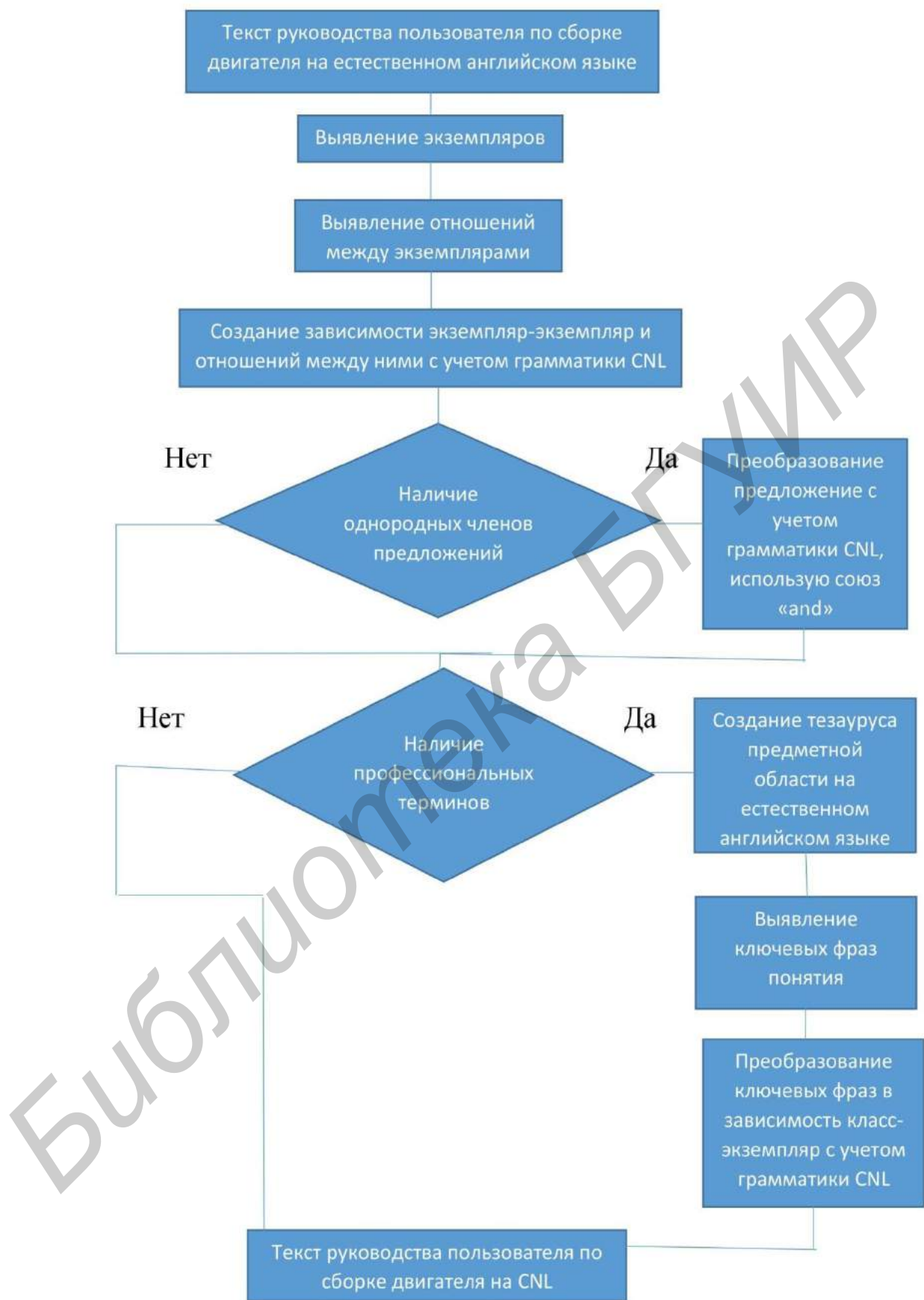


Рисунок 7 – Блок-схема алгоритма преобразования текста из естественного английского языка в CNL

Заклучение

Проведен семантический анализ технического текста на естественном английском для преобразования его в онтологию на CNL, путем некоторых упрощений, но с соблюдением профессиональных терминов. Соблюдение профессиональной терминологии является важным обстоятельством при создании онтологии, так как это язык межперсонального взаимодействия [Ландэ и др., 2014]. Функционал и возможности онтологического редактора Fluent Editor позволяют осуществлять семантический анализ текстов на естественном английском. Следует отметить, что инструменты визуализации редактора удобны для проверки полноты руководств пользователей и отношений в рассматриваемой предметной области.

Библиографический список

[2,0 DOHC-16V engine] 2,0 DOHC-16V engine. Engine Assembly – Dismantle and Assemble (engine removed). Scorpio '95. 09/97 [Электронный ресурс] - <http://www.fordscorpio.co.uk/manual/engines/dohc16v/DOHC16VDismant.pdf>. – с.30

[Боргест, 2010] Боргест Н.М. Онтология проектирования / уч. пособие Самарского государственного аэрокосмического университета, 2010. – с.91

[Габдрахманов и др., 2012] Габдрахманов И.Н., Кучуганов В.Н., Медведев Д.В. и др. Технология генерации онтологии базы данных с помощью онтологического толкового словаря, «Онтология проектирования» №4, 2012. – с.106

[Ландэ и др., 2014] Ландэ Д.В., Снарский А.А. Подход к созданию терминологических онтологий, «Онтология проектирования» №2, 2014. – с. 114

[O'Brien, 2003] Sharon O'Brien Controlling controlled English an analysis of several controlled language rule sets/ School of Applied Language and Intercultural Studies Dublin City University Dublin 9, Ireland. [Электронный ресурс] – <http://mt-archive.info/CLT-2003-Obrien.pdf> - с.10

SEMANTIC ANALYSIS OF TEXT USER MANUAL ENGINE ASSEMBLY WITH ONTOLOGY EDITOR FLUENT EDITOR

Orlova A.A, Borgest N.M.

*Samara State Aerospace University (National
Research University), Samara, Russia*

a.aorlova@yandex.ru

borgest@yandex.ru

In works deals semantic analysis of the engine assembly text in a natural English in the ontology of the Controlled Natural Languages (CNL). The possibility of using the editor Fluent Editor, where ontologies are created by CNL, to verify the completeness of the manuals, as well as for their preparation. Ontology of the manual assembly of the engine filled the thesaurus used by professional terms and concepts.

Introduction

In recent years more and more to address the ambiguity of using controlled natural languages. Created ontology editors, are used to build ontologies CNL, for example, the Polish editor Fluent Editor.

Controlled Natural Language is a simplified version of a natural language. Therefore, CNL can be used as a language for describing the technical texts, as they have a strict structure and system concepts. In turn, the Editors, which laid the basis for CNL used as a tool for the preparation of technical documents (including guidelines on the assembly).

Main Part

When writing ontology CNL must adhere to the rules of the controlled language, and use the thesaurus described domain to eliminate ambiguity professional terms. Therefore, we first address the issue of identifying instances of classes and relations manual for the text of the English language in a natural ontology to CNL. The technology used to simplify the thesaurus professional terms.

The original version of the ontological editor does not recognize the text, since it does not conform to the syntax and grammar of CNL. Window, visualizing the hierarchy relationship of classes and instances of the ontology - Taxonomy tree window (from the English. Taxonomy Tree) remains empty. Therefore, it is necessary to simplify the text in a natural English to CNL.

The conversion algorithm description manual assembly of the natural English in CNL consists of the following:

1. Create an instance and depending on the instance.
2. Creating a relationship between instances.
3. Conversion of homogeneous terms of the proposals.
4. Conversion of the thesaurus to the assembly guide on CNL.

Conclusion

Conducted technical semantic analysis of text in a natural English for its transformation into an ontology at CNL, by some simplifications, but in compliance with the professional terms. Compliance with professional terminology is an important factor in the creation of ontology, as is the language of interpersonal interaction. From the above it can be concluded about the possibilities of ontological editor Fluent Editor serve as a mechanism to semantic analysis texts in natural English. It should also be noted that visualization tools are convenient editor to verify the completeness and uniqueness of user guides and relationships in the subject area.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.522

СВЯЗЬ МОДУЛЬНОГО ИНТЕРФЕЙСА ИНТЕЛЛЕКТУАЛЬНОГО ПОМОЩНИКА ПРОЕКТАНТА С CAE СИСТЕМОЙ ANSYS

Власов С.А., Коровин М.Д.

*Самарский государственный аэрокосмический университет имени академика С.П. Королева
(национальный исследовательский университет), г. Самара, Россия*

savlasoglavv@mail.ru

maks.korovin@gmail.com

В работе рассмотрен процесс создания модульного интерфейса интеллектуального помощника проектировщика с CAE (Computer-Aided Engineering) системой ANSYS. Цель работы модульного интерфейса выполнить аэродинамический расчёт созданной параметрической 3D модели самолёта при использовании интеллектуального помощника проектировщика для получения комплексной оценки технико-экономических показателей проекта самолёта.

Ключевые слова: робот-проектант, интеллектуальный помощник, 3D модель, аэродинамический расчёт.

Введение

Проектирование новых технических систем относится к одному из наиболее сложных видов инженерной творческой деятельности. Главная особенность концептуального проектирования состоит в необходимости принятия множеств решений при недостаточной или, наоборот, избыточной информации. Робот-проектант служит помощником проектировщика и способен помочь принять решение, просчитать все возможные варианты на основе построенных математических моделей, осуществить параметрическое построение 3D модели самолёта. Связь с CAE (Computer Aided Engineering) системой ANSYS делает возможным проверку параметрической модели на все возможные случаи аэродинамических нагрузок. Аэродинамическая оценка определяет нагрузки, необходимые для прочностного анализа и расчёта конструкции, что позволяет в дальнейшем оценить массу планера, а также позволяет оценить значение потребных тяг двигателей на различных режимах полета и соответственно выбор силовой установки.

1. Робот-проектант

Системы современного автоматизированного проектирования постепенно переходят от существования в виде инструмента к службе «умный помощник» для проектировщика. Высокий уровень методик расчёта позволяет CAD (Computer

Aided Design) системам выполнять, постоянно увеличивающееся количество задач без необходимости участия самого проектировщика. Знание CAD – основное качество для современного инженера.

Внедрение автоматических систем — это сложный и дорогой процесс, иногда требующий значительных затрат времени, однако автоматизация часто возникающих типовых задач способна значительно повысить временную и экономическую эффективность работы проектировщика [Власов, 2015].

Проектирование самолёта представляет собой сложный междисциплинарный процесс, цель которого состоит в поиске рационального сочетания проектных параметров, которое наилучшим образом удовлетворяло бы поставленным задачам и выбранным критериям.

Робот-проектант – это компьютер с периферийными устройствами, инструментарию, включающие в себя языки описания, СУБД (системы управления базами данных), CAD системы, редакторы онтологий, и база знаний, как совокупность тезауруса, баз данных, правил и процедур, со сценариями проектирования [Боргест, 2012a].

Результатом работы робота-проектанта является модель изделия. Она состоит из нескольких взаимосвязанных частей – матрицы проекта с логикой расчёта (в текущем варианте выполненной

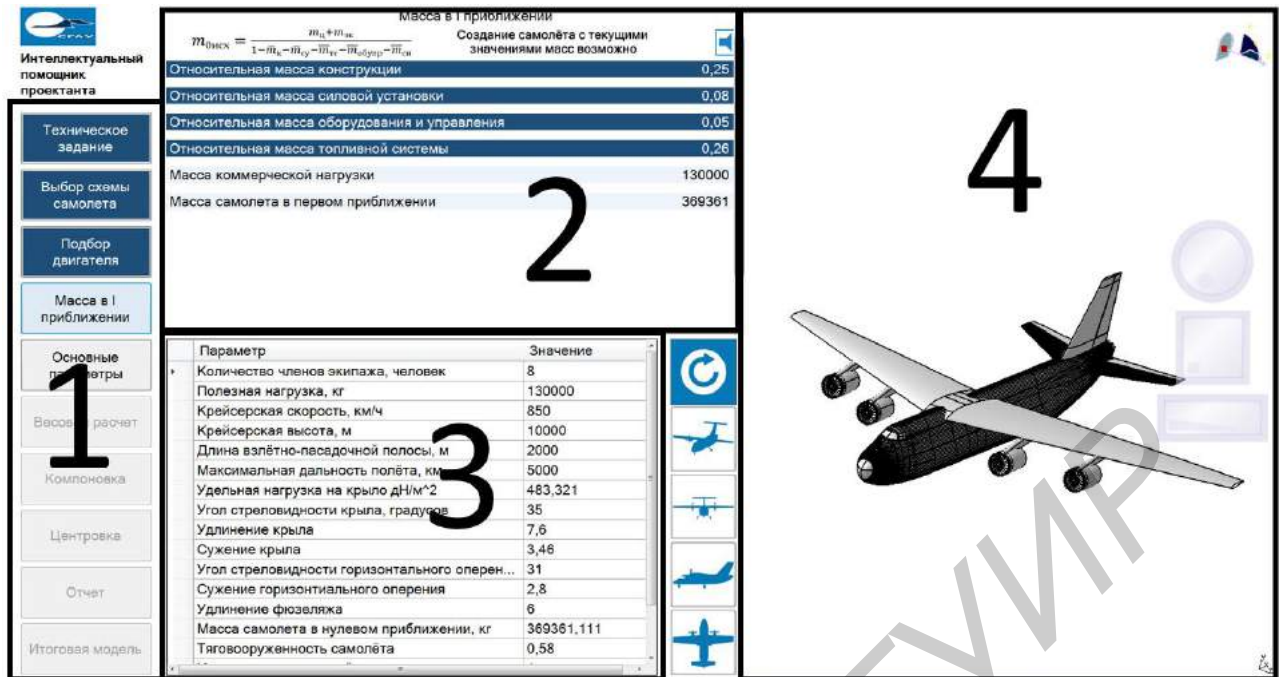


Рисунок 1 – Интерфейс робота-проектианта

на основе таблиц MS Excel), тезауруса предметной области (самолётостроение) и параметризованной трехмерной модели, которая автоматически меняет собственную конфигурацию в зависимости от данных, содержащихся в матрице проекта [Боргест, 2015b].

Задача аналитического проектирования самолета требует построения математической модели изделия, которая характеризуется совокупностью внешних параметров (требования технического задания на проект самолета: масса и другие параметры полезной нагрузки; потребная дальность полета; крейсерская скорость; класс аэродрома базирования и, соответственно, длина и ширина взлетно-посадочной полосы, её характеристики; категория и параметры системы автоматической посадки на аэродроме базирования и т. д.); и совокупностью внутренних параметров (параметры самолета, определяемые в процессе решения задачи: прежде всего, взлетная масса, удельная нагрузка на крыло, стартовая тяговооруженность, аэродинамические характеристики, размеры самолета и его агрегатов, состав и параметры систем и т. д.), а так же целевой функцией (критерием или критериями эффективности), позволяющей выбрать среди альтернативных проектов лучший, обеспечивающий экстремальное значение целевой функции. [Blessing, 2009]

Робот-проектиант имеет модульную схему – отдельные подзадачи (например, определение удлинения крыла или его массы) решаются отдельными подпрограммами, выполнение которых инициируется роботом по заданной логике [Боргест и др., 2015b].

Интерфейс предназначен для предварительного проектирования самолёта на основе мета-модели

самолёта (созданного в CAD-системе) и базы данных (в настоящее время на базе MS Excel и др.) Внешний вид интерфейса представлен на рисунке 1 [Боргест, 2015a].

Интерфейс робота-проектианта состоит из нескольких частей: таймлайна проекта (1), интерактивного экрана изменения проектных параметров (2), матрицы проекта и окна предварительного просмотра модели (4).

Таймлайн проекта – древовидная структура, отражающая ход выполнения расчета проекта. В силу того, что процесс проектирования самолета включает большое количество этапов и операций, не представляется возможным выводить все содержимое матрицы проекта на экран одновременно. Вместо этого разработан динамически изменяющийся таймлайн, отображающий важную на текущем этапе расчета информацию, а также предоставляющий пользователю информацию о том, какие этапы расчета предшествуют или следуют за текущим. Таймлайн присутствует как на информационном экране, где он выполняет справочную функцию, так и на экране, предназначенном для ввода данных, где он служит для навигации между этапами работы над проектом, а также для связи отдельных окон в единый комплекс подобно тому, как матрица проекта служит интегрирующим элементом для всего набора расчетных модулей [Боргест, 2015b].

В качестве предметной области для реализации робота-проектианта было выбрано предварительное проектирование самолета. С одной стороны это сфера деятельности, которая всегда требовала творческих решений, с другой – она достаточно хорошо формализована и алгоритмизирована. Результатом работы робота-проектианта является

модель изделия. Она состоит из 2 взаимосвязанных частей – матрицы проекта с логикой расчета (в текущем варианте выполненной на основе таблиц MS Excel) и параметризованной трехмерной модели, которая автоматически меняет собственную конфигурацию в зависимости от данных, содержащихся в матрице проекта

При изменении любого параметра самолёта в меню программы, CAD система автоматически пересчитывает и редактирует модель самолёта. В данной работе используется CAD/CAM/CAE система Catia. Интерфейс программы написан на Visual Studio 2012.

2. Параметрическая 3D-модель

На основании данных, полученных при расчете агрегатов, интеллектуальный помощник строит параметризованную трехмерную модель самолета в системе САПР.

Представленная параметрическая 3D модель самолёта была создана в системе CATIA V5 R19, являющейся современной САПР с последовательной объектно-ориентированной архитектурой. Набор имеющихся в ней инструментов, арсенал которых включает возможность формулировки правил взаимосвязи

параметров, проверку накладываемых условий проектирования и создание аналитических зависимостей для автоматической реализации

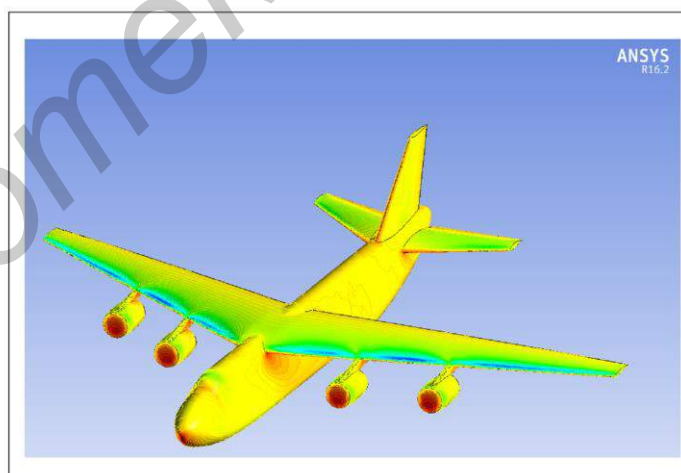


Рисунок 2 – Картина распределения давления по поверхностям самолёта

На рисунке 2 представлен экран интеллектуального помощника проектировщика с картиной распределения давления по поверхностям проектирования. Пользователь может выбрать необходимую визуальную картину представления расчёта: распределения давления, распределения, скорости, линии тока, линии равного давления. Графическое представление считывается с экрана ANSYS и передаётся в меню интерфейса.

функций проектирования, позволяет создавать сложные параметрические модели объектов, в том числе, частей самолёта в виде шаблонов

3. Расчёт аэродинамики

Вычислительная аэро(гидро)динамика или, сокращённо и на иностранный манер, CFD, занимается виртуальным моделированием процессов, связанных с движением газообразной или жидкой сплошной среды. Ее использование позволяет снизить потребность в дорогостоящих натуральных экспериментах («продувок» в аэродинамических трубах), сократить связанные с этим денежные и временные затраты и время выхода изделия на рынок [Чигарев и др., 2001].

Для того, чтобы рассчитать построенную параметрическую 3D модель в CFD, необходимо поместить её в шаблон уже рассчитанной модели.

Общий порядок расчёта: переносится геометрическая модель области пространства, заполненной жидкостью (газом), затем она подвергается дискретизации на конечные элементы, учитываются граничные условия (скорость потока, втекающего через поверхности), осуществляется сам расчёт и отображение результатов [Абдуллин, 2011].

Для расчётов использовалась CFD-система ANSYS CFX, интегрированная в оболочку ANSYS Workbench [Ментер и др., 2005].

самолёта. Расчёт производится только после прохождения всех этапов

Пользователь видит только результат аэродинамического расчёта.

В процессе аэродинамического расчета снимаются значения коэффициента подъёмной силы, коэффициента аэродинамического

сопротивления и коэффициента давления на разных углах атаки и при разных числах Маха. Далее эти результаты передаются в матрицу проекта для сравнения и корреляции результатов расчетов, выполняемых классическими методами в расчетном блоке работа-проектанта.

Заключение

Интеграция с расчётным ядром ANSYS позволит провести аэродинамический расчёт созданной параметрической 3D-модели. Это даёт проектанту точную оценку аэродинамических параметров модели, сравнить с допускаемыми показателями. Интегратор уменьшит временные затраты проектанта на расчёт, не будет необходимости открывать отдельно открывать модель, задавать начальные и граничные, интегратор выполнит это автоматически. В дальнейшем интегратор можно использовать в расчёте конструкции на прочности.

Библиографический список

- [Абдуллин, 2011] Абдуллин А. Я. Верификация программного комплекса ANSYS CFX на задачах обтекания жидкостью удобообтекаемых тел / А.Я. Абдуллин, Н.С. Сеношкин, К.В. Порошкин // Молодой ученый. — 2011. — №7. Т.1. — С. 49-53.
- [Боргест и др., 2015a] Боргест Н.М., Власов С.А., Коровин М.Д. Реализация удаленного управления параметризованной трехмерной моделью самолета с помощью клиент-серверного приложения / Н.М. Боргест, С.А. Власов, Д.В. Шустова // материалы международной научно-технической конференции OSTIS-2015. БГУИР. 2015
- [Боргест и др., 2015b] Боргест Н.М. Власов С.А., Громов Ал.А., Громов Ан.А., Коровин М.Д., Шустова Д.В. Робот-проектант: на пути к реальности. // Онтология проектирования, N4(18), 2015 — с. 429-449.
- [Боргест и др., 2012a] Боргест Н.М., Чернов Р.В., Шустова Д.В. Разработка интерфейса интеллектуального помощника проектанта / Н.М. Боргест, Р.В. Чернов, Д.В. Шустова // материалы международной научно-технической конференции OSTIS-2012. БГУИР. 2012
- [Боргест и др., 2012b] Боргест Н.М., Робот-проектант: фантазия и реальность/ Н.М. Боргест, М.Д. Коровин, А.А. Громов, А.А. Громов, Д.В. Шустова // Онтология проектирования, N4(6), 2012 — с. 73-94.
- [Власов, 2015] Власов С.А., Разработка интерфейса помощника робота проектанта на основе тезауруса предметной области// материалы международной научно-технической конференции ИПРМ'2015. УГАТУ. 2015.
- [Чигарев и др., 2001] Чигарев, А.В. и др. ANSYS для инженеров: Справочное пособие. / А.В. Чигарев и др.// Справочное пособие. М.: Машиностроение-1. —2001. — 510 с.
- [Blessing, 2009] Blessing, L.T.M., Chakrabarti, A.: DRM, a Design Research Methodology, // Blessing, , L.T.M., A. Chakrabarti, / Springer, 2009. 413 p.
- [Ментер и др., 2005] Ментер Ф., Лантри Р. Аэродинамика пассажирского самолёта / Ф. Ментер, Р. Лантри // Инженерно-технический журнал «ANSYS Solutions. Русская редакция» №1(1), 2005. — с. 12-14.

CONNECTION OF MODULE INTERFACE INTELLECTUAL ASSISTANT WITH CAE-SYSTEM ANSYS

Korovin M.D., Vlasov S.A.

*Samara State Aerospace University
(national research university),
Samara, Russian Federation*

maks.korovin@gmail.com

In the article the basic principles of creating modular interface intelligent assistant to the designer with CAE (Computer-Aided Engineering) system Ansys. The aim of this work is to obtain aerodynamic calculation of the created parametric models of the aircraft when using the smart assistant robot designer. Work includes the creation of additional integration of the communication module of the robot designer and CAE-system Ansys, the results of the aerodynamic calculation.

Introduction

The design of new technical systems is one of the most difficult types of engineering creative activity. [1] the Main feature of conceptual design is the necessity of the adoption of sets of solutions when there is insufficient or, conversely, redundant information, the robot the planner serves as an assistant planner, able to help make a decision, calculate all possible options, to simplify mathematical calculation and parametric 3D model of the aircraft. Getting the final results, it is necessary to check the validity of the calculations, for this is a special integrator, allowing you to transfer model for calculation in Ansys.

Main Part

The interface technology of the robot-designer and ANSYS CFX describes:

- Advantages of the Integrator;
- Intellectual interface Assistant;
- Parameterized 3D models planes;
- Aerodynamic calculation of 3D models.

Conclusion

Integration with the ANSYS calculation kernel will allow for aerodynamic calculation of the created parametric 3D models. This gives the designer an accurate assessment of the aerodynamic model parameters, compare with acceptable performance. The integrator will reduce the amount of time spent on the calculation of the project designer, will not need to separately open to open the model, set the initial and boundary, an integrator will do this automatically. In the future, the integrator can be used in calculating the design strength.



УДК 005.8:615.478

АЛГОРИТМЫ ОБРАБОТКИ ИНФОРМАЦИИ В АДАПТИВНОМ РЕКОНФИГУРИРУЕМОМ МОДУЛЕ САПР ОК ДЛЯ ВИЗУАЛИЗАЦИИ КОНТУРОВ ТЕПЛОВЫХ ПОТОКОВ

Сорокин О.Л. Сидоркина И.Г.

Поволжский государственный технологический университет

г. Йошкар-Ола, Россия

oleg-ussr2@yandex.ru

igs592000@mail.ru

В работе рассмотрены основные алгоритмы работы модуля визуализации контура тепловых потоков с использованием нечеткой логики. Описано применение матриц нечетких решений и введение базы знаний на основе фреймовой структуры. Приведены основные этапы функционирования алгоритмов и их преимущества.

Ключевые слова: тепловой поток, модуль визуализации, САПР, принятие решений, адаптация, определение ситуации, САПР ограждающих конструкций, нечеткая логика.

Введение

С целью разрешения неоднозначных ситуаций, в том числе для более точной визуализации в настоящее время появляется все больше тенденций внедрения нечеткой логики в автоматизированных системах [Номерчук,2013]. Использование алгоритмов с нечеткой логикой и внедрение базы знаний с фреймовой структурой позволит избежать ошибок в расчетах тепловых потоков и устранить ситуации с двумя возможными вариантами. Предложен способ создания адаптивного реконфигурируемого модуля для визуализации контуров тепловых потоков (АРМ), на основе реализации алгоритмов классификации и идентификации ситуаций, прогнозирования их развития в реальном времени, а также своевременного синтеза актуальных описаний о поведении системы в различных ситуациях, на основе нечеткой логики.

1. Алгоритм идентификации ситуаций и принятия решений

Исходными данными для формирования алгоритмов являются:

– Обобщенная целевая функция адаптивного модуля и совокупность функций для визуализации.

– Информационная модель адаптивного модуля для визуализации тепловых потоков, построенная на основе совокупности проведенных измерений с лабораторной установки.

– Иерархия наложений потоков (от нагревателей, от ограждающих конструкций)

– Логические закономерности и методы для распознавания и идентификации ситуаций теплового контура.

Результатом обработки измерительной информации получаемой с множества датчиков – распознавание вариантов ситуаций с последующей трассировкой нагревателей. Иерархия наложения потоков представляет собой слияние двух фронтов – теплого от нагревателей и холодного от контура. Результатом является общий тепловой поток, который используется в дальнейшем для принятия решений о перекомпоновке нагревателей. Идентификация ситуаций теплового контура в конечном итоге имеет только два варианта: отсутствие проблемных зон (зон промерзания) или их наличие.

Определим основные шаги алгоритма идентификации ситуаций и принятия решений:

Шаг1. На основе сформированной априорной матрицы ограничений с использованием методики линейного программирования формируем доверительные интервалы.

Шаг2. Формируем матрицу ограничений для контура, определяющую границы воздействия теплового потока.

Шаг3. Проверяется матрица «код-решение» на существование адекватного решения

Шаг 4. Если решение найдено то по коду операции принимаем решение о визуализации теплового потока определенным образом.

Шаг5. Если решение не найдено или существует

несколько решений (нечеткая ситуация), то с использованием матрицы вероятностей находится решение с максимальной вероятностью.

Шагб. Если при опросе матрицы «код-решение» однозначная ситуация отсутствует, по бинарному коду отношений из памяти значений выбирается наиболее подходящий сектор фреймов, где затем по критерию уверенности идентифицируется наиболее подходящая ситуация.

Таким образом, функция выбора новой ситуации реализована путем выбора (по коду бинарных операций) наиболее подходящего сектора фреймов в базе знаний (БЗ).[2]

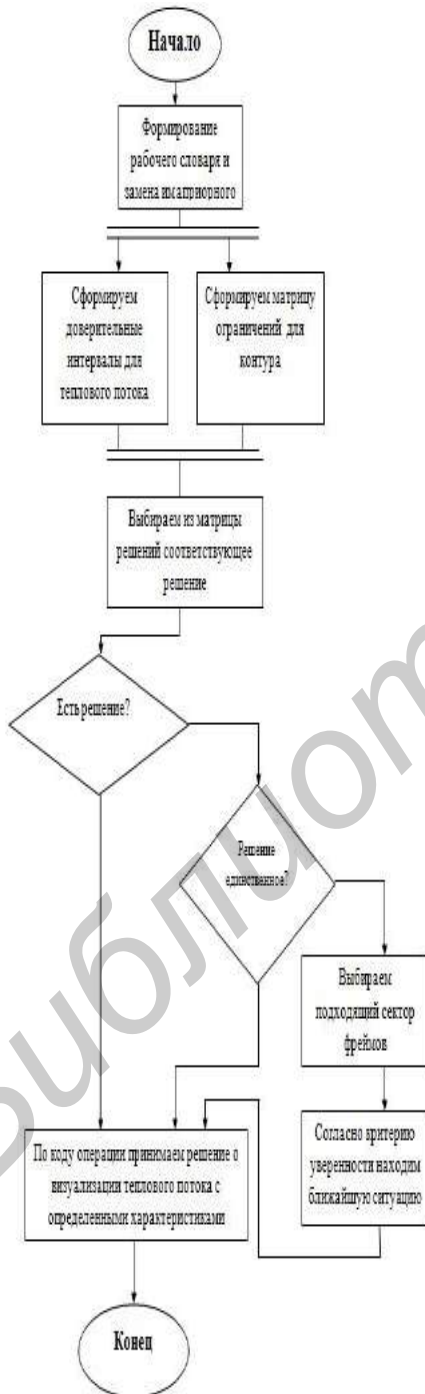


Рисунок 1 -- Алгоритм идентификации ситуаций и принятия решений в САПР ОК

2.Обобщенный алгоритм работы адаптивного реконфигурируемого модуля для визуализации контуров тепловых потоков

Обобщенный алгоритм работы модуля представлен на Рис. 2.



Рисунок 2 -- Обобщенный алгоритм идентификации ситуаций и принятия решений в САПР ОК

На начальном шаге работы алгоритма происходит анализ изменений вида контура, в ходе чего инициализируется команда на изменение

параметров. При ее инициализации выполняется детектирование температуры с датчиков и формирование выборок данных. На основе выборок температурных данных происходит распознавание ситуации: наличие или отсутствие проблемных зон. Суть метода распознавания ситуации заключается в возможности задания температурных условий окружающей среды вне контура, а также выдаче советов в случае, когда тепловой поток от нагревателей не может прогреть весь контур, например на Рис.3. показана нерациональная компоновка.

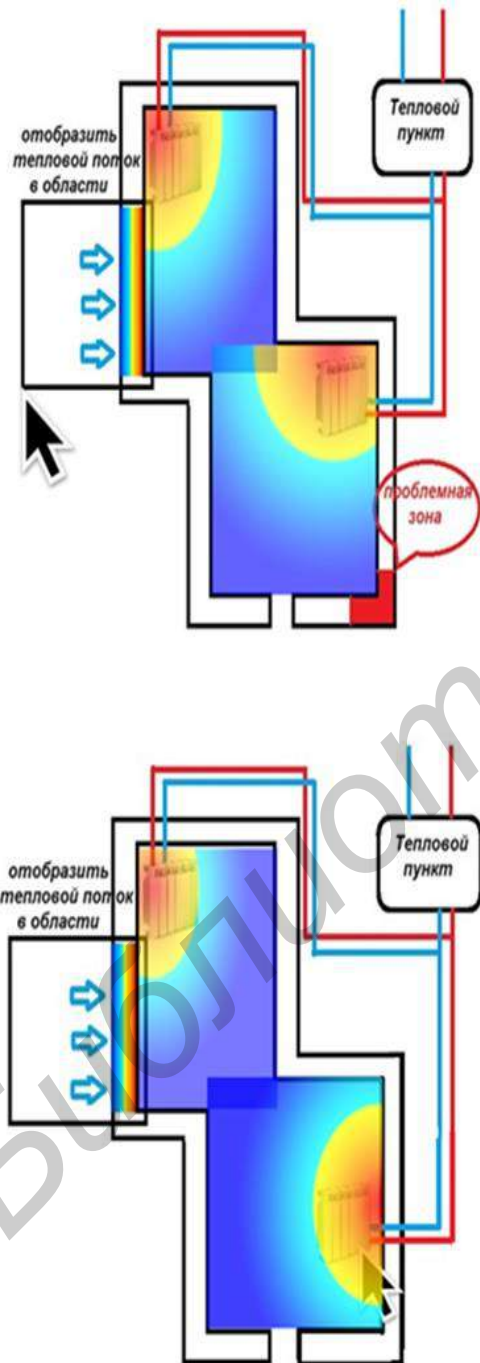


Рисунок 3 -- Пример распознавания ситуации по наличию или отсутствию проблемных зон в САПР ОК.

Задача модуля [Сорокин,2015]САПР ограждающих конструкций (ОК) предложить

наиболее оптимальный вариант компоновки, выполнить трассировку связи с тепловым пунктом, а также визуализировать полученные тепловые потоки от нагревателей и в контуре здания. Конечным этапом алгоритма является визуализация характера теплового потока для определенной ситуации с учетом алгоритма идентификации ситуаций и принятия решений. Практическое использование предлагаемого модуля возможно, как при проектировании новых зданий и сооружений, так и при реконструкции старых. Также данный модуль будет полезен для моделирования стресс тестов для контура здания с целью определения срока износа конструкции при заданных параметрах. Пример визуализации контура теплового потока представлен на Рис.4.

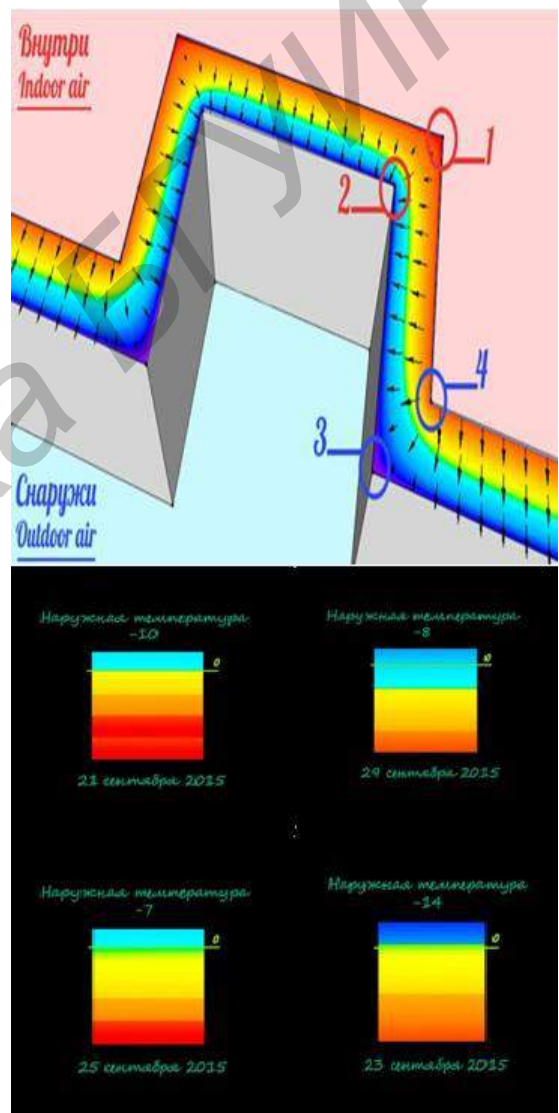


Рисунок 4 -- Пример визуализации контура теплового потока с выделением проблемных зон.

Разработка САПР наружных инженерных сетей, в частности наружных ОКзданий и сооружений, позволяет избежать ряда проблем. Визуализация данных в расчетах модуля САПР позволяет снизить ошибку в вычислениях и повысить наглядность результатов. Использование САПР ОК позволяет избежать сноса зданий или дорогостоящей их

обшивки снаружи различными теплоудерживающими материалами. Эффективность в расчетах контура обогрева позволяет снизить время на расчеты параметров стены до минимума и повысить их точность, которая является важным требованием при проектировании зданий и сооружений.

Заклучение

Необходимость использования нечеткой логики и основных приемов для синтеза и идентификации ситуаций в автоматизированных системах, позволило применить алгоритм идентификации ситуаций и принятия решений, адаптивного реконфигурируемым модулем, который берет во внимание все возможные исходы в ходе визуализации данных: определенный, с использованием вероятности и с помощью фреймовой БЗ. На основе такого алгоритма разработан общий алгоритм работы реконфигурируемого модуля для визуализации контуров тепловых потоков.

Использование комплекса данных алгоритмов позволит наиболее точно визуализировать тепловые потоки в мультимедийном интерфейсе САПР ОК, что повысит точность всех расчетов и повлияет в дальнейшем на качество компоновки нагревательных элементов в контуре конструкции.

Библиографический список

[Номерчук, 2013]Номерчук, А.Я. Методы построения автоматизированных систем обработки информации управления на основе нечеткой логики/С. Б. Фрадкин// Технические науки ЮФУ-2013. С. 56-60

[Чаус, 2011] Формирование базы данных и базы знаний в эксплуатируемых системах технического диагностирования / ЧаусО.В.– Вестник ГГТУ, 2011.

[Сорокин,2015]Модуль определения стационарного режима в САПР наружных инженерных сетей/ Сорокин О.Л.-«IS&IT–Интеллектуальные САПР 2015» труды конгресса по интеллектуальным системам и информационным технологиям – Таганрог: Изд-во ЮФУ,2015-Т.1. – С.70-75

DATA PROCESSING ALGORITHMS IN ADAPTIVE RECONFIGURABLE MODULE PROTECTING DESIGHS CAD FOR RENDERING A PROFILE OF HEAT FLOW

Sorokin O.L., Sidorkina I.G.

*Volga State University of Technology,
Yoshkar-Ola,Russia*

oleg-ussr2@yandex.ru

igs592000@mail.ru

The basic algorithms visualization module circuit heat flows using fuzzy logic. The application of fuzzy matrices making and administering the database based on frame structure. The main steps of operation of the algorithms and their advantages.

Introduction

To resolve ambiguous situations, including more accurate visualization currently there are more trends implementation of fuzzy logic in automated systems. Using algorithms, fuzzy logic and implementation of a database with frame structure to avoid errors in the calculation of heat flows and address the situation with two possible options.

Main Part

Based on the temperature data samples occurs recognition situation: the presence or absence of problem zones. The method of recognition of the situation lies in the ability to set ambient temperature conditions outside the contour, as well as granting councils in the case where the heat flux from the heater can heat the entire path.Task CAD to offer the most optimal variant layout, trace a connection point with the heat, as well as visualize the resulting heat flow from the heater and the loop of the building. The final step of the algorithm is to visualize the nature of heat flow to a specific situation in view of the identification algorithm situations and decision-making.

The result of the processing of the measuring information received from multiple sensors - Recognition of options situations subsequent trace heaters. Hierarchy overlay flows represents a fusion of the two fronts - the heat from the heaters and the cold loop. The result is the total heat flow, which is then used to make decisions about repackaging heaters. Identifying situations thermal circuit eventually has only two options: no problem zones (zones freezing), or their availability.

The task module CAD walling to offer the most optimal variant layout, trace a connection point with the heat, as well as visualize the resulting heat flow from the heater and the loop of the building. The final step of the algorithm is to visualize the nature of heat flow to a specific situation in view of the identification algorithm situations and decision-making. The practical use of the proposed module is possible, as in the design of new buildings and the reconstruction of old ones.

Conclusion

The need to use fuzzy logic and basic techniques for the synthesis and identification of situations in automated systems, allowed to use the algorithm of identification of situations and decision-making, adaptive reconfigurable module, which takes into account all the possible outcomes in the visualization of data definition, using probability and with the help of framing knowledge base. On the basis of this algorithm, we developed a general algorithm of the reconfigurable module to visualize the contours of heat flows.

Using complex algorithms, the data will allow the most accurate visualization of heat flows in multimidiynom CAD interface, which will increase the accuracy of calculations and impact in the future on the quality of the layout of the heating elements in the circuit design.



УДК 681.3.053

ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДЕЛЕЙ ЗРЕЛЫХ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ В КОНТЕКСТЕ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ПОТОКАМИ КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКИХ РАБОТ

Ларин С.Н.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

larinmars@rambler.ru

В статье предлагается и описывается комплексный подход по управлению конструкторско-технологическими решениями. На основе моделей зрелости производственных процессов проведены исследования организационных предпосылок к системному совершенствованию конструкторско-технологических решений (КТР) в рамках потоков работ конструкторско-технологической подготовки производств. На основе объединения совокупности задач КТПП обосновывается необходимость создания инструментальной среды для моделирования потоков технологических работ. Представлены обобщенные задачи моделирующей среды TechWIQA.

Ключевые слова: конструкторско-технологическая подготовка, зрелость производственных процессов, методология зрелых производственных процессов (СММ), автоматизированное управление потоками технологических работ, проектирование, разработка технологии, поэтапное совершенствование.

Введение

В настоящее время особую важность приобретает разработка автоматизированных систем (АС), интенсивно использующих программное обеспечение. Проблематичность данной предметной области регулярно подтверждается статистическими отчетами корпорации Standish Group. Чрезвычайно низкая успешность разработок АС указывает на то, что исследования этой проблемы исключительно актуальны [Соснин, 2008]. А значит, в разработках АС, ориентированных на использование критериев и факторов успешности, необходим конструктивный учет конструкторско-технологической деятельности, обеспечивающей создание аппаратных составляющих АС, согласованных с соответствующим программным обеспечением.

В статье предлагается адаптировать к конструкторско-технологической деятельности (в процессах разработок АС) модель профессиональной зрелости процессов, доведенную в ее приложении к программному обеспечению до стандарта Capability Maturity Model Integrated for Development: Version 1.3 (CMMI Dev.1.3). Адаптация доведена до модели профессиональной зрелости конструкторско-технологических процессов, для которой ниже будет использоваться аббревиатура СММ-КТР.

1. Исходные предпосылки

Модель профессиональной зрелости СММ ориентирует тех, кто ее внедряет в свои профессиональные процессы, на их постоянное совершенствование, нацеленное на эволюционное достижение нормативных уровней: «повторяемый» (repeatable, уровень 2), «определенный» (defined, уровень 3), «управляемый» (managed, уровень 4) и «оптимизированный» (optimized, уровень 5). Сам факт такого изначального абстрагирования подсказывает, что этот принцип «постоянного управляемого совершенствования профессиональных процессов» можно применить к любой профессиональной деятельности, а значит и к конструкторско-технологической деятельности (рис. 1).

Поскольку этот принцип наиболее последовательно и детально раскрыт и специфицирован в стандарте CMMI Dev.1.3, этот стандарт логично использовать как источник заимствований для построения аналога для конструкторско-технологической деятельности (для построения модели СММ-КТР).

В модели СММ-КТР различаются два вида активности – конструкторская и технологическая, которые принципиальным образом связаны друг с другом, что раскрывает следующий принцип:

Усилия по совершенствованию профессиональной технологических процессов осуществляются в контексте конструкторских процессов, совершенствование которых должно выполняться в контексте технологических процессов.	зрелости должны в контексте процессов,
--	--

2. Специфика модели СММ-КТР

Анализ разработок, выполненных в предыдущие годы [Соснин, 2007], подтверждает осознанное стремление проектировщиков к интенсификации своего труда путем автоматизации и интеграции конструкторских и технологических разработок. Необходим механизм, который накапливает, хранит и предоставляет для использования проверенные на практике и специально подготовленные комплексы знаний по конструкторско-технологическому проектированию технических средств (ТС), подготовке и организации их производства.

Структура зрелости процессов, в которую вошли эти принципы качества, была впервые намечена Филиппом Кросби. Сетка зрелости управления качеством, приведенная Кросби, описывает пять эволюционных фаз во внедрении системы управления качеством. Эта структура зрелости была адаптирована для процесса управления конструкторско-технологическими разработками.

Модель зрелости процессов разработки КТР предоставляет организации-разработчику руководящие принципы управления своими процессами разработки и сопровождения КТР. СММ-КТР предназначена для выбора стратегий совершенствования процессов путем определения текущего уровня зрелости производственного процесса и выявления некоторых вопросов, наиболее значимых для повышения качества создаваемого КТР и совершенствования процессов.

Объединение преимуществ систем 3D-моделирования с возможностями информационных систем, обеспечивающих коллективную работу конструкторов, проектантов, технологов, операторов станков с ЧПУ (системы PDM и Work Flow), позволяет использовать процессные методы организации управления и на этой основе кардинально изменить систему КТПП. Для этого необходимо запараллелить последовательный процесс: конструкторская разработка – конструкторская документация – технологическая разработка – технологическая документация – разработка программ для станков с ЧПУ.

Практическая реализация workflow-систем для управления КТПП ставит ряд вопросов интеграции, связанных с необходимостью свести все элементы многокомпонентной системы воедино, преодолеть функциональную несовместимость различных систем. Для этого необходим взаимоувязанный

набор практик, поддерживающих интеграцию различных задач в рамках ТПП.

Концептуальная структура зрелости производственного процесса упорядочивает стадии КТР таким образом, что усовершенствования на каждой предшествующей стадии являются фундаментом усовершенствований последующей стадии. Таким образом, стратегия усовершенствования, предлагаемая концептуальной структурой зрелости производственного процесса, обеспечивает наиболее прямой путь постоянного улучшения уровня конструкторско-технологических решений.

Каждое описание уровней зрелости КТР разбивается на составные части. Разбиение каждого уровня зрелости, кроме первого, варьируется от кратких обзоров уровня до его рабочего определения в ключевых практиках, как показано на рисунке 1.

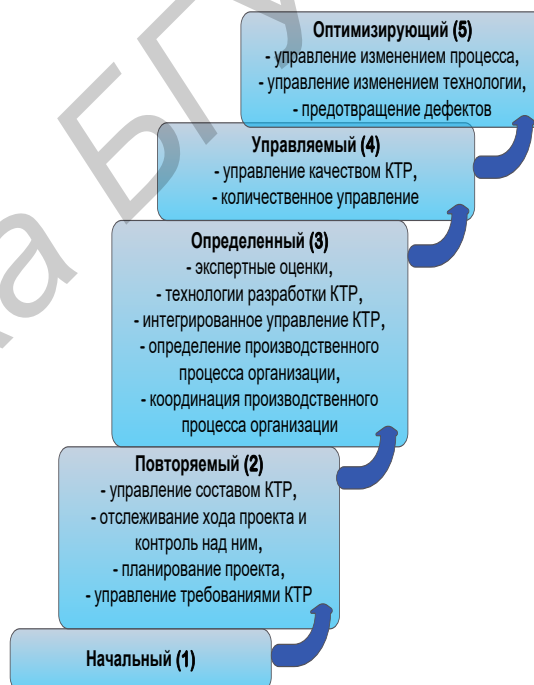


Рисунок 1 – Распределение групп ключевых процессов по уровням зрелости.

В разработках аппаратного обеспечения АС, необходим конструктивный учёт конструкторско-технологической деятельности $R^{КТР}$, обеспечивающей взаимосвязь множества проектных задач, согласованных с соответствующим программным обеспечением.

Конструктивные и технологические составляющие общей работы по подготовке производства взаимно дополняют друг друга, формируют единый поток работ, приводящий к

общему согласованному конструкторско-технологическому решению.

$$R^{KTP} = R_{Z_1}^K \cup R_{Z_2}^{TO} \cup R_{Z_3}^{TC}, \quad (1)$$

Где $R_{Z_1}^K$ - конструкторская деятельность;

$R_{Z_2}^{TO}$ - деятельность по освоению новых и модернизируемых аппаратных средств АС;

$R_{Z_3}^{TC}$ - деятельность по обеспечению высоких технико-экономических показателей серийных изделий.

Деятельность по освоению новых и модернизируемых аппаратных средств АС $R_{Z_2}^{TO}$ представляет следующее множество:

$$R_{Z_2}^{TO} = \{Z_m^{TP}; Z_m^{TC}; Z_m^{OF}; Z_m^{UP}\} \quad (2)$$

Где Z_m^{TP} - обеспечение производства комплектом технологических процессов;

Z_m^{TC} - обеспечение производства средствами технологического оснащения;

Z_m^{OF} - создание необходимых организационных форм производства;

Z_m^{UP} - управление технологической подготовкой производства; m – количество задач.

Несмотря на имеющиеся разработки и прикладные решения, инструментарий проектировщика ТС не имеет обобщенных прикладных компонентов, обеспечивающих принятие зрелых, эффективных КТР. Таким прикладным компонентом становится программный комплекс TechWIQA, созданный на базе вопросно-ответной моделирующей среды WIQA [Ларин и др. 2011]

Дерево задач и вопросно-ответные модели задач, порождаемые в этой среде, позволяют моделировать потоки работ в различных версиях их оперативного «смешивания», обеспечивая поиск рациональных решений с использованием библиотек для типовых конструктивных и технологических работ.

На рисунке 2 показано дерево задач (Z) ТПП исследуемого изделия.

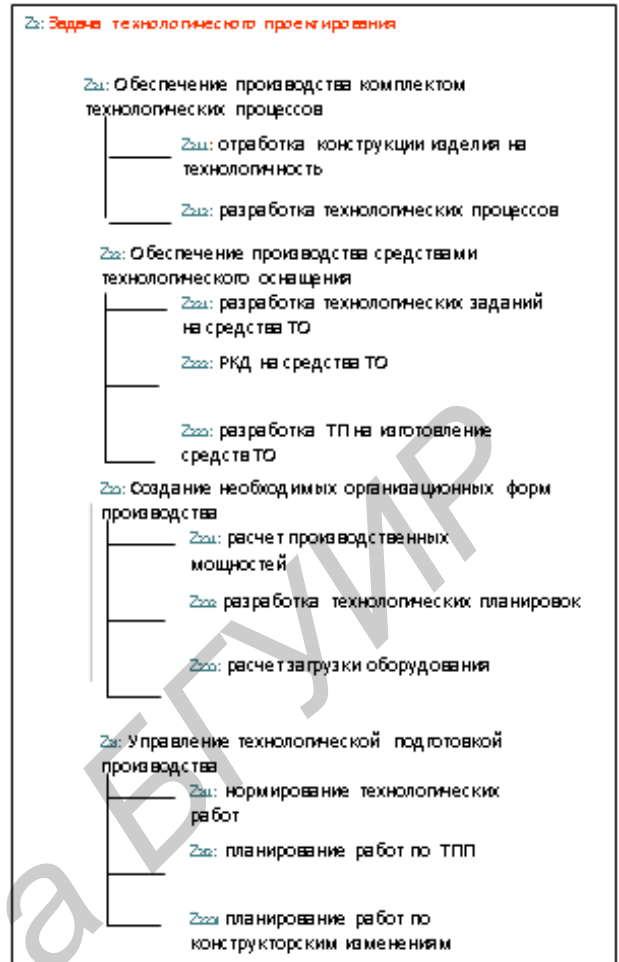


Рисунок 2 – Дерево задач технологического проектирования.

Технологическая подготовка производства наукоемких изделий является важнейшим элементом технической подготовки производства, весьма значительным по своему объему и сложности. На основании уровней зрелости СММ технологическая подготовка разделена на 5 уровней, на каждом из которых решается n типов задач Z_n (рис. 2). Каждая задача характеризуется различным содержанием, объемом работы и степенью точности создаваемой информации. Например, первая задача – первичная технологическая подготовка – характеризуется созданием информации, необходимой для планирования материально-технического снабжения и запуска установочных партий изделий. На этом этапе решаются следующие задачи:

- разработка межцеховых технологических маршрутов;
- составление графиков проектирования и изготовления инструмента с разбивкой на очереди;
- расчет норм расхода материала на детали и сборочные соединения;
- укрупненный расчет размера партий и длительности циклов;
- укрупненный расчет нормативов опережения запуска и выпуска.



Рисунок 3 – Технологические задачи, порождаемые уровнями зрелости КРТ.

Требования СММИ, в части постоянного совершенствования производства и улучшения качественных и экономических показателей выпускаемой продукции вызывают необходимость непрерывной работы технологов и конструкторов по дальнейшему усовершенствованию ранее разработанных мероприятий технологической подготовки производства.

В общем случае объем и содержание технологической подготовки производства характеризуются следующими задачами, основными из которых являются следующие:

- составление технологической документации на подготовку производства (технологические карты, инструкции и др.);
- составление ведомости необходимого технологического оборудования;
- составление ведомости необходимого покупного инструмента;
- проектирование и изготовление специального оснащения;
- проектирование и изготовление нестандартного оборудования;
- составление ведомости норм расхода материалов;
- расчет рабочей силы и обслуживающего персонала;
- расчет необходимых производственных и вспомогательных площадей;
- проектирование внутрицехового и межцехового транспорта;
- разработка системы технического контроля производства;
- разработка системы планирования производства.

На основании проведенных исследований очевидно, что обеспечение технологичности конструкции изделия играет исключительную роль в функции подготовки производства, которая предусматривает взаимосвязанное решение конструкторских и технологических задач, направленных на повышение производительности труда (СММ-КТР (3 уровень)), и сокращение времени на производство (СММ-КТР (4 уровень)),

достижение оптимальных трудовых и материальных затрат (СММ-КТР (5 уровень)).

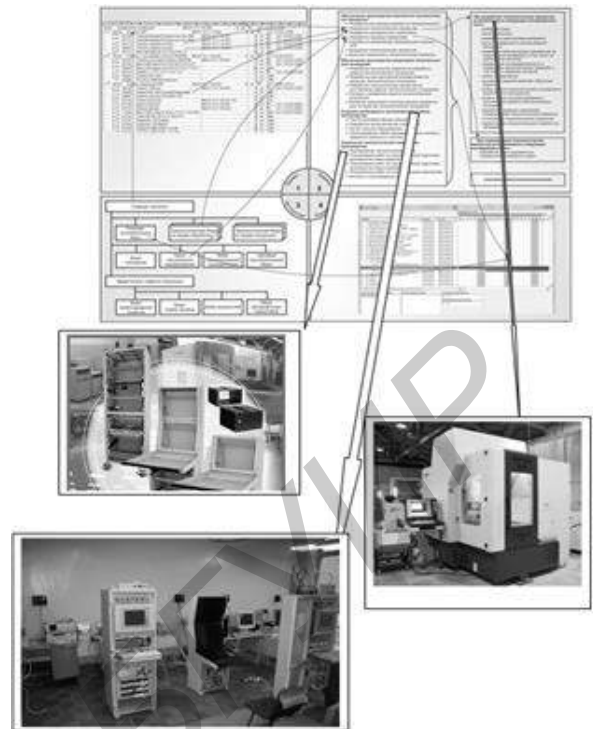


Рисунок 4 – Структура программного комплекса TechWIQA.

Обеспечение технологичности конструкции изделия включает:

- отработку конструкции изделия на технологичность на всех стадиях разработки изделия при технологической подготовке производства и в обоснованных случаях при изготовлении изделия;
- совершенствование условий выполнения работ при производстве, эксплуатации и ремонте изделий и фиксацию принятых решений в технологической документации;
- количественную оценку технологичности конструкции изделия;
- технологический контроль конструкторской документации;
- подготовку и внесение изменений в конструкторскую документацию по результатам технологического контроля, обеспечивающих достижение базовых значений показателей технологичности.

Знание этих свойств и применение математических методов для их количественной оценки (СММ-КТР (4 уровень)) позволяют воздействовать на них в требуемом направлении и оптимизировать (СММ-КТР (5 уровень)) перечисленные выше затраты при безусловном обеспечении установленных значений других показателей качества и принятых условиях разработки и изготовления.

Таким образом, обеспечение технологической рациональности конструкций и оптимизация уровня преемственности конструктивных решений (СММ-

КТР (5 уровень)) выступают в качестве основных источников повышения эффективности КТР и позволяют сформулировать следующий принцип:

Повышение зрелости процессов принятия конструкторско-технологических решений должно приводить к существенному сокращению циклов подготовки производства и создавать предпосылки для использования концепций параллельного инжиниринга (Concurrent Engineering /параллельное проектирование/)

Возможности разработанного и апробированного программного комплекса (TechWIQA) [Карпушин и др. 2012] позволяют выполнять такую интегральную оценку влияния локальных КТР и выдавать ее специалистам проектировщикам в режиме реального времени. Предложенная авторами версия системы автоматизированного проектирования с применением «технологического блока» в дальнейшем нашла развитие на базовом предприятии ФНПЦ ОАО «НПО «Марс» при создании программного обеспечения для информационного обеспечения САМ-систем.

Автоматизированная система вопросно-ответного концептуального проектирования технологической документации и управления процессами технологической подготовки производства при изготовлении наукоемких изделий значительно повышает эффективность управления технологической подготовкой производства на предприятии за счет:

- сокращения сроков проведения проектов и, соответственно, всего процесса производства в целом, за счет повышения зрелости производственных процессов;
- повышения качества КТР в целом для объектов наукоемкого производства;
- снижения затрат на разработку конструкторско-технологической документации (маршрутных карт, ведомостей деталей).

Важным достоинством разработанного комплекса средств является моделирование технологических процессов в формах потоков работ и средств управления потоками, базовой единицей которых является «задача».

Для представления «задач» и их связанных совокупностей используется разработанный программный комплекс TechWIQA, созданный на базе вопросно-ответной моделирующей среды WIQA.

В текущей версии TechWIQA поддерживается формирование маршрутных карт (рис. 5) и технологических инструкций.

Описание функциональных возможностей и принципов практического использования разработанных компонент, проведенный в контексте общего сценария разработки и управления

единичными технологическими процессами изготовления, ремонта изделий и их составных частей с использованием вопросно-ответной среды TechWIQA, детально раскрыт в рамках выполнения НИР «Комплекс программных средств разработки и управления процессами технологической подготовки производства изделий ФНПЦ «ОАО НПО «МАРС»».

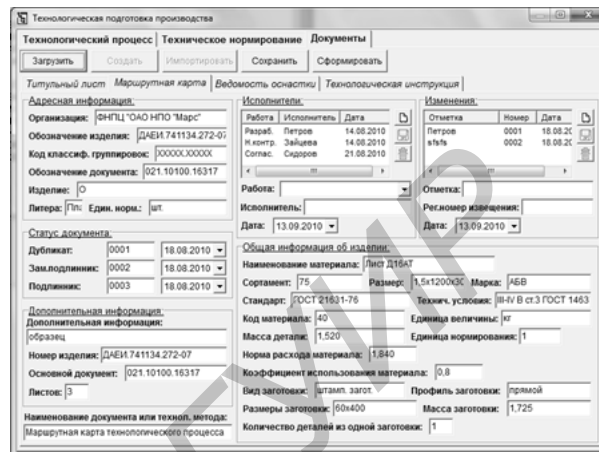


Рисунок 5 – Структура программного комплекса TechWIQA.

Маршрутная карта может быть сформирована в готовом для печати виде по стандарту ГОСТ 3.1118-82 в документе Microsoft Word. Комплекс программных средств разработки и управления процессами технологической подготовки производства изделий предназначен для автоматизации формирования комплектов документов на единичные технологические процессы изготовления, ремонта изделий и их составных частей, а также формирования и управления потоками технологических работ, планирования сроков.

Разработанный программный комплекс TechWIQA обеспечивает:

- распределение зарегистрированных задач исполнителям и рабочим группам, а также ввод плановых сроков начала и окончания выполнения задач;
- автоматизированный ввод данных структуры изделий, а также регистрацию задач по формированию комплектов документов на изделия и их составные части;
- возможность групповой работы с данными, посредством удаленного авторизуемого доступа к серверу;
- моделирование организационной структуры коллектива, разрабатывающего единичные технологические процессы изготовления, ремонта изделий и их составных частей, а также комплекты документов на них;
- автоматизированный ввод и импорт необходимой справочной информации;
- автоматизированный ввод данных операций технологических процессов и данных технического нормирования;
- автоматизированный ввод данных

документов на единичные технологические процессы, в соответствии с выбранной комплектностью и видом описания;

Конкретные практики, подлежащие выполнению в каждой группе ключевых процессов, эволюционируют по мере достижения организацией более высоких уровней зрелости.

SMM-KTR обеспечивает понимание и организует системную работу по интеграции всех информационных ресурсов предприятия, что приводит к существенному снижению стоимости и повышению эффективности бизнес-процессов предприятия, что позволяет получить следующие результаты:

1. Получение оперативной информации о текущих результатах деятельности конструкторско-технологических подразделений как в целом, так и с полной детализацией по отдельным видам ресурсов;
2. Решение задач оптимизации производственных и материальных потоков;
3. Планирование и контроль за всем циклом принятия КТР с возможностью влияния на него в целях достижения оптимальной эффективности в использовании производственных мощностей и всех видов ресурсов.

Библиографический список

[Соснин, 2008] Соснин П. И. Концептуальное моделирование компьютеризованных систем: учеб. пособие. УлГТУ, 2008-. Ульяновск. -220 с.

[Соснин, 2007] Соснин П. И. Вопросно-ответное моделирование в разработке автоматизированных систем.: УлГТУ, 2007-Ульяновск. - 333 с.

[Ларин и др. 2011] С.Н. Ларин, В.А. Маклаев, П.И. Соснин. Методический базис конструкторско-технологических решений с позиций зрелости производственных процессов/Автоматизация процессов управления. Вып. 4 (26) – Ульяновск., 2011 – С. 55-65.

[Карпушин и др. 2012] А.Н. Карпушин, С.Н. Ларин, П.И. Соснин. Комплекс средств аспектно -ориентированного проектирования систем потоков работ конструкторско-технологической подготовки опытного приборостроительного производства/Автоматизация процессов управления. Вып. 6 (28) – Ульяновск., 2012 – С. 66-75.

PRACTICAL IMPLEMENTATION OF MODELS OF MATURE FIELDS PROCESSES IN THE CONTEXT OF AUTOMATED CONTROL OF THE FLOWS OF DESIGN-ENGINEERING WORKS

Larin S. N.

*Ulyanovsk State Technical University,
Ulyanovsk, Russian Federation*

larinmars@rambler.ru

Introduction

The paper proposes and describes an integrated approach for the management of design and technological solutions. Model-based maturity of production processes conducted a study of organizational preconditions for systematic improvement of design and technological solutions (KTR) within the workflow of design and technological preparation of production. By combining the targets etc is the necessity of the tool environment for modeling flows of technological work. Generalized task modelling environment TechWIQA.

Main Part

Practical implementation of workflow systems for the management of design and technological preparation of production poses a number of integration issues associated with the need to bring all the elements of a multicomponent system together, to overcome the functional incompatibility of different systems. This requires an interconnected set of practices that support the integration of different tasks within the technological preparation of production. A set of software development tools and management processes of technological preparation of production of products is designed to automate the formation of sets of documents on single technological processes of manufacture, repair products and their components, as well as the formation and management of technological flows of works, planning of deadlines

Conclusion

Specific practices to be performed in each group of key processes evolve as the organization will achieve higher levels of maturity.

SMM-KTR understands and organizes systematic work on the integration of all information resources of the enterprise, resulting in significant cost reduction and increased efficiency of business processes of the enterprise



УДК 658.512.22:004.9

РАЗРАБОТКА СПРАВОЧНИКОВ ОБЪЕКТОВ ОПЕРАТИВНО-ПРОИЗВОДСТВЕННОГО ПЛАНИРОВАНИЯ НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО ПОДХОДА

Карпаев С.А., Ларин С.Н.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

neonix3000@mail.ru

larinmars@rambler.ru

В статье обосновывается необходимость разработки и применения справочников объектов планирования основанного на онтологическом подходе, в качестве информационной основы для создания комплекса программ автоматизированного проектирования оперативно-производственного плана производства.

Ключевые слова: посменно-суточное планирование, справочник, словарь, онтология, 1С:Предприятие.

Введение

В настоящее время эффективность приборостроительного предприятия в значительной степени определяется автоматизацией его бизнес-процессов (БП) и их интеграцией. Автоматизация процесса планирования и построения очередности выполнения задач является одной из важных приоритетных направлений автоматизации. Реализация модели оперативно-производственного планирования (ОПП) является в настоящее время трудоёмким и нетривиальным, однако поддается алгоритмизации и автоматизации. Это процесс, требующий внимания к алгоритмам и особенностям производственного процесса [Первин и др., 1973].

Любое предприятие, которое производит электронику, представляет из себя производство с большим потоком операций (конвейерного или позаказного производства) представленного типа (рис.1). Как правило, поток начинается с большого объёма материалов и покупных компонентов, которые объединяются по мере продвижения по маршруту. Итогом всего пути является готовая продукция или некий продукт, который можно использовать как готовый элемент большой системы. К такому типу можно отнести технологические, сборочные и прочие операции. К подобным системам можно отнести производство электроники любого вида. Если при конвейерном производстве, работа подразделений (участков, цехов) сводится к постоянному поддержанию остатков готовых изделий на складе, то при

заказном производстве основной целью является изготовление изделий точно в срок.

В настоящее время существует множество программных решений по реализации распределения потоками работ (такие как, «SAP», «1С:Предприятие», «Order» и др.), однако во всех решениях отсутствует аналитика выполнения потоков задач, вследствие чего формируемые планы неточны. Так же отсутствуют принципиально необходимые в программных продуктах критерии, по которым сама возможность загружать оборудование будет оправдана с максимальным эффектом.

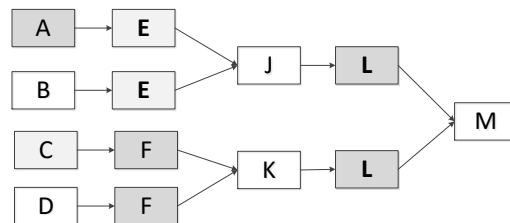


Рисунок 1 – Поток задач на производстве электроники.

Метод, который позволит решить проблему планирования – метод перебора дерева решений с многовариантными расчетами [Татевосов, 1985].

1. Постановка проблематики исследования

На сегодняшний момент одним из узких мест в формировании ОПП плана является формирование плана без учёта ограничений, которые присуще

каждому производственному предприятию. К таким ограничителям можно отнести: объёмы доступных мощностей, объёмы доступной рабочей силы, коэффициент переработки, срок изготовления изделия и т.п. При разработке ОПП необходимо учесть так же технологические особенности производства, например, особенности при перемещении из цеха в цех полуфабрикатов, межоперационное время изготовления, время простоев рабочих центров, исключительные ситуации (например, неработоспособность станка или отсутствие в связи с болезнью исполнителя) [Чудаков и др., 1986].

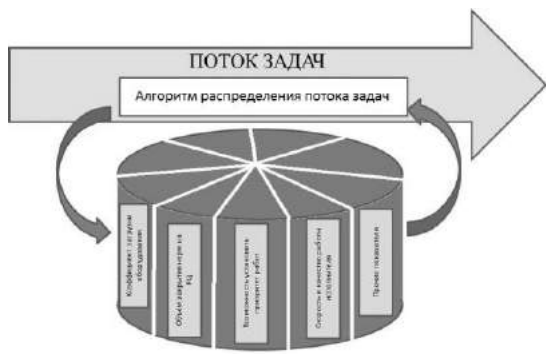


Рисунок 2 – Модель управления потоком задач

Исходя из выше сказанного в ходе проектирования системы ОПП необходимо формировать на каждый вид объекта системы справочные данные.

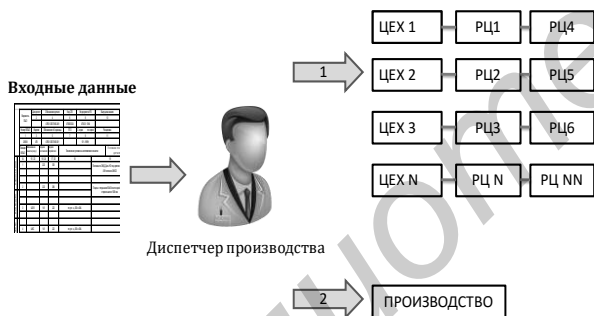


Рисунок 3 – Варианты исполнения плана

Следует отметить, что при формировании планов все словари связываются между собой образуя при этом множественные отношения: «один ко многим» (1:M) и «многие к одному» (M:1).

Основываясь на изложенных выше положениях, авторами предлагается разработать более подробный справочник объектов по ОПП.

2. Разработка и назначение справочников ОПП

Справочник ОПП (рис. 4) - представляет собой систематизированный свод наименований классификационных группировок объектов классификации – планов, деталей, станков и исполнителей.

В справочнике ОПП при каждом новом

формировании и корректировке плана формируется группа объектов и связей между ними. Формируются характеристики объекта изготовления. Таким образом накапливаются знания о том, какой участок и каким исполнителем в принципе может быть изготовлено изделие, которое отмечено в плане на выполнение. Собирая сведения и храня в справочной информации данные об изготовлении - система сможет на основании алгоритмов формировать информацию о формируемых типах и видах изготавливаемых номенклатур с привязкой к участку (цеху).

Процесс формирования справочника начинается с момента обработки неупорядоченного списочного перечня заданий, который выдается на месяц в виде выгружаемого файла (рис.3).

№№	Дата изготовления	Наименование изделия	Наименование участка	Код РЦ	Исполнитель	Наименование станка	Специальный станок	Классификация	Время изготовления
1010	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ1	Иванов И.И.	Станок DMU	DMU1010	1	120
1015	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ2	Петров П.П.	Станок DMU	DMU1010	1	120
1020	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ3	Сидоров С.С.	Станок DMU	DMU1010	1	120
1025	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ4	Кузнецов К.К.	Станок DMU	DMU1010	1	120
1030	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ5	Лебедев Л.Л.	Станок DMU	DMU1010	1	120
1035	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ6	Зиничев З.З.	Станок DMU	DMU1010	1	120
1040	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ7	Березин Б.Б.	Станок DMU	DMU1010	1	120
1045	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ8	Васильев В.В.	Станок DMU	DMU1010	1	120
1050	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ9	Попов П.П.	Станок DMU	DMU1010	1	120
1055	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ10	Смирнов С.С.	Станок DMU	DMU1010	1	120
1060	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ11	Мухоморов М.М.	Станок DMU	DMU1010	1	120
1065	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ12	Ильин И.И.	Станок DMU	DMU1010	1	120
1070	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ13	Воробьев В.В.	Станок DMU	DMU1010	1	120
1075	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ14	Антонов А.А.	Станок DMU	DMU1010	1	120
1080	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ15	Христов Х.Х.	Станок DMU	DMU1010	1	120
1085	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ16	Иванов И.И.	Станок DMU	DMU1010	1	120
1090	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ17	Петров П.П.	Станок DMU	DMU1010	1	120
1095	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ18	Сидоров С.С.	Станок DMU	DMU1010	1	120
1100	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ19	Кузнецов К.К.	Станок DMU	DMU1010	1	120
1105	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ20	Лебедев Л.Л.	Станок DMU	DMU1010	1	120
1110	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ21	Зиничев З.З.	Станок DMU	DMU1010	1	120
1115	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ22	Березин Б.Б.	Станок DMU	DMU1010	1	120
1120	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ23	Васильев В.В.	Станок DMU	DMU1010	1	120
1125	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ24	Попов П.П.	Станок DMU	DMU1010	1	120
1130	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ25	Смирнов С.С.	Станок DMU	DMU1010	1	120
1135	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ26	Мухоморов М.М.	Станок DMU	DMU1010	1	120
1140	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ27	Ильин И.И.	Станок DMU	DMU1010	1	120
1145	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ28	Воробьев В.В.	Станок DMU	DMU1010	1	120
1150	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ29	Антонов А.А.	Станок DMU	DMU1010	1	120
1155	20.12.2018	19172 "Сетка АК-АЛ"	Цех №4-001	РЦ30	Христов Х.Х.	Станок DMU	DMU1010	1	120

Рисунок 4 – Графическое представление перечня файла.

Используя механизмы онтологического подхода записи файла разбиваются на слова (понятия), которые и формируют основу словаря [Соснин, 2007]. Сразу после формирования словаря формируется связи между словами. Связи между словами хранятся в виде дерева иерархии словаря. Детальные классификационные характеристики объекта изготовления так же систематизируются в словари.

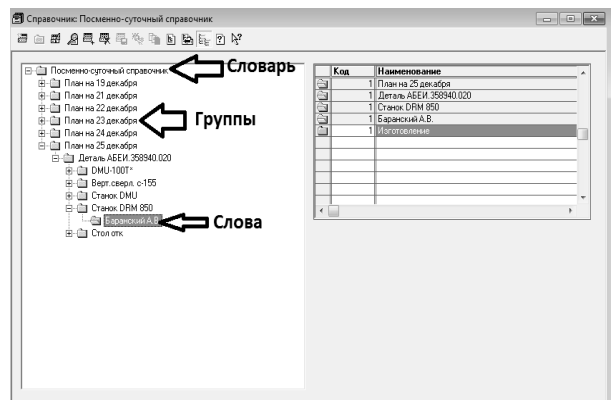


Рисунок 5 – Графическое представление дерева словаря.

Основанием для разработки справочника явилось отсутствие системы накопления опыта, которая позволяет наглядно диспетчеру проанализировать ситуации, связанные с перераспределением задач с учетом возможностей,

наложенных на момент запуска плана и провести ручную корректировку плана.

Классификатор разработан в качестве информационной основы для использования полученных данных в комплексе алгоритмов программы автоматизированного проектирования распределения потоков задач. Одной из основных задач справочника является информационная поддержка системы автоматического проектирования в момент интеллектуального принятия решения в блоке балансировки (рис.5) на предложенной авторами модели.

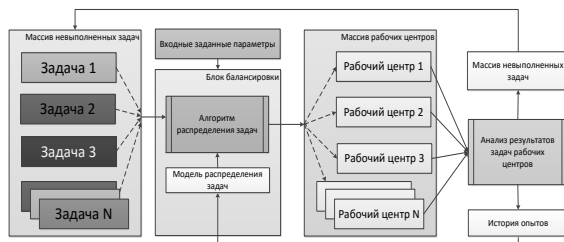


Рисунок 6 – модель распределения потоком задач.

Модель представляет из себя несколько последовательных блоков, на каждом из которых выполняются определенные операции, рассмотрение которых в этой статье не предусмотрено. Однако стоит заострить внимание на блоке балансировки – в нем присутствует «Алгоритм распределения задач», который отвечает за передачу задачи рабочему центру, данные в том числе, которые берутся из словаря ОПП. Словарь определяет только взаимосвязи объектов и позволяет алгоритму выбирать необходимые связи при определении нового объекта из потока невыполненных задач. Описанная технология реализована на базе 1С:Предприятие.

3. Реализация справочника оперативно-производственного планирования на основе онтологического подхода в среде «1С:Предприятие».

Для обеспечения поддержки справочной информацией в программном комплексе реализован блок «справочники» (рис.6), который построен на простой структуре хранения слов разных типов в разных справочниках «1С:Предприятие».

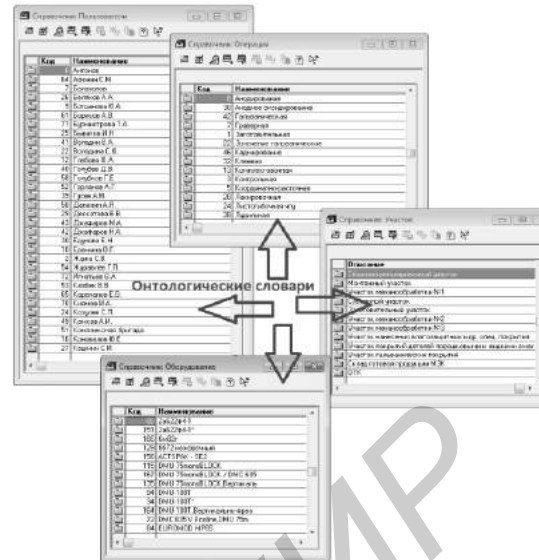


Рисунок 7 -- Графическое представление словарей в виде справочников «1С:Предприятие».

Реализация основного справочника ОПП выполнена в виде словаря (рис.4, рис.8) который построен по смешанному методу, основанному на дедуктивном логическом делении классифицируемого множества. Этим методом достигается конкретизация признаков классификации слов на каждой последующей ступени классификации. В словаре выделяются классификационные группировки (группы понятий) составленные по определенным признакам и слова, являющиеся формальным описанием шаблонов.

Разработка справочника основана на следующих логических правилах:

- на каждой ступени исчерпывается объем делимого множества;
- деление множества слов произведено на каждой ступени группировки по одному и тому же признаку или их сочетанием;
- слова не могут повторяться в рамках группы;
- множество делится последовательно.

N №	Операция, ПП	Треть	Оборудование	Износак
10	50	Контрольная		926
11	55	Обезжиривание	0.5000 Ванна	653
12	60	Фрезерная	0.3000 Печь и 132.00.000 Плита 1	657
13	65	Листогибочная чпу	0.7500 Тумблвант	657
14	70	Слесарная	1.0000 Верстак слесарный	654
15	75	Контрольная		926
16	80	Подготовительная	0.1500	
17	85	Обезжиривание	0.5000 Ванна	653
18	90	Сварочная	0.4860 Плита сварочная цеховая	653
19	95	Сварочная	0.9330	653
20	100	Контрольная		926
21	105	Фрезерная	0.3000 Печь и 132.00.000 Плита 1	657
22	110	Слесарная	1.2000 Верстак слесарный Плита 1	654
23	115	Фрезерная	5.1500 Мл 800р. ланка	657
24	120	Слесарная	5.3330 Рыв сверл. зв. 832 Верстак	654
25	125	Контрольная		926
26	130	Подготовительная	0.1500	
27	135	Сварочная	1.6000	653
28	140	Слесарная	1.0660 Верстак слесарный	653
29	145	Контрольная		926
30	150	Фрезерная	0.3000 Печь и 132.00.000	657
31	155	Слесарная	0.9000 Плита ригельовая	654
32	160	Салт на стл		656

Рисунок 7 – маршрут движения.

Для того, чтобы при распределении слов исчерпывался объем делимого множества, в справочнике предусмотрена классификационная группировка под наименованием "План на 25

декабря». Эти группировки, как правило использованы на для описания состава плана и также имеют свои слова и характеристики (рис.7).

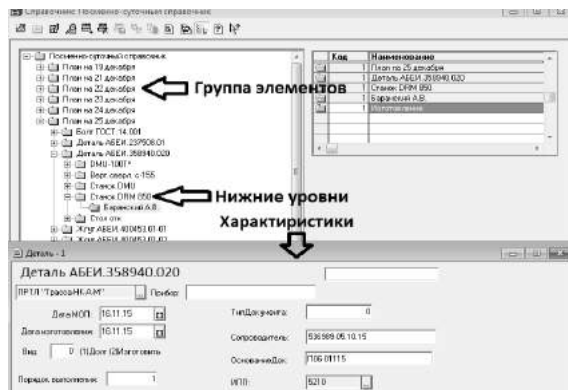


Рисунок 8 – Классификация элементов шаблонов по признакам.

Благодаря разработанному инструментарию в «1С:Предприятие» имеется возможность полного описания необходимых характеристик в онтологии словаря.

Имея в наличии файл с неупорядоченным списком перечня заданий (рис.3) и его составляющие элементы (рис.6) создаются связи для формализации задачи по признакам "Деталь", т.к. в него входят определяющие его элементы, "наследование" т.к., в большинстве своём случаев в словаре у определяемого слова имеется родитель или предок, и заполняются необходимые атрибуты (рис.8).

Заклучение

В статье представлен подход к проектированию системы формирующей посменно-суточные планы производства с учетом выставляемых ограничений. Предлагаемые средства реализованы в среде «1С:предприятие».

Использование разработанных средств открывает возможности для концептуального моделирования имеющихся функций построения планов с возможностью их древовидного представления и аккумуляции в онтологическом словаре.

Представленная материализация может быть использована как объективная информация для диспетчера с целью минимизации ошибок при корректировке планов, а так же позволяет использовать накопленную информацию в алгоритме интеллектуального перераспределения потока задач по рабочим центрам, способствуя значительному повышению уровня автоматизации производственных процессов.

Библиографический список

- [Первин и др., 1973] Первин Ю.А. Португал В. М. Семенов А.И., Планирование мелкосерийного производства в АСУП: информационное и математическое обеспечение, 1973. - М. 29-49
- [Татевосов, 1985] Татевосов К.Г., Основы оперативно-производственного планирования на машиностроительном предприятии: Учеб. Пособие для инженерно-экономических

специальностей вузов. – 2-е изд., 1985. – Л. - 278 с.

[Чудаков и др., 1986] Чудаков А.Д. Фалевич Б. Я., Автоматизированное оперативно-календарное планирование в гибких комплексах механообработки, 1986. – М. - 224 с.

[Соснин, 2007] Соснин П.И. Концептуальное моделирование компьютеризованных систем. – Ульяновск: УлГТУ, 2008. – 198 с.

DEVELOPMENT OF A DICTIONARY OPERATIVE INDUSTRIAL PLANING BASED BY ONTOLOGICAL APPROACH

Karpaev S.A., Larin S.N.

*Ulyanovsk State Technical University,
Ulyanovsk, Russian Federation*

Neonix3000@mail.ru

larinmars@rambler.ru

The necessity of the development and application operative industrial planning based on the ontological approach as an information basis for the creation of complex computer-aided design model the industry.

Introduction

Effectiveness of production depend from plan industry. Realization operative industrial planning to allow increase productivity and to carry out production in precisely in time.

All application operative industrial planning not using selection on demand and peculiarity production industry. In exist application is absent analysis of execution technological operations. Ontological approach to allow create objects and ties for using by algorism of analysis information.

Main Part

Fundamental problem then we creating operative industrial plans it is a calculation all peculiarity of production.

For creating applications need insert all conditions and limitations for ties objects. For creating ties need create dictionary (reference book) based by ontological approach. All information from dictionary transferred to main algorithm.

Algorithm for creating of dictionary realize in «1С:Предприятие» using ontological approach.

Conclusion

The paper presents an approach to the design of the dictionary of operative industrial planning environment «1С:Предприятие».

Using creating application opens possibility for conceptual modeling using functions for creating operative industrial plans .



УДК 65.011.56

АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УПРАВЛЕНИЯ ЭВОЛЮЦИЕЙ НЕЧЕТКОЙ МНОГОСТАДИЙНОЙ ТЕХНОЛОГИЧЕСКОЙ СИСТЕМЫ

Палюх Б.В., Егерова И.А.

Тверской государственной технической университет, г. Тверь, Россия

Pboris@tvstu.ru

Irina.egereva@gmail.com

В работе представлена формальная модель управления эволюцией нечеткой технологической системы, а также структура алгоритмического обеспечения для рассматриваемой задачи.

Ключевые слова: нечеткая система, алгоритмическое обеспечение, эволюция технологической системы, многостадийность.

Введение

Достижение требуемого уровня эффективности производства реализуется за счет совершенствования технологических процессов путем непрерывного контроля основных параметров производства и выработки управленческих решений по эволюции технологической системы.

В нечеткой технологической системе [Щербатов, 2014] процесс управления эволюцией характеризуется рядом особенностей, такими как альтернативность и неопределенность путей достижения цели с высоким риском; невозможность точного планирования, ориентация на прогнозные оценки; противоречивость в сфере экономических отношений и интересов участников процесса. В связи с этим возникает необходимость разработки формализованной модели управления эволюцией нечеткой технологической системы, а также алгоритмического обеспечения для рассматриваемой задачи.

1. Модель управления эволюцией нечеткой многостадийной технологической системы

Рассмотрим многостадийную технологическую систему, состоящую из стадий S_1, S_2, \dots, S_N .

Состояние системы на входе n -ой стадии обозначим через x_n , а на выходе через x_{n+1} . При этом результатные показатели n -ой стадии являются входными для $(n+1)$ -ой стадии. Управление на n -ой стадии будем обозначать через u_{n+1} .

Пусть X_0, \dots, X_N – соответствующие пространства состояний, а U_1, \dots, U_N – пространства управлений. Для определенности будем считать, что все пространства $X_0, \dots, X_N, U_1, \dots, U_N$ – являются компактными метрическими пространствами.

Пусть вход $x_0 \in X$ на первую стадию задан. В результате использования управления $u_1 \in U_1$ на выходе первой стадии формируется состояние $x_1 \in X_1$, заранее не известное. В соответствии с [Дзюба и др., 2014, Беллман и др., 1976] известно, что переменные x_0, u_1, x_1 связаны между собой нечетким отношением S_1 с функцией принадлежности $\mu_{S_1}(x_0, u_1, x_1)$. Будем считать, что цель управления характеризуется нечетким целевым множеством G в пространстве X_N с функцией принадлежности μ_G . Также предположим, что все функции $\mu_G, \mu_{S_1}, \dots, \mu_{S_N}$ непрерывны в их области определения. Задача управления заключается в отыскании последовательности управлений

$$u_1, u_2, \dots, u_N, \quad (1)$$

обеспечивающей наибольшее удовлетворение нечеткой цели G при условии, что начальное состояние x_0 задано.

Таким образом, нечеткое множество G представляет собой цель управления и задача состоит в отыскании последовательности управлений (1), обеспечивающей максимальную степень принадлежности состояния x_0 нечеткому

множеству G при условии, что эволюция технологической системы описывается композицией нечетких множеств S_1, \dots, S_N и G . (см [Дзюба и др., 2014]). Последовательность управлений (1) должна быть подобрана таким образом, чтобы при заданном начальном состоянии x_0 максимально удовлетворялась нечеткая цель G на выходе последней стадии.

Посредством равенства

$$D_{N-n} = S_{N-n} \circ \dots \circ S_N \circ G \quad (2)$$

введем в рассмотрение нечеткое множество D_{N-n} в пространстве X_{N-n} условное по u_{N-n+1}, \dots, u_N . Функция принадлежности $\mu_{D_{N-n}}$ множества D_{N-n} удовлетворяет равенству

$$\mu_{D_{N-n}}(x_{N-n} | u_{N-n+1}, \dots, u_N) = \max_{x_{N-n}, \dots, x_N} \min [\mu_S(x_{N-n}, u_{N-n+1}, x_{N-n+1}), \dots, \mu_G(x_N)]$$

$$\mu_{D_{N-n}}(x_{N-n} | u_{N-n+1}, \dots, u_N) = \max_{x_{N-n}, \dots, x_N} \min \left[\begin{array}{l} \mu_S(x_{N-n}, u_{N-n+1}, x_{N-n+1}), \dots, \\ \mu_S(x_{N-n}, u_N, x_N), \mu_G(x_N) \end{array} \right]$$

Тогда значения $\mu_{D_N}(x_0 | u_1, u_2, \dots, u_N)$ функции μ_{D_N} представляют собой степень принадлежности состояния x_0 множеству G при использовании какой-либо фиксированной последовательности управлений вида (3).

Положим

$$\mu_{N-n}(x_n) = \max_{u_{N-n+1}, \dots, u_N} \mu_{D_{N-n}}(x_{N-n} | u_{N-n+1}, \dots, u_N) \quad (3)$$

При этом, очевидно

$$\mu_N(x_N) = \mu_G(x_N) \quad (4)$$

Более того, легко видеть, что

$$\begin{aligned} \max_{u_{N-n+1}, \dots, u_N} \min \mu_{D_{N-n}}(x_{N-n} | u_{N-n+1}, \dots, u_N) &= \\ \max_{u_{N-n+1}, \dots, u_N} \min [\mu_S(x_{N-n}, u_{N-n+1}, \dots, u_N), & \\ \max_{u_{N-n+2}, \dots, u_N} \mu_{D_{N-n+1}}(x_{N-n+1} | u_{N-n+2}, \dots, u_N)] & \end{aligned}$$

Тогда в силу (3) для произвольного N выполнено равенство

$$\mu_{N-1}(x_{N-1}) = \max_{u_{N-1}, x_N} \min \left[\begin{array}{l} \mu_S(x_{N-1}, u_{N-1}, x_N), \\ \mu_{N-n+1}(x_{N-n+1}) \end{array} \right], \quad (5)$$

где $\mu_{N-n}(x_{N-n})$ - максимальная степень принадлежности состояния x_{N-n} множеству G .

Управление (5) с граничным условием (4) представляет собой аналог уравнения Беллмана для рассматриваемой задачи.

Процедура построения оптимального управления эволюцией нечеткой технологической системой распадается на два уровня.

Входными данными первого уровня являются функции принадлежности $\mu_{S_1}, \dots, \mu_{S_N}, \mu_G$. На данном уровне из решения уравнения (4) определяются оптимальные управления u_1^*, \dots, u_N^* . При этом согласно основным принципам теории динамического программирования управления u_1^*, \dots, u_N^* всегда можно построить в виде законов управления с обратной связью, т.е.

$$u_n^* = u_n^*(x_{n-1}), n = 1, \dots, N. \quad (6)$$

На втором этапе должна осуществляться реализация найденных законов управления на рассматриваемой технологической системе. В результате получается замкнутая система управления эволюцией системы, представленная на рисунке 1

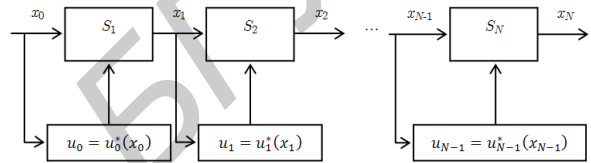


Рисунок 1 – Структура оптимального управления

Таким образом, моделирование нечетких многостадийных технологических систем при помощи нечетких отношений и правила композиции позволяет осуществить синтез управления с обратной связью рассматриваемыми системами.

2. Алгоритмическое обеспечение

Разработка алгоритмического обеспечения управления эволюцией технологической системы включает в себя процесс создания общего алгоритма, состоящего из следующих разделов:

- описание основных задач функционирования технологической системы. В данном разделе описываются обеспечивающие подсистемы, такие как средства производства, ресурсы, исполнители; технологические средства, такие как техническое оснащение, специфические средства; средства обеспечения качества функционирования технологической системы и другие;

- описание вычислительного процесса при решении задач обработки информации и управления аппаратурой, оборудованием, системой контрольно-измерительных приборов. Здесь представляются элементы структуры технологической системы и взаимосвязь между ними.

- описание задач управления эволюцией системы. Раздел содержит описание основных способов управления эволюцией технологической системой:

- на основе управления ростом потенциальных возможностей персонала (например, новые методики мотивационного управления) и формировании персоналом информации о своих возможностях (изучение опыта других аналогичных предприятий; повышение квалификации, изучение новой профессиональной литературы и т.д.);

- управления ростом предельных технологических возможностей (например, внедрение новых способов и возможностей эксплуатации оборудования, технологических узлов; модернизация имеющегося оборудования; корректировка технологических стадий и т.д. [Paliukh и др., 2014]);

- при выработке производственного плана (например, внедрение решений, влияющих на ограничения).

Общее описание содержит перечень алгоритмов и последовательность их применения, условия, при которых используется каждый алгоритм, связи алгоритмов по выходной и входной информации и т.п.

3. Структура алгоритма

При разработке алгоритма управления эволюцией технологической системой необходимо учитывать общую структуру типового программного комплекса, включающего в себя, как правило, несколько компонент и комплексов программ.

Так, структура программного комплекса управления эволюцией нечеткой многостадийной непрерывной технологической системы может включать:

- компоненты первичной обработки нечеткой информации;
- компоненты вторичной обработки нечеткой информации;
- комплекс организации обмена информацией с аппаратурой, системой контрольно-измерительных приборов;
- комплекс контроля технического состояния оборудования;
- комплекс планирования работ, включающий в себя компоненты формирования временной диаграммы функционирования управляющей системы;
- комплекс анализа, моделирования и планирования;
- комплекс поиска инновационных решений по управлению эволюцией промышленной системы;
- комплекс контроля результатов производства (экономическая составляющая, соответствие качества продукции заданным характеристиками и т.п.)

Рассмотрим схему обработки информации и управления эволюцией нечеткой многостадийной непрерывной технологической системой (рисунок 2).

Рассматриваемая система состоит из стадий S_1, S_2, \dots, S_N . Состояние процесса на входе n -ой стадии обозначен через x_n , а на выходе через x_{n+1} .

Получение A_1, A_2, \dots, A_N – первый этап сбора информации A . В результате функционирования каждой стадии формируется множество первичной разнородной информации, связанной как с протеканием технологического процесса, так и с работой обеспечивающих подсистем. B_1, B_2, \dots, B_N – на втором этапе сбора информации B данные, полученные на этапе A , классифицируются по группам характеристик и времени получения информации.

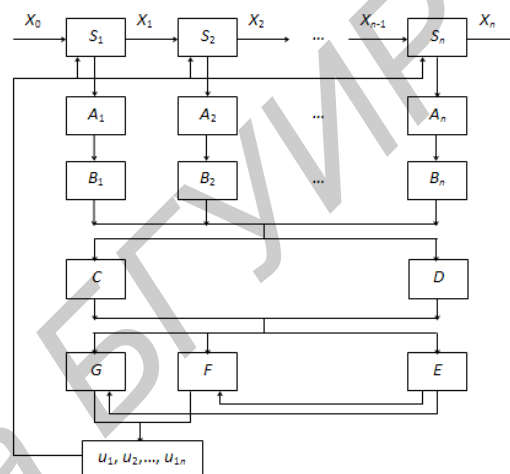


Рисунок 2 – Схема обработки информации и управления эволюцией нечеткой многостадийной непрерывной технологической системой

C – на основе контроля и анализа множества данных, полученных на этапе B , формируется заключение о моральном и техническом состоянии аппаратуры, технологического оборудования, системы контрольно-измерительных приборов путем сравнения фактических данных с лучшими характеристиками аналогичных производств.

D – на данном этапе, одновременно с контролем технического состояния системы осуществляется контроль результатов качества каждой стадии технологической системы.

Следующий этап алгоритма управления эволюцией нечеткой технологической системы включает в себя блоки: E, F, G .

E – поиск инновационных решений для управления эволюцией нечеткой системы. Данный этап является одним из самых важных при управлении технологической системой. В настоящее время существует огромное количество вспомогательной информации, размещенной в хранилищах данных, информационных распределенных систем, содержащих решения по различным проблемам функционирования технологических систем, а также литературы по соответствующей тематике [Иванов и др., 2013]. Кроме того, любая технологическая система содержит интеллектуальный ресурс, который можно использовать при грамотном построении

мотивационного управления. Перед руководителями стоит задача поиска, сбора, обобщения и синтеза управленческих решений, предварительного анализа результатов инновационных предложений с целью повышения эффективности функционирования системы в целом.

F – планирование функционирования систем получения и обработки первичной информации, обработки заявок обслуживающих систем. В соответствии с технологическим регламентом формируется график получения и дальнейшей обработки первичной информации. В случае необходимости осуществляется корректировка данного графика.

В процессе наблюдения за производственной системой наряду с решением основных задач необходимо решать дополнительные задачи, связанные с поддержанием работоспособности системы.

G – этап анализа, моделирования и прогнозирования. На данном этапе решается ряд задач. На основе полученной информации на этапах алгоритма управления *C* и *D*, а также в соответствии с множеством заданных пороговых значений отклонений технологических параметров аналитическим службам предприятия необходимо своевременно сообщать о состоянии безопасного и надежного функционирования технологической системы, а также оценивать соответствие качества продукции заданным в спецификациях требованиям. При выработке управленческих решений на этапе поиска инновационных предложений, необходимо организовать проверку обоснованности и целесообразности внедрения инноваций в технологический процесс.

Заключение

В работе представлена модель управления эволюцией нечеткой технологической многостадийной системы, в процессе разработки которой сформулирована задача синтеза оптимального управления абстрактными нечеткими многостадийными технологическими системами. Для представленной модели описан алгоритм управления эволюцией нечеткой технологической системой, состоящий из этапов сбора, группировки первичной информации, контроля предусмотренных технологическим регламентом параметров, а также этапов формирования и апробации управляющих инновационных решений, направленных на повышение эффективности функционирования технологической системы в целом.

Библиографический список

[Щербатов, 2014] Щербатов, И.А. Сложные слабоформализуемые системы: компонентный подход // Приоритетные научные направления: от теории к практике: сборник материалов XIII Международной научно-практической конференции / Под общ. ред. С.С. Чернова.- Новосибирск: Издательство ЦРНС, 2014. - С. 104-108.

[Дзюба, 2014] Дзюба, С.М., Палюх, Б.В., Егерева, И.А. Об оптимальном управлении нечеткими многостадийными

процессами // XII Всероссийское совещание по проблемам управления, М.: ИПУ РАН. 2014. – С.3968-3972.

[Беллман, 1976] Беллман, Р., Заде, Л. Принятие решений в расплывчатых условиях // Вопросы анализа и процедуры принятия решений. М.: Мир, 1976. С. 172-216.

[Paliukh, 2014] Paliukh, B.V., Vinogradov, G.P., Egereva, I.A. Managing the Evolution of Chemical Engineering System // Theor. Found. Chem. Eng. 2014. Vol.48, №3. – pp. 325-331].

[Иванов, 2013] Иванов, В.К., Виногорова, Н.В. Эвристический алгоритм фильтрации и семантического ранжирования результатов поиска документов // Вестник Тверского государственного университета. Серия: Прикладная математика. 2013. №3 (30). С.97-106.

ALGORITHMIC SUPPORT TO MANAGING FUZZY MULTI-STAGE TECHNOLOGICAL SYSTEM EVOLUTION

Palyukh B.V., Egereva I.A.

Tver State Technical University, Tver, Russia

Pboris@tvstu.ru

Irina.egereva@gmail.com

This work presents a formal management model for fuzzy technological system evolution, as well as the structure of algorithmic support to the problem addressed.

Introduction

In a fuzzy technological system, process of evolution management has its own features, such as: the ways to achieve high-risk goals are alternative and vague; precise planning is impossible, and we have to be guided by predictive estimate; parties involved in the process may have contradictory economic relations and interests. Therefore, there is a need to design a formalised management model for fuzzy technological system evolution, as well as for algorithmic support to the problem addressed.

Main Part

Achieving target production efficiency level is made possible by developing technological processes through continuous control of main production metrics and generation of management decisions for technological systems evolution. This work presents a management model for fuzzy multi-stage technological system evolution, based on algorithmic support to managing fuzzy multi-stage technological system evolution.

Conclusion

This work presents a management model for fuzzy multi-stage technological system evolution; while developing the model, the problem of generating optimal control over abstract fuzzy multi-stage technological systems has been formulated. For the model offered herein, we have described the management algorithm for fuzzy technological systems evolution, comprising the following stages: initial data acquisition, control over process operating procedure parameters, as well as generating and evaluating innovative management decisions, aimed at improving the efficiency of a technological system as a whole.



УДК 658:005.05: 519.859

МОДЕЛИ ИНТЕЛЛЕКТУАЛЬНОЙ АСУТП ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ

Бурдо Г.Б. *, Федотова А.В. **

**Тверской государственной технической университет
г. Тверь, Россия,
gbtms@yandex.ru*

***Московский государственный технический университет им. Н.Э. Баумана
г. Москва, Россия*

Приведена иерархическая структура интеллектуальной АСУТП, отражающая структуру процессов планирования и управления производственными подразделениями. Показаны основные подсистемы АСУТП. Приведены модели и алгоритмы работы подсистем объёмного, календарного и оперативного планирования, и диспетчирования. Охарактеризованы особенности предлагаемой АСУТП, определяющие её малую инерционность. Показаны способы формирования приоритетных схем обслуживания объектов, обеспечивающие возможность выбора оптимального в соответствии с установленным критерием.

Ключевые слова: автоматизированные системы управления технологическими процессами, планирование, диспетчирование, наукоемкая продукция, искусственный интеллект, поддержка решений.

Введение

Известно, что применение автоматизированных систем управления (АСУТП) технологическими процессами в многономенклатурном машиностроении затрудняется необходимостью корректировки планов всех уровней вследствие динамичной производственной ситуации и постоянным появлением новых контрактов.

Второй важной причиной, является проведение упрощенной технологической подготовки производства, при которой детально не расписываются технологические операции, и как следствие, имеются приближенные времена их выполнения.

Ситуация усугубляется для технологических процессов (ТП) с большой долей ручного труда, что приводит еще и к неопределенности времени выполнения неповторяющихся технологических операций.

К таким технологическим процессам, в частности, относятся сборка изделий, техническое обслуживание изделий и их ремонт, изготовление новых образцов наукоемкой продукции.

Ниже рассмотрена АСУТП на примере технического обслуживания авиационной техники,

позволяющая минимизировать воздействие перечисленных выше негативных факторов. В [Бурдо, 2010], [Бурдо, 2011, а] были представлены общие принципы создания АСУТП.

Структура автоматизированной системы управления процессами обслуживания авиационной техники, соответствующая им, показана на рисунке 1.

Иерархия подсистем АСУТП в силу системного принципа преемственности соответствует структуре производственных планов авиаремонтных предприятий, обратные связи между подсистемами обеспечивают возможность корректировки планов всех уровней, что обеспечивает их выполнимость.

1. Подсистемы АСУТП

1.1. Подсистема объёмного планирования

Подсистемой объёмного планирования синтезируются годовые планы работы организации без учета календарного времени выполнения каждого из контрактов.

Алгоритм работы подсистемы объёмного планирования (*ОбП*) итерационный (что соответствует логике человеческого мышления), и представлен следующими шагами:

На первом шаге оцениваются годовые мощности производственных подразделений (*ПП*),

выполняющих техническое обслуживание (ТО) авиатехники. Должно быть:

$$\sum_{i=1}^n (S_i \cdot T_i) \leq 0,8 \cdot S_{ПП} \cdot F_r \cdot N, \quad (1)$$

где S_i – площадь, занимаемая i -м обслуживаемым объектом; n – их количество; T_i – трудоемкости обслуживания i -го объекта, час; $S_{ПП}$ – площадь производственных подразделений; F_r – номинальные годовые фонды времени работы ПП при односменной работе; N – число смен; 0,8 – коэффициент, учитывающий различие между трудоемкостью ТО и его циклом и необходимость наличия резервов.

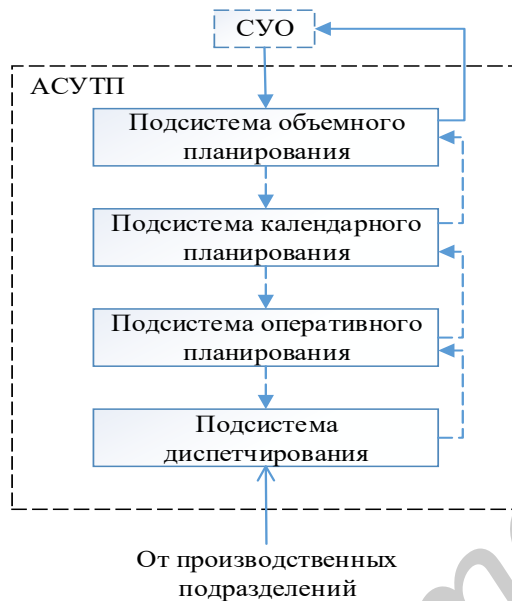


Рисунок 1 - Структура АСУТП: СУО – система управления организацией

$$T_i = \left(\sum_{j=1}^m t_{um.kj} \right)_i - \sum_{k=1}^c t_{um.k}^c \quad (2)$$

где $t_{um.kj}$ – штучно-калькуляционное время выполнения j -ой операции над i -м объектом; $\sum_{k=1}^c t_{um.k}^c$ – суммарное время совмещенных (параллельно-выполняемых во времени) операций ТО над i -м объектом, m – число операций.

Если неравенство (1) выполняется, то переходят к шагу 4 планирования, иначе переход к шагу 2.

На втором шаге итерационно выполняется корректировка ОБП и его согласование с СУО. Возможны следующие варианты. Первый – удаление из плана часто обслуживаемых изделий до выполнения формулы (1) по следующим приоритетным правилам.

Смысл всех приведенных ниже приоритетных правил направлен на попытку оставить в плане

максимальное число контрактов, то есть обеспечить организации получение максимального дохода.

Правило 1. Из плана следует исключить наименьшее число изделий, обеспечивающих выполнение условия (1).

Правило 2. При возможных альтернативных вариантах (при различии между производением $S_i \cdot T_i$ для исключаемых из плана объектов до 10%) из плана ТО исключают объект, имеющий большее значение S_i .

Правило 3. Если различие между S_i для исключаемых объектов до 10%, то из плана ТО исключают объект, имеющий меньшую договорную цену обслуживания.

Правило 4. Из плана не исключаются объекты, ТО которых безоговорочно должно быть выполнено (приоритет организации).

Выявленные правила исключения отражают смысл работы организации и ПП: получить максимальный доход и создание предпосылок эффективной работы.

Если удается полностью откорректировать ОБП, то переход к шагу 4, иначе к шагу 3.

На третьем шаге выполняется попытка удовлетворить неравенства (1) путем увеличения фонда времени работы ПП за счет K дополнительных смен в течении года. Следует отметить, что данный шаг – прерогатива СУО, определяющей допустимое число дополнительных смен K_d .

Должно выполняться:

$$K_d \geq \frac{\sum_{i=1}^n (S_i \cdot T_i) - 0,8 \cdot S_{ПП} \cdot F_r}{8 \cdot S_{ПП}} = K, \quad (3)$$

где 8 – длительность смены в часах.

Если условие (3) удается удовлетворить, то переход к шагу 4, иначе возврат на шаг 2.

На четвертом шаге проводится проверка выполнимости работ исходя из общих фондов времени работы ремонтников, и по видам работ (при специализации видов работ за отдельными исполнителями).

Должны выполняться следующие условия:

По общему фонду времени работ:

$$\sum_{i=1}^n T_i \leq P \cdot F_p \cdot 0,8 \cdot N, \quad (4)$$

где P – общая численность рабочих, выполняющих техническое обслуживание; F_p – годовое время рабочего, час.

По фондам времени по отдельным видам работ для каждого из них m должно выполняться, $m = \overline{1, B}$:

$$\sum_{i=1}^n \left(\sum_{m=1}^1 t_{um.k_m} \right)_i \leq P_m \cdot F_p \cdot 0,8, \quad (5)$$

где P_m – общее число рабочих, выполняющих работу вида m , $m = \overline{1, B}$.

Если рабочие, выполняющие ТО, специализируются на нескольких видах работ m ($m=1, p$), то должно выполняться условие:

$$\sum_{i=1}^n \left(\sum_{m=1}^p t_{um.k_m} \right)_i \leq P_m \cdot F_p \cdot 0,8. \quad (6)$$

Если работа выполняется бригадами, то P_m – общее число бригад. Если условия (4) – (6) выполняются, то переход к подсистеме календарного планирования, иначе на шаг 5.

На пятом шаге проводится попытка выполнить неравенства (4) – (6) за счет введения дополнительных смен, при этом их число не может быть более K_d (см. описание шага 3).

Для случая (4):

$$K_d \geq \left(\sum_{i=1}^n T_i - P \cdot F_p \cdot 0,8 \right) / 8 = K. \quad (7)$$

Для случая (5):

$$K_d \geq \sum_{i=1}^n \left(\sum_{m=1}^1 t_{um.k_m} \right)_i - P_m \cdot F_p \cdot 0,8 / 8 = K. \quad (8)$$

Для случая (6):

$$K_d \geq \sum_{i=1}^n \left(\sum_{m=1}^p t_{um.k_m} \right)_i - P_m \cdot F_p \cdot 0,8 / 8 = K. \quad (9)$$

Для дальнейших расчетов принимается максимальное значение K , а если выполнялся шаг 3, то наибольшее значение (с учетом и полученного по формуле (3)).

Принципиально, возможно и увеличение суммарных фондов времени за счет увеличения численности работающих P и P_m . Однако, этот шаг в условиях рассматриваемых производственных систем не всегда реализуем из-за дефицита квалифицированных рабочих данного профиля и трудности прогнозирования последующих заказов.

Если условие выполняется, то переходят к подсистеме календарного планирования, иначе к шагу 6.

На шестом шаге производится исключение работ из годового объемного плана по следующим приоритетным правилам.

Правило 1. Из плана не исключается изделия, ТО которых должно быть безоговорочно выполнено.

Правило 2. Из плана следует исключать наименьшее число изделий.

Правило 3. Исключением из плана изделий сначала обеспечивают выполнение неравенства (9), затем (8) и в последнюю очередь – (7).

Правило 4. При возможных альтернативах (при различии в числе K не более 10%), исключают из плана объект, имеющий меньшую договорную цену обслуживания.

Возврат к шагу 4.

Указанные правила обеспечивают достаточно «точное» формирование ОбП и получение максимального дохода организации.

В целом, особенностью работы рассматриваемой подсистемы является достаточно высокая формальность правил выработки советующих управляющих воздействий, ясность процедур синтеза решений и проверка возможностей производственных подразделений не только по мощностям, но и по фондам времени работы рабочих.

Хотелось бы отметить и возможность использования на практике технического обслуживания такой возможности, как освоение работниками смежных видов работ, что создает высокую гибкость производственных систем (рисунок 2).

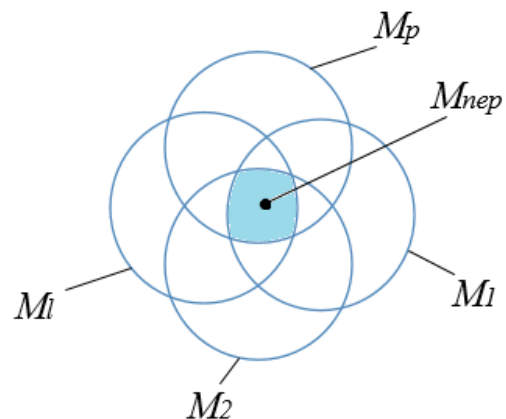


Рисунок 2 - Множества $M_1, M_2, \dots, M_1, \dots, M_p$ видов работ по обслуживанию, выполняемых рабочими (бригадами) рабочих

В пределе, когда каждый рабочий (бригада) может выполнять любую работу, необходима проверка на шаге 4 лишь по формуле (4), а на шаге 5 – по (7).

Также будут упрощены и задачи, решаемые при календарном и оперативном планировании вследствие уменьшения числа анализируемых

вариантов, повысится производительность труда по причине сокращения времени перехода рабочих от одного к другому обслуживаемому объекту.

Поэтому, целевой функцией при подборе и обучении рабочих производственных подразделений может являться мощность пересечения

множеств видов работ, выполняемых рабочими (бригадами): $M_{пер} \rightarrow max$.

1.2. Подсистема календарного планирования

Работа подсистемы календарного планирования также осуществляется на ряде шагов.

На первом шаге выполняется предварительный набор заказов для первого календарного плана (обычно квартального). Формирование осуществляется исходя из более ранних сроков окончания работ по обслуживанию объектов и приоритетности заказов.

Набор заказов заканчивается при невозможности удовлетворить неравенство (1) при присоединении еще одного любого заказа. В формуле вместо значения F_2 подставляется $F_{кв}$ (квартальные фонд времени работы).

На втором шаге рассчитываются длительности технологических циклов выполнения работ по каждому i -му заказу:

$$C_i = T_i + \Pi_i \cdot T_{n.ср} + t_{он} \quad (10)$$

где Π_i – ожидаемое количество перерывов (ожиданий) при обслуживании i -го объекта; $T_{n.ср}$ – среднее время перерыва, Π_i и $T_{n.ср}$ определяются из опыта выполнения обслуживания; $t_{он}$ – среднее время ожидания начала обслуживания, тратится на высвобождение рабочего места, установку нового объекта и подготовку к обслуживанию; T_i – см. расшифровку к формулам (1) и (2).

На третьем шаге выполняется размещение объектов по календарным срокам выполнения работ. При этом объекты делятся на 2 группы: 1-я – объекты с заданными предельными сроками; 2-я – объекты с окончанием работ в пределах всего календарного периода (рисунок 3).

На рисунке: $S_1, S_2, S_{ин}$ – площади, занимаемые обслуживаемыми объектами, S_{III} – суммарная площадь производственных подразделений; C_1, C_2, C_m – технологические циклы обслуживания объектов; $t_{ок1}, t_{ок2}$ – предельные сроки обслуживания 1-го и 2-го объектов. Как можно увидеть, для объектов 1-ой предусмотрен резерв времени.

К примеру, i -ый объект может быть обслужен в пределах всего календарного срока, его фактический цикл $C_i^Ф$ разбивается на две части, при этом он больше C_i (см. формулу (10)) на величину $t_{ож}$. Такой подход обеспечивает более полное использование мощностей предприятия.

Для построения пространственно-временной структуры загрузки производственных подразделений используются алгоритмы, аналогичные алгоритмам листового раскроя материалов, на которые накладываются ограничения (в данном случае S_i и C_i ассоциируются с длиной и шириной заготовки, S_{III} и F_k – ширина листа) следующего вида.

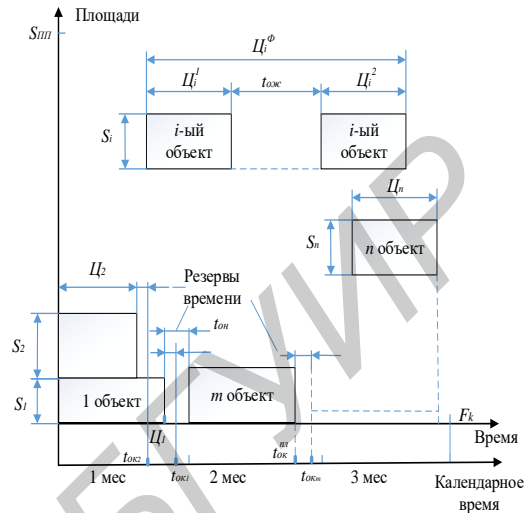


Рисунок 3 - Размещение обслуживаемых объектов в пространстве и во времени: $t_{ок,m}^m, t_{ок}^m$ - директивные и планируемые сроки окончания работ

1. Если m -ый объект имеет оговоренные сроки завершения обслуживания $t_{ок,m}$, то оно должно быть завершено с опережением на l дней (обычно $l = 2-3$ дня), т.е. $t_{ок}^m - t_{ок,m} = 2-3$ дня.

2. Площадь, не закрытая объектами, не должна превышать 10% от общей.

3. Если условие 2 не удовлетворяется и остаются необслуженные объекты, то для объектов 2-ой группы выполняется разделение технологического цикла на этапы по следующим правилам.

Правило 1. Первыми разделение выполняют для объектов с наиболее длительным циклом.

Правило 2. Первым заполняется свободное рабочее место подходящей площади, расположенное наиболее близко к началу календарного периода (т.е. слева на рис.3).

Правило 3. Разделение цикла выполняют с обеспечением целостности операций (см. формулу (2) до приближения выделенной части цикла к длительности незанятости рабочего места.

Далее алгоритм распределения запускают вновь.

3. Если условие 2 удовлетворяется после разбиения циклов, и имеются необслуженные объекты, то переход к шагу 4.

4. Если условие 2 удовлетворяется, и не имеется необслуживаемых объектов, то переход к шагу 5.

5. Если условие 2 удовлетворяется и не имеется необслуженных объектов, то переход к шагу 6.

На четвертом шаге для обеспечения выполнимости планов анализируется возможность введения дополнительных смен, т.е. увеличение величины F_k (см. формулу (3), K_d берется в расчете на квартал). При отсутствии возможности заказы переносятся на следующий квартал. Переход к шагу 6.

На пятом шаге анализируется возможность обслуживания объектов из следующего календарного периода в данном.

Эти объекты относят ко 2-ой группе в данном календарном периоде. В этом случае из следующего календарного периода берутся объекты с более ранними сроками выполнения и возвращаются на шаг 2.

При корректировке пространственно-временной структуры на шаге 3 ранее запланированные объекты не трогаются, заполняются лишь свободные площади.

На шестом шаге выполняется проверка по фондам времени рабочих, выполняющих техническое обслуживание по каждому месяцу календарного плана.

Проверка выполняется по формулам, аналогичным (5) и (6), вместо расчетного годового времени подставляется фонд времени F_M (фонд времени за месяц периода календарного планирования).

Если планы выполнимы для всех месяцев, то переходят к подсистеме оперативного планирования, в противном случае выполняют попытки создать дополнительное число рабочих смен (см. формулы (8) и (9), K_d берется в расчете на месяц).

Если данное мероприятие удастся реализовать, то переходят к подсистеме оперативного планирования, иначе – к шагу 7.

На седьмом шаге выполняется выделение из месячного календарного плана обслуживания объектов до выполнения условия, оговоренного на предыдущем шаге, по следующим приоритетным правилам, и перенос их обслуживания на следующий месяц.

Правило 1. Из плана не выделяются объекты 1-й группы (с жестко оговоренным сроком окончания обслуживания).

Правило 2. Выделение начинают с объекта, имеющего минимальную трудоемкость обслуживания.

Правило 3. Если невозможно удовлетворить условиям (см. шаг 6) выделением одного объекта, то переходят к выделению второго и т.д. При удовлетворении условий - возврат на шаг 3.

1.3. Подсистема оперативного планирования

Синтезированные месячные календарные планы работ служат основой для оперативного планирования и управления работой

технологического подразделения, выполняющего техническое обслуживание.

С этой целью покажем формальную постановку задачи составления календарного плана-графика (расписания) работы подразделения, в котором выполняются техническое обслуживание. Постановка сделана на основе работ [Бурдо, 2011, а], [Прилуцкий и др., 2008], [Смоляр, 1988], [Прилуцкий, 2007].

Календарные планы-графики (КПГ) разрабатываются на интервалы оперативного планирования (ИОП), равные 5 календарным дням. Таким образом, месячный календарный план работ должен быть представлен, как совокупность КПГ.

Пусть задано подразделение, выполняющее ТО с установившимся режимом работы. В подразделении должно быть выполнено ТО некоторого множества объектов $O = \{O_i\}$, $i = \overline{1, d}$; d – количество объектов.

По каждому из объектов должна быть выполнена процедура PP_i . Каждая процедура PP_i состоит из множества операций $\{OP_{ij}\}$, $j = \overline{1, m_i}$, где m_i – количество операций обслуживания i -го объекта.

Каждой операции OP_{ij} присваивается индекс k из множества $\{1, 2, \dots, l\}$, характеризующий вид выполняемой работы в операции (вид операции), т.е. $PP_i = \{OP_{ijk}\}$.

Каждая операция характеризуется трудоемкостью ее выполнения, равной штучно-калькуляционному времени ее выполнения $T_{ij} = t_{ум.к.ij}$.

Обслуживание выполняется множеством R бригад рабочих, $R = \{r_m\}$; $m = \overline{1, c}$, где c – количество бригад. Каждая бригада r_m выполняет p операций OP_m^r определенного вида.

$$OP_m^r = \{OP_{mn}^r\}; m = \overline{1, c}; n = \overline{1, l^r}, \quad (11)$$

где l^r – множество видов операций, выполняемых бригадой.

Должно быть для обеспечения выполнимости операций $\{OP_{ijk}\}$ выдержано условие, означающее, что все виды операций могут быть выполнены исполнителями:

$$[G = U(1, 2, \dots, l)] \subset G^r = \cup(1, 2, \dots, l^r). \quad (12)$$

Порядок следования операций $\{OP_{ijk}\}$ в процедуре PP_i задается следующим образом: Задаются булевы переменные α и β . Для $\{OP_{ijk}\}$:

$$\alpha_j = \begin{cases} 0, & \text{если операция независима и может выполняться} \\ & \text{в любом месте процедуры, в т.ч. параллельно с другими;} \\ 1, & \text{если порядок следования оговаривается.} \end{cases}$$

Для двух операций j' и j'' с $\alpha_{j'}=1$ и $\alpha_{j''}=1$ значение

$$\beta_{j',j''} = \begin{cases} 0, & \text{если операции могут выполняться параллельно,} \\ 1, & \text{если операция } j' \text{ предшествует операции } j'' \\ & (\text{операции несовместимы во времени}). \end{cases}$$

Ограничения на формируемый КПП, следующие:

1. Непрерывность выполнения работы над объектом. Для этой цели времена начала t_{ij}^o и окончания t_{ij}^k работы на ОП_{ijk} связываются соотношением: $t_{ij}^k = t_{ij}^o + T_{ij}$.

2. Невозможность параллельного выполнения на одном объекте несовместимых работ, т.е. для них условие: $t_{ij''}^k \geq t_{ij'}^k$, операция j' предшествует j'' , (операции не совместимы). Это новое ограничение по сравнению с известным.

3. Невозможность проведения одновременного числа операций большего, чем число бригад, т.е. для любого множества операций $\{ОП_{ij}\} = \{ОП_{ij}, \dots, ОП_{ij'}\}$ условие $t_{ij}^o \leq t_{ij'}^o \leq t_{ij}^k$ не может выполняться более c раз (c – число бригад).

4. Невозможность одновременного проведения операций одинаковых видов большего, чем число специализированных для этих операций бригад. Для учета этого ограничения введем переменную:

$$X_{ijl} = \begin{cases} 1, & \text{если вид технологической операции ОП}_{ijk} \\ & \text{совпадает с видом операции,} \\ & \text{выполняемых рабочими ОП}_{mn}^l, \text{ т.е. } k=n, \\ 0, & \text{в противном случае.} \end{cases}$$

В любой момент времени работы подразделения $t_{ij}^o \leq t \leq t_{ij}^k \sum X_{ijl} \leq B$, где B – число бригад, которые могут выполнять работу группы n .

5. Учитывая, что, как правило, реальная производственная ситуация, складывающаяся в ПП, динамична и не всегда формальна, введем ряд ограничений – приоритетов, отражающих текущее состояние подразделений и директивы управленческого персонала.

В общем виде его можно сформулировать так: на объекте с более высоким приоритетом в первую очередь выполняется операция свободившейся бригадой, т.е. $t_{ijk}^o < t_{i'jk}^o$, если приоритет объекта i выше i' . Это новое ограничение по сравнению с известными постановками.

Применение приоритетов во многом упрощает алгоритмы построения КПП.

Требуется построить $КПП = \{t_{ijk}^o\}$, отвечающий ограничениям, перечисленным выше. В качестве целевой функции выступает общий минимальный цикл технического обслуживания всех объектов, T_o :

$$T_o = \max_{ijk} \{t_{ijk}^o\} \rightarrow \min. \quad (13)$$

Генерация КПП выполняется на основе известного аналитико-приоритетного метода с использованием пошаговых процедур (имитации) выполнения операций ТО над объектами [Прилуцкий, 2007], [Прилуцкий и др., 1997], [Султан-Заде и др., 1996], [Танаев и др., 1975], [Блехерман, 1979].

В качестве правил, определяющих приоритеты обслуживаемых объектов, могут быть следующие [Бурдо, 2011,b]:

1. Установление приоритетов объектов руководителем подразделения. В этом случае каждому объекту присваивается ранг, скажем цифрами от 1 до d , где d – общее число объектов.

Для получения вариантов КПП, и учитывая, что на объекте может быть выполнено одной и той же бригадой несколько работ, то следует в одном варианте предусмотреть дополнительный приоритет выполнения операции максимальной трудоемкости, а во втором – минимальной.

2. Установление приоритетов нескольким объектам, остальные имеют приоритеты ниже первых. Для них приоритеты могут быть установлены следующим образом, определяющим два варианта возможных КПП.

2.1. Первым освободившейся бригадой обслуживается объект с минимальной остаточной суммарной трудоемкостью, т.е. этот объект имеет максимальный приоритет. Этот приоритет исходит из посылки, что лучше быстрее заканчивать ТО тех объектов, которые имеют меньший объем остаточных работ.

2.2. Второй приоритет имеет противоположный смысл, чтобы охватить две крайние ситуации. Его смысл – выровнять объекты проведенного ТО по всем объектам: первым освободившейся бригадой выполняется обслуживание объекта с максимальной остаточной суммарной трудоемкостью операций.

Учитывая, что по наиболее приоритетному объекту может быть выполнено несколько операций, то смыслу приоритетов будет отвечать выбор операции максимальной трудоемкости.

3. Если приоритеты обслуживания не назначаются руководством подразделения, то по вышеприведенным правилам приоритеты назначаются для всех объектов.

1.4. Подсистема диспетчирования

Диспетчирование работ осуществляется путем контроля за ходом технического обслуживания объектов сравнением плановых и фактических времен окончания операций, т.е. при этом анализируются следующие данные:

1. Отставания (опережение) выполнения КПП по подразделению, час;
2. Изменения отставания (опережения) КПП в данном ИОП по сравнению с предыдущим (т.е. тенденция изменения), час/ смена;
3. Фактические и действительные фонды времени выполняющих обслуживание рабочих (бригад);
4. Расчетные нормы и фактические времена выполнения операций обслуживания, час;
5. Расчетный и фактический коэффициенты параллельного по времени выполнения операций, %.

Продукционными моделями (в силу громоздкости не приводятся) на основе указанных данных вырабатываются варианты управленческих решений по приведению фактических КПП к расчетным, в т.ч. и пересчет ранее сформированных КПП и календарных планов.

Заключение

Предложенная АСУТП разрабатывалась, как «советующая», помогающая оперативно принимать решения на всех уровнях планирования, и имеющая элементы искусственного интеллекта.

Её первой отличительной особенностью от известных систем класса «Scada» и ERP [Балахонов и др., 2006] является реализация оперативной связи между планами всех уровней и возможность их комплексного пересчета при появлении новых заказов (заключение контрактов в течении года) с различными приоритетами.

Второй чертой - высокая степень формальности процедур, и как следствие, точность рекомендуемых решений.

Третьей – возможность обучения путем анализа расчетных норм и фактических времен выполнения операций, и расчетного и фактического коэффициентов параллельного по времени выполнения операций.

При доработке, АСУТП может быть реализована в качестве системы поддержки решений на различных предприятиях многоименчатого машиностроения, занимающихся обслуживанием, ремонтом и сборкой наукоемких изделий.

Работа выполнена при финансовой поддержке РФФИ, проект № 14-01-00324.

Библиографический список

- [Бурдо, 2010] Бурдо, Г.Б. Принципы построения автоматизированной системы управления технологическими процессами в многоименчатурных производствах / Г.Б. Бурдо // Вестник Саратовского государственного технического университета. - 2010. - №3(48). - С.113-118.
- [Бурдо, 2011, а] Бурдо, Г.Б. Интеллектуальные процедуры планирования и управления в производственных системах геофизического приборостроения/ Г.Б. Бурдо // Программные продукты и системы. - 2011. - №3(95). - С.107-110.
- [Прилуцкий и др., 2008] Прилуцкий, М.Х., Власов, В.С. Метод ветвей и границ с эвристическими оценками для конвейерной задачи теории расписаний/ М.Х. Прилуцкий, В.С. Власов // Вестник Нижегородского университета им. Н.И. Лобачевского. 2008. - №3. - С.143-157.
- [Смоляр, 1988] Смоляр Л.И. Модели оперативного планирования в дискретном производстве. М.: Наука, 1988. - 320 с.
- [Бурдо, 2011, б] Бурдо, Г.Б. Оперативное планирование на основе приоритетных схем в производственных системах геофизического приборостроения/ Г.Б. Бурдо // Каротажник. - 2011. - №6 (204). - С. 92-100.
- [Прилуцкий, 2007] Прилуцкий, М.Х. Многокритериальные многоиндексные задачи объемно-календарного планирования/ М.Х. Прилуцкий // Известия АН. Теория и системы управления. - 2007. - №1. - С.78-82.
- [Прилуцкий и др., 1997] Прилуцкий, М.Х. Метод декомпозиции для решения комбинаторных задач упорядочения и распределения ресурсов / М.Х. Прилуцкий, Д.И. Батищев, Э.Д. Гудман, И.П. Норенков // Информационные технологии. 1997. №1. С.29-33.
- [Султан-Заде и др., 1996] Султан-заде, Н.М. Повышение производительности ГПС путем оптимизации расписаний / Н.М. Султан-заде, Р.Р. Загидуллин. М.: СТИН, 1996. №12. С.9-13.
- [Танаев и др., 1975] Танаев, В.С. Введение в теорию расписаний / В.С. Танаев, В.В. Шкурба. М.: - Наука, 1975. 256 с.
- [Блехерман, 1979] Блехерман, М.Х. Оптимизация загрузки оборудования автоматизированных участков из станков с ЧПУ/Блехерман М.Х. // Станки и инструмент. -1979. -№ 5.- с. 3-5.
- [Балахонов и др., 2006] Балахонова, И.В., Волчков, С.А., Капитуров, В.А. Интеграция процессов с помощью ERP-системы. - СПб.: Приоритет, 2006.-464 с.

MODEL OF INTELLIGENT PROCESS CONTROL SYSTEMS MAINTENANCE

Burdo G.B. *, Fedotova A.V.**

*Tver State Technical University (TvSTU),
Tver, Russia
gbtms@yandex.ru

**Moscow Bauman state technical University
Moscow, Russia

Given the hierarchical structure of intelligent control system, reflecting the structure of the processes of planning and management of production units. Shows the main subsystems of APCS. Given the model and the algorithms used in the subsystem volume, calendar and operational planning and scheduling. Outlines the main features of the proposed control system governing low inertia. The methods of forming the priority schemes of service objects that provide the variability of the calculations and choice of optimal in accordance with the established criterion.

Introduction

It is known that the use of automated control systems (process control systems) of technological processes in multiproduct engineering hampered by the necessity of adjusting the plans of all levels due to rapid industrial situation and the constant emergence of new contracts.

The situation is exacerbated for technological processes (TP) with a great deal of manual labor, which leads to uncertainty of run-time non-recurring operations. Below we discuss process control systems technical maintenance of aircraft equipment, allowing to minimize the impact of the above negative factors. In [Burdo, 2010], [Burdo, 2011, 1] were presented the General principles of creation of automated control systems.

The structure of automated control system of processes of aircraft maintenance, the corresponding shown in Fig. 1. The hierarchy of subsystems of automated control systems in power system the principle of continuity corresponds to the structure of the production plans of the aircraft repair enterprises of feedback between subsystems provide the ability to adjust plans at all levels, which ensures their feasibility.

Main Part

The main problems encountered when using the automated control systems of technological processes (ACSTP) in multiproduct production, which often are part of the production system of enterprises, producing high technology products, due to the high dynamism of the situation in the companies.

This is due to the constant emergence of new contracts during the calendar year, resulting in the adjustment of production plans of enterprises; the approximation of technical regulation because of deficiency of time for technological training, leading to inaccuracies operational planning. In the authors' opinion, the proposed automated control system of technological processes (for example, technical maintenance of aircraft) is devoid of these shortcomings.

This is due to the following factors. The hierarchical structure of ACSTP reflects the hierarchy of the production plans of the enterprise, and presents subsystems, volume, calendar, operational planning and scheduling. Process control systems, as a system in whole, the information associated with the management system of the organization and is its subsystem. To ensure the scheduling, communication and implemented with the operations.

All the subsystems of the process control information and connected, it is possible to correct all types of plans. Planning is accomplished through the use of iterative procedures, obespechivaushyi "thin" the fulfillment of production plans, it is possible due to the feedbacks adjusting production plans of higher levels, which ensures their feasibility.

When the production planning provides for local capacity reserves and funds operating time, the measures for the use of their reserves. The subsystem of operational planning, ending the scheduling of production units based on the relative priority method.

Shows refined problem statement of the schedule of the production unit, particularly associated with the performance of work groups of workers and the use of priority schemes of service. Identified and the methods of forming the priority schemes of service objects that provide a variety of calculations and possibility of a choice of optimum in accordance with the established criterion.

In refining, process control systems can be implemented as a support system solutions for various enterprises diversified engineering, servicing, repair and Assembly of high technology products.

Conclusion

In the The proposed control system was developed as an "advising" to help make decisions quickly at all levels of planning, and has elements of artificial intelligence. Its first distinguishing feature of known class systems Scada and ERP [Balakhonov et al., 2006] is the implementation of operational links between plans at all levels and a comprehensive recalculation of new orders (contracts during the year) with different priorities.

The second feature is the high degree of formality of procedures, and as a result, the accuracy of the recommended solutions.

Third – the possibility of learning through the analysis of the calculation rules and actual lead times operations and the design and actual factors of the parallel execution time of operations. In refining, process control systems can be implemented as support systems solutions on various diversified engineering companies engaged in servicing, repairing and Assembly of high technology products.

The work is executed at financial support of RFBR, project № 14-01-00324.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 519.859

САМООРГАНИЗУЮЩИЕСЯ СЕТЕВЫЕ ЭКСПЕРТНЫЕ СРЕДЫ В ЗАДАЧАХ РАЗВИТИЯ НАУКОЕМКИХ ПРОИЗВОДСТВ

Виноградов Г.П., Виноградова Н.Г.

Тверской государственной технической университет, г.Тверь, Россия

wgp272ng@mail.ru

Рассматривается проблема построения самоорганизующихся сетевых экспертных сред для обеспечения инновационного развития производств, выпускающих высокотехнологичную и наукоемкую продукцию и услуги.

Ключевые слова: экспертная среда; самоорганизация; сетевые технологии; согласование представлений; инновационное развитие.

Введение

В XXI веке развитие организационных систем во все большей степени переходит от эволюционного к проектному. Управление этим процессом предполагает сочетание казуального подхода с теологическим (целевая детерминация). Движущей силой развития становится деятельность, представленная множеством институциональных построений, цель которых – совершенствование инновационной активности [Gubanov, 2014] путем развития видов информационных взаимодействий в функционировании социально-экономических систем.

Из них наиболее перспективной формой поиска, производства нового знания и определения направления его использования становятся саморазвивающиеся сетевые экспертные среды [Райков, 2009]. Интеграция в сетевой структуре естественных интеллектов образует коллективный стратегический субъект, который является средством синергии знаний и действий в процессе развития.

В связи с этим необходима разработка методологических основ организации подобных сред и управления ими. Центральным аспектом в этом становится оценка субъектами (в дальнейшем агенты) ценности, полезности нового знания и технологий, оценка рисков, определение механизмов их исключения, соотнесения нового знания с целями и задачами развития. Это требует, в свою очередь, новых моделей формирования представлений агентов о предметной области при определении направления развития.

1. Принципы создания и применения систем управления эволюцией организационно-технологической системы

Состояние организационно-технологической системы (ОТС) должно рассматриваться как целеустремленное состояние, а результат принятия решения – это результат компромисса между целеустремленными выборами агентов и центром в ситуации целеустремленного состояния с учетом воздействия на систему управляющих воздействий и возмущений внешней среды [Виноградов, 2013]. Информация о состоянии активных элементов системы, получаемая в процессе управления, используется центром для настройки механизма функционирования ОТС, который включает в себя процедуры формирования данных о резервах, планирования, стимулирования, обучения, коучинга, регулирования и т.п. Особенность производства, использующего наукоемкие технологии, связана со специализацией знаний и повышением степени интеллектуализации труда персонала, распределением точек принятия решений и ответственности, что делает зависимым его функционирование от результатов выбора агентов. Как результат это приводит к различной степени использования ими своего интеллектуального, производственного, финансового и другого потенциала. Поэтому одной из важнейших проблем для центра является повышение эффективности ОТС путем повышения эффективности функционирования активных агентов за счет использования эндогенных факторов путем развития их способности видения и учета перспективы, а также последствий

принимаемых решений, влияния изменений во внешней среде и случайных факторов. Решение ее возможно на основе соответствующих информационных технологий и средств автоматизации обработки информации. Их внедрение должно позволять согласовывать в целевом состоянии интересы агентов и центра на каждом периоде функционирования системы и их оценки ценности ситуации целеустремленного состояния должны быть согласованы для поддержания соответствующего уровня мотивированности и единого понимания задач персоналом [Виноградов, 2011].

Основные принципы и методы создания и применения систем управления эволюцией ОТС отражают специфику построения и применения процессов согласования интересов и оценок ситуации целеустремленного состояния путем учета субъективных представлений агентов и активизации их творческого потенциала, построения эффективных схем согласования интересов и определения на этой основе направления эволюции ОТС.

1. *Принцип активного окружения.* Основными элементами внешней среды для организационно-технологической системы являются поставщики сырья, энергоресурсов, полуфабрикатов, потребители выпускаемой продукции, представители экологических служб, отвечающих за безопасность окружающей среды. Для понимания ситуации целеустремленного состояния системы надо рассматривать как минимум цепочку с глубиной рефлексии – два. Эта цепочка функционирует в определенной организационно-технологической или организационно-экономической среде. Такой схеме соответствует трехуровневая активная многоагентная система.

Поставщиков, конкурирующих между собой, может быть множество. Они образуют группу агентов низшего уровня, и их цель состоит в поставках по возможно большей цене. Агенты-производители образуют второй уровень активной системы. Они основной своей целью считают приобретение сырья, полуфабрикатов и услуг возможно более высокого качества. К центру третьего уровня относятся представители системы, в которую входят потребители исходных продуктов, им делегированы права на приобретение этих продуктов. Поэтому их основной целью является получение реального экономического эффекта, который они получают от приобретения продуктов.

Одной из возможностей для этого является анализ производства у поставщиков и потребителей продукции и по возможности у потребителя, являющегося потребителем того, кому поставляется продукция. Естественно поставщик и потребитель должны обладать подобным знанием. Такое рода знание создает предпосылки определения желаемых состояний в пространстве экономических показателей за счет

появления идей улучшения организации производства, выбора шагов решений в этом пространстве с одновременной трансляцией усовершенствований потребителю. Это позволяет не только учитывать пожелания клиента, но сформировать его потребности, что создает направление дальнейших нововведений. Информационная система должна обеспечить этот процесс соответствующими данными и знаниями.

2. *Принцип коалиции.* Этот принцип заключается в образовании при разработке и внедрении системы управления эволюцией ОТС ответственной группы, включающей в себя поставщиков информационных технологий, разработчиков системы, пользователей и руководителей, осуществляющих финансирование работ, для согласования и подтверждения факта выявления резервов за счет предлагаемых методов совершенствования технологического процесса и методов управления им. Такая коалиция осуществляет проработку и принятие решений по всем стадиям создания системы управления эволюцией ОТС, по объемам финансирования, по срокам и этапам работ, оценке достигаемых результатов и т.д.

3. *Принцип моделирования.* Вследствие распределения знания о свойствах технологического процесса и ограниченных когнитивных возможностей человека особенностью построения и использования системы управления эволюцией ОТС является активизация творческого потенциала производственного персонала, что обуславливает использование процедур согласования и согласованной оптимизации. Их цель формирование представлений на всех уровнях о возможных состояниях системы. Получение подобной информации вызывает необходимость создания на стадиях создания и эксплуатации системы управления эволюцией ОТС совокупности моделей (концептуальных, функциональных, информационных, структурных, алгоритмических, эвристических, математических, имитационных и др.) человеко-машинных процессов принятия согласованных решений. Они должны описывать различные функциональности системы, использовать как количественную, так и качественную информацию и образовывать интегрированный комплекс, соответствующий точкам принятия решений. Их интеграция должна быть выполнена на основе теории многоагентных систем, теории интеллектуальных организаций, теории активных систем, методов и средств моделирования, включая алгоритмический подход, структурное проектирование, на основе синергетического принципа дополтельности [Вагин и др., 2008].

4. *Принцип гибкости.* Система управления эволюцией ОТС должна реализовывать технологию выявления и построения моделей поведения агентов, согласования решений с помощью разработанных и применяемых

информационных и программных продуктов с развитым интерфейсом для формирования состояния убежденности у агентов за счет информационного управления с учетом ограничений на их информационную культуру. Она должна обеспечить применение методов и моделей построения представлений пользователей о предметной области на всех уровнях, их согласование и согласованной оптимизации на множестве компромиссных решений, принятых на основе субъективных представлений, при использовании современных обеспечивающих информационных технологий (хранилищ данных, экспертных систем, средств выявления знаний, предметно ориентированных программных систем и т.д.). Она должна обеспечить решение этих задач при изменении структуры предметной области, модернизации методов и моделей, программных и технических средств.

5. *Принцип применения.* Средства выявления и моделирования формирования субъективных представлений агентов и принятия на их основе решений, процессы управления представлениями, согласования и согласованной оптимизации должны быть составной частью информационных технологий на всех уровнях ОТС и обеспечить методическую, математическую и информационную поддержку в деятельности агентов. Они должны быть встроены в естественный процесс управления технологическим процессом, функциональностями в организационно-экономической составляющей ОТС на основе средств автоматизации для придания им желаемых свойств.

6. *Принцип поэтапного внедрения.* Эффект от разработки и внедрения АСУ различного назначения по зарубежному опыту лишь в 30% случаев соответствует ожиданиям. Основная причина в этом состоит в различии представлений о целях, задачах, применяемых средств и т.п. у пользователей и разработчиков. Дело в том, что их разработка и внедрение осуществляется в изменяющейся активной экономической среде, точное описание которой получить в принципе невозможно или нецелесообразно. Поэтому для сокращения сроков и получения реального эффекта от внедрения системы управления эволюцией ОТС необходимо использовать принцип распределенного внедрения. Он должен реализовываться на всех стадиях создания системы, начиная от предпроектного обследования до внедрения. Для оценки эффективности решений необходимо проведение комплексных экспериментальных работ при использовании предлагаемых средств, реализованных решений в виде игрового имитационного моделирования в упрощенной форме, но учитывающей основные свойства объектов управления.

7. *Принцип визуализации.* Его использование при построении пользовательского интерфейса для обеспечения процессов анализа и согласования информации направлено на формирования образа как вырабатываемого, так и принятого решения.

Это дает возможность пользователям выполнять оценку, как собственной деятельности, так и качества решений при создании системы управления ОТС [Гаврилова, 2008].

2. Механизм самоорганизации сетевых экспертных систем

Саморазвитие предполагает вовлечение через использование сетевых технологий продвинутых потребителей товаров и услуг в инновационный процесс совершенствования потребительских свойств товаров и услуг, получения идей или контента путем обращения к их креативным способностям в обмен на вознаграждение, соответствующее вкладу. В Советском Союзе такое решение проблем называлось движением новаторов. На Западе оно было несколько расширено, модифицировано и названо краудсорсингом. Для его реализации разработаны соответствующие технологии. Однако их прямая реализация приводит появлению большого количества информационного шума и мусора, содержащего, как правило, бездоказательные критические высказывания. Поскольку при оценке сложных систем, продуктов и услуг требуется не только гражданская позиция, но и квалификация, то необходима система, обеспечивающая направление процесса поиска решений и идей, а также обеспечивающая фильтрацию, обобщение информации, определение ее ценности и перспективности. Способом повышения краудсорсинговых решений является синтез технологий краудсорсинга и сетевой экспертизы [Райков, 2009].

По сравнению с традиционными технологиями поиска решения эти два подхода обладают следующими преимуществами:

Самоорганизация. Создание идей, предложений по совершенствованию продукции и услуг практически неограниченным числом экспертов.

Независимость. Возможность получения информации, отражающей различные точки зрения с объяснением позиции.

Малобюджетность. Люди участвуют в подготовке предложений и выполняют оценки, как правило, безвозмездно или за небольшое вознаграждение, варьируя уровень вознаграждения в зависимости от результата.

Поскольку модели принятия решений в сетевой экспертизе должны строиться достаточно быстро, то задачи по извлечению информации, очистки и предварительной обработки (выделению признаков, семантическая кластеризация, построение онтологий и т.п.) необходимо выполнять средствами краудсорсинга.

Центр формирует запросы, оформляют их дизайн, дорабатывают семантическую кластеризацию и онтологию предметной области.

Цикл взаимодействия центра с экспертами будет продолжаться до тех пор, пока вклад новой информации в рост полезности сформированных представлений будет несущественным. Полученная модель представлений предметной области используется для формирования модели способа действия. Анализ удовлетворенности от принятого решения позволяет корректировать как модель представлений, так и модель способа действия (решения).

3. Математическая постановка и описание задачи принятия согласованных решений

Пусть задана иерархическая структура производства и потребления наукоемкой продукции перечислением входящих в нее агентов, описанием их прав и интересов, в соответствии с которыми они принимают решения. Кроме того, пусть определена регламентация процедур доступа, обработки и передачи информации. Будем считать, что возможности и права агентов в системе не одинаковы (что позволяет говорить об их неоднородности [Новиков, 2008]). Неоднородность проявляется в различной степени влияния агентов друг на друга.

Для целей анализа будем рассматривать базовую модель иерархической системы, состоящей из центра (агент верхнего уровня) и агентов нижнего уровня. Согласно [Vinogradov, 2011] разделим цели верхнего уровня условно на следующие группы:

1. Определение и согласование целей выживания и развития системы в целом в условиях неопределенности и динамики рынка;
2. Формирование согласованного восприятия внешней и внутренней среды системы;
3. Формирование согласованного представления о ситуации целеустремленного состояния;
4. Разработка согласованного плана действий, учитывающего интересы всех подсистем;
5. Оптимизация по возможности обобщенных показателей эффективности функционирования системы, представляющих собой функции от параметров всех подсистем (агентов).

Решение последних двух задач может быть записано следующим образом:

$$(u, v) \in \Omega, \quad (1)$$

где u – управление центра, $v = (v_1, v_2, \dots, v_n)$ – вектор управлений подсистем, Ω – множество таких управлений, которые переводят систему в одно из состояний множества X (X – область желаемых состояний системы).

Условие (1) может быть выполнено лишь на основе некоторой совокупности предположение о

поведении и информированности всех агентов системы. Примем следующие предположения:

1. центр может сформулировать желаемое состояние системы в виде некоторого набора обобщенных показателей, которые являются функциями параметров агентов. Конкретные зависимости центру в общем случае неизвестны, но для их определения он может использовать данные прошлых периодов функционирования системы, информацию о результатах, полученных конкурентами в данном сегменте рынка и результаты сетевой экспертизы. Такое предположительное знание будем называть представлениями центра о возможностях агентов и направлении развития системы;

2. на основе представлений о возможностях агентов и желаемых состояниях центр определяет свое управление (план действий и информационный запрос) $u \in U$ и сообщает его агентам;

3. при известном управлении u_i -й агент выбирает свое управление $v_i \in V_i(u_i)$, которое переводит его в состояние $y_i \in Y_i$, где Y_i – множество возможных состояний. Будем считать, что агенты взаимозависимы по выбору своих состояний. Это предполагает при выборе учет глобальных ограничений Y^{2l} . Следовательно, при выполнении условия (1) система может находиться в одном из следующих состояний

$$Y = Y^{2l} \cap \left(\prod_{i=1}^n Y_i \right). \text{ Здесь } \prod_{i=1}^n Y_i \text{ – это множество,}$$

определяемое локальными ограничениями, например, по объему выпуска, качеству, затратам и т.д.

4. пусть поведение агента соответствует гипотезе рационального поведения. Тогда выбор агента будет сделан так, чтобы максимизировать свой субъективно понимаемый критерий эффективности $G_i(u, v_i)$;

5. центр, в общем случае, приблизительно знает пространство управлений и критерии эффективности агентов, поэтому он должен для создания представлений о желаемых состояниях агентов организовать с ними обмен информацией. Для этого он запрашивает у агентов оценки качества продукции и услуги желаемые состояния y_i , уточняет их интересы в окрестности полученного решения в обмен на стимулирование за получения встречной информации;

6. получив новую информацию, центр пересчитывает решение и задает новые вопросы агентам до тех пор, пока не будет получено точное или близкое к нему решение.

7. при наличии неопределенности факторов центр стремится в соответствие с принципом детерминизма снизить ее уровень за счет применения процедур, повышающих степень полезности своих представлений.

Описанная модель формирования плана развития производства предполагает, что агенты, как управляющие своими подсистемами, так и использующие ее на уровне своих представлений хорошо знают и «чувствуют» свои объекты управления и могут с достаточной степенью точности ответить на запросы центра.

Эти предположения позволяют определить множество допустимых управлений центра следующим образом

$$U^0 = \left\{ u \in U \mid V(u) = \prod_{i=1}^n V_i(u) \neq \emptyset, \right. \\ \left. (u, v) \in \Omega, \forall v \in R(u) = \prod_{i=1}^n R_i(u) \right\},$$

$$\text{где } R_i(u) = \text{Arg} \max_{v_i \in V_i(u)} G_i(u, v_i).$$

Если у центра имеется критерий эффективности $F(u, v)$, то оптимальным гарантирующим управлением является такое $u^0 \in U^0$, что

$$F^0 = \sup_{u \in U^0} \inf_{v \in R(u)} F(u, v)$$

Общность интересов центра и агентов определяется условиями на суммарное вознаграждение

$$\sum_{i=1}^n G(u_i, v_i) + F^0 \leq C(\bar{Y}), \quad (2)$$

где \bar{Y} – доход от выполнения заказов, $C(\cdot)$ – фонд материального поощрения или фонд оплаты труда.

Таким образом, при согласовании модели развития производства товаров и услуг в системе с не противоположными интересами необходимо рассматривать интеграционный тип возможного компромисса, поскольку величина распределяемого ресурса согласно (2) является величиной переменной, зависящей от их суммарных усилий. Поэтому цель согласования состоит в достижении суммарного выигрыша, и он может быть получен путем сообщения достоверной информации и совместным решением проблем. В системах, где агенты связаны общими интересами и целью, справедлив принцип запрета эгоизма, так как какой-либо агент может нанести ущерб системе, только если при этом он наносит ущерб себе.

Пусть объект управления i -го агента описывается уровнем затрат y_i^{6x} , уровнем выпуска

y_i^{6bx} и набором режимных параметров z_i , с помощью которых агент может влиять на ход процесса в своей предметной области. Тогда множество возможных состояний i -го агента можно определить следующим образом

$$Y_i = \left\{ \begin{aligned} &y_i \mid \bar{y}_i^{6x} \leq y_i^{6x} \leq \bar{y}_i^{6x}, y_i^{6bx} = \\ &= w_i^*(y_i^{6x}, z_i) \in Y_i^{6bx}, z_i \in Z_i \end{aligned} \right\}$$

Сформулированная задача является задачей поиска максимина со связанными ограничениями.

Обозначим через $y_i^{6bx} = w_i^*(y_i^{6x}, z_i) \in Y_i^{6bx}$ представление агента о функционировании объекта управления, тогда выражение

$$Y_i^* = \left\{ \begin{aligned} &y_i \mid \bar{y}_i^{6x} \leq y_i^{6x} \leq \bar{y}_i^{6x}, y_i^{6bx} = \\ &= w_i^*(y_i^{6x}, z_i) \in Y_i^{6bx}, z_i \in Z_i \end{aligned} \right\}$$

можно рассматривать как субъективные представления агента о множестве возможных состояний. Очевидно, оно определяется его знанием, опытом.

В свою очередь центр может иметь собственное представление о возможностях агентов в виде

$$Y_i^* = \left\{ \begin{aligned} &y_i \mid \bar{y}_i^{6x} \leq y_i^{6x} \leq \bar{y}_i^{6x}, y_i^{6bx} = \\ &= w_i^*(y_i^{6x}, z_i) \in Y_i^{6bx}, z_i \in Z_i \end{aligned} \right\}$$

В случае эффективных производств можно с достаточной степенью точности предположить выполнение гипотезы о полной информированности центра о возможностях, желаемых состояниях агентов и их продуктивности с точностью до параметров. Это означает, что центр располагает всей необходимой информацией о моделях $V_i, i = \overline{1, n}$ агентов, но не знает выбор агентов. В этом случае, центр, используя это знание, рассчитывает управление $u^0 \in U^0$, а агент делает только один ход, выбирая свое состояние y_i и управление v_i из множества допустимых состояний $B_i(u_i) = Y_i(u_i) \cap Y_i^*$, пытаясь при этом максимизировать значение своей целевой функции

$$G_i(u_i, v_i) \rightarrow \max_{v_i \in B_i(u_i)} \quad (3).$$

Тогда результат выбора всех агентов определяется в этом случае как множество $R(D)$

$$\text{всех ситуаций, где } D = Y^{2l} \cap \prod_{i=1}^n B_i(u_i)$$

множество возможных состояний всей системы.

Варьируя управление $u = \{u_i, i = \overline{1, n}\}$, центр может проигрывать возможные сценарии поведения системы.

4. Алгоритм построения агентом множества согласованных состояний

Агент, обладая более детализированной информацией, может при создании центром более привлекательных условий u_k путем решения задачи (3) определять для себя такой вектор $x_k^{(h)}$, который позволяет расширить множество $X_k^{(h)}$ представлений центра о его возможностях, h – шаг итерационного процесса.

Обозначим через $\omega_k = \{\omega_k^{(h)}, h = \overline{1, H}\} \in A_k$ – вектор параметров состояния, определяющий значения вектора действий $x_k^{(h)} = \{x_{kj}^{(h)}, j \in \overline{1, m_k}\} \in X_k^{(h)}$. Можно считать, что этот вектор описывает знание агента возможностей контролируемого им объекта управления. Здесь A_k – множество возможных значений вектора состояния. Будем считать, что агент обладает способностями, знаниями, которые гарантируют существование $\Psi_k: A_k \rightarrow X_k^{(h)}$.

Доступный агенту уровень знаний о конструктивных особенностях потребляемой продукции и услуг делают справедливым предположение о существовании для k -го агента предельного множества параметров состояния. Обозначим через

$O_k^* = \{o_k^*/o_k(x_k), x_k \in X_k^{(h)}(\omega_k^{(h)}), \omega_k^{(h)} \in A_k\}$ – множество достижимости или множество предельных возможностей.

Будем предполагать, что агент за счет своих креативных способностей, способности к самообучению и поиску новой информации при соответствующем стимулирующем воздействии центра способен определять такие состояния

$\omega_k^{(1)} \in A_k$ и $\omega_k^{(2)} \in A_k$, что возможно $\omega_k^{(2)} \succ \omega_k^{(1)}$, где символ \succ означает «более значимо» и при этом $X_k^{(1)}(\omega_k^{(1)}) \subseteq X_k^{(2)}(\omega_k^{(2)})$. Следовательно, существует такая последовательность $\omega_k^{(1)}, \omega_k^{(2)}, \omega_k^{(3)}, \dots$, что

$$\lim_{h \rightarrow \infty} o_k^{(h)}(x_k^{(h)}(\omega_k^{(h)})) = O_k^*.$$

То есть агент путем изучения объекта управления способен определить его предельные возможности для достижения желаемого состояния. Последовательность

$\omega_k^{(1)}, \omega_k^{(2)}, \omega_k^{(3)}, \dots$ будем называть последовательностью вскрытия резервов от потенциальных конструктивных решений за счет их доработки и упрощения в обслуживании.

Такая способность агента формировать расширяющееся множество способов действия позволяет определить следующие свойства целевой функции агента и областей достижимости:

$$\forall \omega_k^{(1)}, \omega_k^{(2)} \in A_k, \omega_k^{(2)} \succ \omega_k^{(1)}, X_k^{(1)}(\omega_k^{(1)}) \subseteq X_k^{(2)}(\omega_k^{(2)}) \mapsto E\varphi_k(x_k^{(2)}) > E\varphi_k(x_k^{(1)})$$

Это условие означает, что поведение агента при выполнении принципа рациональности соответствует закону повышающихся потребностей, который в литературе по психологии поведения определяет мотивированность и целеустремленность агента.

В реальных условиях агент при превышении некоторого порога значимости $\Delta = E\varphi_k(x_k^{(2)}) - E\varphi_k(x_k^{(1)})$ изменения ценности ситуации целеустремленного состояния по результату способен идентифицировать предпочтительные способы действия и видит открывающиеся возможности при изменении структуры своей информированности (знания).

Выработка решения при таком подходе заключается в реализации совокупности последовательных процедур, предназначенных для поиска промежуточных решений, на основании которых агент уточняет свои возможности и формирует окончательное решение. Полный цикл его формирования k -м агентом состоит в выполнении следующих шагов на этапе h :

1. Формирование множеств A_k и $X_k^{(h)}$ на основе знаний, опыта, интуиции и располагаемой информацией о параметрах состояния $\omega_k^{(h)}$.

Просмотр множества A_k и формирование точки

$$O_k^{(h)*} = \left\{ \begin{array}{l} o_k^{(h)*}/o_k^{(h)*}(x_k^{(h)}), x_k^{(h)} \in \\ \in X_k^{(h)}(\omega_k^{(h)}), \omega_k^{(h)} \in A_k \end{array} \right\}. \text{ Проверка,}$$

существует ли $x_k^{(h)*}$ такое, что

$o_k^{(h)}(x_k^{(h)*}) = o_k^{(h)*}$. Если – да, то $x_k^{(h)*}$ – это компромиссное решение, а $o_k^{(h)*}$ – прогнозируемая ситуация, в противном случае переход к п.2.

2. Решение задачи поиска потенциально-предпочтительного набора действий $x_k^{(h)*} \in X_k^{(h)}(\omega_k^{(h)})$, позволяющего сформировать вектор $\bar{o}_k^{(h)}$ предельных значений критериев при использовании имеющегося на данный момент знания о правиле $\Psi_k^{(h)}$ и структуре множества A_k . Так как компоненты $\bar{o}_k^{(h)}$, $i = \overline{1, N}$ порознь достижимы, а вместе – нет, то делается попытка найти компромиссное решение. Если агент не согласен попытаться найти компромиссное решение за счет компенсаторных уступок по каждому критерию, которые несколько хуже решения $\bar{o}_k^{(h)}$, то переход к п. 3, иначе к п. 5.

3. Исследование направлений возможного расширения множества A , организация процедур поиска новой информации (знания) о $\omega_k^{(h)} \in A_k$ и правиле $\Psi_k: A_k \rightarrow X_k^{(h)}$.

4. Если расширение множества A_k возможно, то переход к п. 1, иначе фиксация ситуации, что компромиссное решение не может быть найдено при выбранном векторе $\bar{o}_k^{(h)*}$.

5. Получение сведений от агента достаточных для определения вектора $\bar{o}_k^{(h)} < \bar{o}_k^{(h)*}$, где $\bar{o}_k^{(h)}$ минимальные требования агента к принимаемым им во внимание результатам.

6. Выполнение процедуры поиска минимально-предпочтительной точки в пространстве критериев по направлению предпочтения $\bar{o}_k^{(h)}, \bar{o}_k^{(h)*}$, определение вектора $\omega_k^{(h)*} \in A_k$ и $x_k^{(h)*} \in X_k^{(h)}(\omega_k^{(h)*})$ – минимального значения плановых показателей, соответствующих значениям компонент вектора $\bar{o}_k^{(h)}$.

7. Если полученные значения для $x_k^{(h)*}, \bar{o}_k^{(h)*}$ принимаются как компромиссное решение, то процедура останавливается, в противном случае переход к п. 8.

8. Для ограничений на $\bar{o}_k^{(h)}$ определяется приоритетная координата $i \in [\overline{1, N_k}]$, по которой делается расширение множеств A_k и $X_k^{(h)}$, так чтобы $o_k^{(h)}(x_k^{(h)}) = o_k^{(h)} + \Delta_k^{(h)}$, где $\Delta_k^{(h)}$ минимально возможное улучшение, которое является значимым для агента и определяется по его высказываниям о “гибкости” ограничения на основе выполнения процедур поиска дополнительной информации. Переход к п. 1.

Описанный алгоритм использует три типа механизмов, применение которых порождает интерактивный процесс для построения компромиссного решения.

Механизм анализа – это механизм, с помощью которого агент в момент r обрабатывает сведения, полученные на шаге $r-1$ для построения множеств P_k и X_k с целью определения $x_k^{O*}, y_k^{O*}, c_k^{O*}, z_k^{O*}$. Здесь сопоставляются результаты, полученные на момент r с результатами на шаге $r-1$. Формируется представление о значениях вектора o^{O*} и значений уступок по его компонентам. Строится предварительное представление о желательных значениях показателей $E\varphi_k(o(y^{O*}))$ и $EE_k(o(y^{O*}))$.

Механизм целеполагания – это механизм, который по результатам анализа определяет условия возможности достижения желательных значений $x_k^{O*}, y_k^{O*}, c_k^{O*}, z_k^{O*}, E\varphi_k(o(y^{O*}))$, $EE_k(o(y^{O*}))$. Рассчитывается идеальная точка в пространстве оценок o , связанная с множествами P_k и X_k – это точка \bar{o}^* , имеющая координаты $\bar{o}_i^* = \max_{p \in P_k, x \in X_k} o_i(x(p)), i = \overline{1, L}$. То есть \bar{o}^* –

наибольшее значение оценки по координате i , которое может быть получено с помощью намеченных действий. Эта точка образует центр области поиска предпочтительных состояний. Если при найденном значении координат точки \bar{o}^* ограничения не выполняются, то агент рассматривает возможность расширения множеств P_k, X_k и C_k за счет включения дополнительных переменных, осуществления структурных изменений и т.п. для ослабления наиболее жестких ограничений так, чтобы выполнялось включение $P_k(r) \subseteq P_k(r+1)$ и $X_k(r) \subseteq X_k(r+1)$. Если это невозможно, то формируется сведения о

координатах $\underline{o}_i < \bar{o}_i^*$ точки $\underline{o}_i \neq \bar{o}_i^*$, которая была бы достижимой и лучшим компромиссом в окрестности точки \bar{o}^* . Путем минимизации потерь $\sum_{i=1}^L |o_i - o_i^{o*}|$ определяются величины $p_i = o_i - o_i^{o*} \neq 0$ примерно эквивалентные, небольшие и значащие для агента.

Механизм самоорганизации – это механизм, который приводит к получению знаний о правилах Δ и Ψ для расширения множеств P_k , X_k и C_k . Он может иметь различные формы: 1) проведение экспериментальных исследований на объекте с целью улучшения своего представления о его функционировании; 2) анализ обстановки V_{-k} ; 3) привлечение экспертов для подготовки рекомендаций по улучшению режимов ведения процесса; 4) процедуры опроса персонала; 5) процедуры голосования и т.п.

Заключение

В статье приведены соображения, являющиеся основанием для перехода с простого маркетингового анализа рынка и пассивного изучения потребностей потребителей товаров и услуг на использование активных методов. Основой их построения следует считать создание самоорганизующихся экспертных сред на базе современных сетевых и телекоммуникационных технологий. Такой подход означает также активное использование различных средств информационного управления для создания соответствующих информационных структур у экспертов.

Библиографический список

- [Gubanov, 2014] Gubanov, D. E-Expertise: Modern Collective Intelligence, Springer. Series: Studies in Computational Intelligence, //Gubanov, D., Korgin, N., Novikov, D., Raikov, A. Vol. 558, 2014, XVIII, 112 p.
- [Райков, 2009] Конвергентное управление и поддержка решений. / Райков А.Н. - М.: Издательство ИКАР, 2009. – 245 с.
- [Виноградов, 2011] Виноградов, Г.П. Моделирование поведения агента с учетом субъективных представлений о ситуации выбора / Виноградов, Г.П., Кузнецов В.Н. // Искусственный интеллект и принятие решений. 2011. № 3. с. 58-72.
- [Vinogradov, 2011] Vinogradov, G. Decision Making based of Subjective Conceptions of Decision situation / Vinogradov G. // Interactive Systems and Technologies: the Problems of Human-Computer Interaction, - Collection of Scientific papers. Uljanovsk, 2011, pp. 403-414.
- [Новиков, 2008] Новиков, Д. А. Математические модели формирования и функционирования команд / Новиков Д.А.; - М.: Физматлит, 2008.
- [Виноградов, 2013] Виноградов Г.П. Методы и алгоритмы принятия решений в автоматизированных системах управления производствами с непрерывной технологией на основе субъективных представлений: монография. / Г.П. Виноградов. Тверь: ТГТУ, 2013. 256 с.

[Вагин и др., 2008] Достоверный и правдоподобный вывод в интеллектуальных системах / Вагин В.Н. [и др.]; - М. : ФИЗМАТЛИТ, 2008.

[Гаврилова, 2008] Гаврилова, Т. А. Визуальные методы работы со знаниями: попытка обзора / Т. А. Гаврилова, Н. А. Гулякина // Искусственный интеллект и принятие решений, 2008, № 1, С. 15-21.

SELF-ORGANIZING NETWORK EXPERT COMMUNITY IN THE TASK OF DEVELOPING SCIENCE-INTENSIVE INDUSTRIES

VinogradovGP, Vinogradov NG

Tver State Technical University, Tver, Russia

wgp272ng@mail.ru

The problem of building a network of self-organizing expert environment for the development of innovative industries that produce technology-intensive products and services.

Introduction

In the XXI century the development of organizational systems are increasingly shifting from an evolutionary to the project. Managing this process involves a combination of casual approach to the theological (target determination). The driving force behind development is the activity provided by a variety of institutional constructions whose purpose - improving the innovation activity through the development of types of information interactions in the functioning of the socio-economic systems.

In this connection the development of methodological bases of the organization of such media and management. A central aspect of this is the assessment of the subjects (hereinafter agents) values, the utility of new knowledge and technologies, risk assessment, determination of mechanism of exclusion, correlating new knowledge with the objectives of development. This requires in turn new models of formation of representations about the domain agents in determining the direction of development.

Main Part

Report theses in volume of 4 paragraphs.

Conclusion

The paper presents the considerations are the basis for the transition from a simple market analysis and study of the needs of passive consumers of goods and services for the use of active methods. The basis of their construction is the creation of self-organizing expert environment on the basis of today's network and telecommunication technologies. This approach also means the active use of various means of information management for the establishment of appropriate information structures from the experts.



УДК 004.942

НЕЧЕТКИЕ МОДЕЛИ ПОВЕДЕНИЯ ПРЕДПРИНИМАТЕЛЯ ПРИ ПЕРЕГОВОРАХ ОБ ИНВЕСТИРОВАНИИ В УСЛОВИЯХ ТОЧНОГО ЗНАНИЯ ХАРАКТЕРА ПОТЕНЦИАЛЬНОГО ИНВЕСТОРА

Палюх Б.В., Ключин А.Ю., Мутовкина Н.Ю.

*Тверской государственной технической университет,
г. Тверь, Российская Федерация*

Pboris@tstu.tver.ru

klalex@inbox.ru

letter-boxNM@yandex.ru

В статье рассматриваются модели поведения предпринимателя при переговорах с инвестором, объектом которых является инвестиционный проект. При выборе инвестор руководствуется не только объективными критериями оценки проектов, но и своими субъективными предпочтениями. Предприниматель должен учитывать данное обстоятельство и построить беседу так, чтобы убедить потенциального инвестора принять участие в финансировании предлагаемого проекта. От того, насколько удачной будет стратегия поведения предпринимателя, зависит результат переговоров.

Ключевые слова: инвестиционный проект; нечеткие множества; расплывчатые цель и ограничения; стратегия поведения.

Введение

Как правило, инвестор рассматривает предлагаемые ему предпринимателем проекты, опираясь как на объективные факты, так и на собственные суждения, предпочтения и мнения, что создает определенные субъективные условия, в которых инвестор решает: принять проект к финансированию или нет. К объективным критериям относятся, например, показатели доходности инвестиционного проекта: чистый дисконтированный доход (Net Present Value – NPV), внутренняя норма доходности (Internal Rate of Return – IRR), дисконтированный срок окупаемости (Payback Period – PP). [Матвеев и др., 2005]. Субъективные критерии выражают отношение инвестора к предпринимателю и, следовательно, к проектам последнего. Другими словами, это отношения предпочтения: «доверяет – не доверяет», «есть взаимопонимание – нет взаимопонимания», «готов к сотрудничеству – не хочет сотрудничать», «вызывает симпатию – вызывает неприязнь» и т.п. Часто, при выборе стратегии своего поведения, как инвестор, так и предприниматель руководствуются интуитивными способностями. Но и интуиция тоже не всегда срабатывает. Иногда из-за личной неприязни или недоверия одному предпринимателю инвестор может выбрать проект другого

предпринимателя, несмотря на то, что тогда он получит меньший финансовый выигрыш. Предприниматель же, выбрав неверную модель поведения при переговорах с потенциальным инвестором, рискует остаться без источника финансирования своего проекта. И это при том, что в целом проект удовлетворяет всем финансовым критериям.

В дальнейшем будем называть инвестора и предпринимателя взаимодействующими агентами, каждый из которых имеет некоторую цель и, действуя в расплывчатых условиях, стремится ее достичь. Инвестор, предприниматель (активные элементы) и проект, предназначенный для финансирования (пассивный элемент), а также выстроенные на основе проекта агентские взаимосвязи представляют собой стохастическую систему, функционирующую в расплывчатых условиях. Для успешной адаптации и выживания каждый активный элемент системы должен обладать определенным запасом гибкости и устойчивости при взаимодействии с другими агентами.

1. Постановка задачи

В [Мутовкина и др., 2014] установлено, что уровень субъективности в принятии решения зависит от психо-поведенческого типа агента и для

каждого из агентов желательна ситуация, при которой разность между уровнем субъективности и объективными воззрениями в любой момент времени t будет стремиться к минимуму:

$$\begin{cases} \gamma_{inv}^c(t, r_{inv}, r_{ent}, \varepsilon) - \gamma_{inv}^o(t, CF, \varepsilon) \rightarrow \min, \\ \gamma_{ent}^c(t, r_{ent}, r_{inv}, \varepsilon) - \gamma_{ent}^o(t, CF, \varepsilon) \rightarrow \min, \end{cases} \quad (1)$$

где r_{inv} – поведенческий тип инвестора;

r_{ent} – поведенческий тип предпринимателя;

CF – множество показателей доходности проекта;

ε – случайная составляющая, под которой понимается возможное влияние факторов, не зависящих от поведенческих моделей агентов.

Поведенческий тип агента ($r, r \in R$), отражающий его предпочтения, является оценкой его состояний, в которых он пребывает при выполнении конкретной задачи; R – множество возможных предпочтений агента. Минимизация в (1) возможна при управлении поведенческим типом агентов и локализации случайной компоненты. Управление поведенческим типом осуществляется в процессе взаимодействия агентов: при переговорах агенты оказывают друг на друга определенное влияние (V).

Нечеткость моделей поведения обеих сторон обусловлена не только возможными различиями в их интересах и предпочтениях, но и исходными условиями взаимодействия (на старте инвестиционного процесса). Можно выделить четыре ситуации, каждая из которых предполагает наличие нечетких условий взаимодействия при переговорах об инвестировании:

- один проект – один инвестор;
- несколько проектов – один инвестор;
- несколько проектов – несколько инвесторов;
- один проект – несколько инвесторов.

Первый случай – самый простой. Цель предпринимателя заключается в убеждении инвестора принять участие в проекте, а инвестор может выбрать одну из двух стратегий поведения: участвовать в предлагаемом проекте или отказаться от участия в нем. Цель инвестора при этом состоит в выборе такой стратегии поведения, при которой он будет в выигрыше от участия в проекте, в наихудшем случае – его потери будут равны нулю.

Второй случай для инвестора сводится к решению задачи выбора оптимального проекта из множества альтернативных проектов. Инвестор выберет тот проект, который по совокупности критериев наберет большее количество баллов. Если таких проектов несколько, то инвестор при выборе будет руководствоваться личностными предпочтениями и выберет проект того предпринимателя, который произвел наилучшее впечатление при знакомстве, общении, презентации

проекта. Подготовка и презентация тщательно продуманного бизнес-плана, финансовой модели, установочного резюме, отличное знание и умение ориентироваться в собственном бизнесе, технологиях, маркетинге, финансах, уверенность в своей команде и стратегии – вот залог успеха предпринимателя при общении с инвестором. Инвестор наряду с фактами руководствуется своими субъективными предпочтениями, поэтому предприниматель должен завоевать репутацию человека, которому можно доверять и с которым приятно иметь дело.

Следует отметить, что один предприниматель может предложить потенциальному инвестору сразу несколько проектов на выбор. Среди предпринимателей возникает соперничество. Модель их поведения должна способствовать достижению цели, которой является привлечение инвестора в конкурентных условиях. Но рассмотрение конкурентных моделей – задача дальнейшего исследования, проводимого авторами.

Для третьего случая характерны аналитико-игровые модели поведения, предполагающие взаимный анализ состояний как предпринимателей, так и потенциальных инвесторов посредством переговоров по предлагаемым инвестиционным проектам. Каждый из предпринимателей выстраивает вектор потенциальных инвесторов от менее благонадежных к более представительным. У каждого инвестора формируется вектор инвестиционных проектов от менее приоритетных к более привлекательным. При этом также возможно соперничество и даже конфликтные ситуации. Цель предпринимателей состоит в нахождении каждым из них наилучшего по выделенным критериям отбора инвестора, а цель инвесторов – участие в проекте, максимизирующим выигрыш каждого из них.

Последняя ситуация встречается довольно редко и характерна для уникального инвестиционного проекта, имеющего все шансы на успех. Инициатором такого проекта, как правило, является не начинающий предприниматель, а компания, хорошо известная и зарекомендовавшая себя должным образом в рыночной среде. В этом случае конкурентные модели поведения характерны для множества инвесторов, целью каждого из которых является участие в заведомо прибыльном проекте.

В любой ситуации решение принимается инвестором в процессе переговоров с предпринимателем относительно проекта последнего. При этом предприниматель оказывает на инвестора определенные управляющие воздействия с целью убеждения последнего принять как можно более полное участие в финансировании данного проекта. Характер и частота управляющих воздействий зависят от личностных характеристик инвестора, т.е. его поведенческого типа (r_{inv}). Если потенциальный инвестор настроен благожелательно, склонен к сотрудничеству, то

времени на убеждение его принять участие в финансировании проекта обычно требуется меньше, нежели в других случаях: когда инвестор крайне осторожен в принятии решения или когда в его намерения входит изменение проекта предпринимателя или условий финансирования с выгодой для себя, но не для предпринимателя (агрессивная стратегия).

Поведенческий тип как инвестора, так и предпринимателя допустимо определять по формуле:

$$r = \frac{\sum_{t=1}^T s_t}{T}, \quad (2)$$

где s_t – состояние агента в момент t ;

T – длительность переговоров, под которой может подразумеваться, например, количество встреч агентов.

В [Мутовкина и др., 2013] были введены следующие семантические правила: агент является уклоняющимся, если $r \in (0; 0,5)$; агент называется компромиссным (сотрудничающим), если $r \in [0,5; 0,8]$ и агент будет принуждающим в случае $r \in (0,8; 1)$. Применительно к инвестору эти правила означают следующее: в первом случае инвестор по жизни – очень осторожный человек; он не склонен к доверию и живет по правилу: «Семь раз отмерь – один раз отрежь». Инвестор, относящийся ко второму поведенческому типу, считает, что шанс надо дать каждому. Такой человек всегда приветлив с окружающими, особенно – с потенциальными партнерами по бизнесу. Третий поведенческий тип характерен для людей, которые хотят получить все и сразу. Это непримиримые люди, постоянно находящиеся в роли «завоевателей». Вести переговоры с такими людьми трудно и зачастую – бесполезно вовсе, однако в случае, если они обладают значительными ресурсами, с ними все же приходится считаться.

В зависимости от того, к какому типу больше тяготеет инвестор, будут отличаться управляющие воздействия предпринимателя. Каждое состояние инвестора s_t , в котором инвестор находится в момент времени t , $t = 0, 1, 2, \dots, T$, принадлежит заданному конечному множеству возможных состояний $S = \{s_1, \dots, s_n\}$; при этом управляющее воздействие v_t в момент t является элементом множества $V = \{v_1, \dots, v_m\}$. Эволюция состояний инвестора во времени описывается уравнением:

$$s_{t+1} = f(s_t, v_t), \quad t = 0, 1, 2, \dots, T, \quad (3)$$

где f – заданная неслучайная функция, отображающая $S \times V$ в S , т.к. предприниматель

имеет определенное представление об инвесторе и знает, что от него ожидать.

Таким образом, $f(s_t, v_t)$ представляет собой последующее состояние для s_t при управляющем воздействии v_t . Предполагается, что в каждый момент времени t на управляющее воздействие накладывается расплывчатое ограничение L_t , являющееся расплывчатым множеством в V с функцией принадлежности $\mu_t(v_t)$. Действительно, беседу следует продолжать, исходя из того, в каком состоянии находится инвестор в момент времени t . Именно от этого зависит выбор предпринимателем средств и методов убеждения или же он приходит к выводу, что проще найти для переговоров другого инвестора. Цель предпринимателя интерпретируется как расплывчатое множество G_T в S , определяемое функцией принадлежности $\mu_{G_T}(s_T)$, где T – время окончания переговоров. Также предполагается, что предпринимателю известно начальное состояние инвестора s_0 . Задача заключается в нахождении вектора управляющих воздействий для максимизации вероятности достижения расплывчатой цели в момент времени T при условии, что должны быть выполнены расплывчатые ограничения L_0, \dots, L_{T-1} .

2. Алгоритм решения

Результирующее решение определяется пересечением заданных ограничений и цели, как было предложено в [Беллман и Заде, 1976, с. 193]:

$$D = L_0 \cap L_1 \cap \dots \cap L_{T-1} \cap G_T. \quad (4)$$

Следовательно, для функций принадлежности справедлива формула:

$$\mu_D(v_0, \dots, v_{T-1}) = \mu_0(v_0) \wedge \dots \wedge \mu_{T-1}(v_{T-1}) \wedge \mu_{G_T}(s_T), \quad (5)$$

где s_T может быть выражено как функция от v_1, \dots, v_{T-1} и s_0 путем последовательного применения соотношения (3) [Беллман и Заде, 1976]. Задача сводится к нахождению последовательности управляющих воздействий v_0, \dots, v_{T-1} , максимизирующей μ_D в формуле (5). Решение представимо в виде:

$$v_t = \psi_t(s_t), \quad t = 0, 1, 2, \dots, T-1, \quad (6)$$

где ψ_t – принятая модель поведения, т.е. принятое правило выбора управляющего воздействия v_t в зависимости от реализовавшегося s_t .

Для получения как ψ_t , так и максимизирующего (оптимального) решения v_0^*, \dots, v_{T-1}^* целесообразно

№ п/п	Усл. об.	Лингвистическое описание
1	v_1	стандартное описание проекта, его сильных, слабых сторон и полезности
2	v_2	описание проекта с использованием дополнительных доказательств его важности, в т.ч. – общественной значимости
3	v_3	предоставление предпринимателем дополнительных сведений о себе и проектах, которые уже были им успешно реализованы
4	v_4	демонстрация готовности пересмотреть, изменить проект в соответствии с пожеланиями инвестора
5	v_5	информирование потенциального инвестора о наличии других инвесторов и о возможности упустить существенную выгоду в случае отклонения инвестором данного проекта

применить метод динамического программирования. Так, используя (3) и (4), получаем:

$$\mu_D(v_0^*, \dots, v_{T-1}^*) = \max_{v_0, \dots, v_{T-1}} (\mu_0(v_0) \wedge \dots \wedge \mu_{T-2}(v_{T-2}) \wedge \mu_{G_{T-1}}(s_{T-1})), \quad (7)$$

$$\mu_{G_{T-1}}(s_{T-1}) = \max_{v_{T-1}} \left(\mu_{T-1}(v_{T-1}) \wedge \mu_{G_T}(f(s_{T-1}, v_{T-1})) \right),$$

где $\mu_{G_{T-1}}(s_{T-1})$ рассматривается как функция принадлежности расплывчатой цели в момент $t = T - 1$, индуцированная заданной целью G_T в момент $t = T$.

Повторяя процесс обратных итераций, получаем систему рекуррентных уравнений:

$$\mu_{G_{T-k}}(s_{T-k}) = \max_{v_{T-k}} \left(\mu_{T-k}(v_{T-k}) \wedge \mu_{G_{T-k+1}}(s_{T-k+1}) \right), \quad (8)$$

$$s_{T-k+1} = f(s_{T-k}, v_{T-k}), \quad k = 1, \dots, T,$$

которая дает решение задачи. Оптимальное решение получается путем последовательной максимизации величин v_{T-k} в (8), причем v_{T-k}^* определяется как функция от s_{T-k} , $k = 1, \dots, T$.

Инвестор до принятия решения об инвестировании должен определиться в собственной системе приоритетов, в качестве которых наиболее распространены такие критерии, как:

- общественная значимость проекта;
- влияние на имидж инвестора;
- соответствие целям и задачам инвестора;
- соответствие финансовым возможностям инвестора;
- соответствие организационным возможностям инвестора;

- рыночный потенциал создаваемого продукта (услуги);
- период окупаемости проекта;
- прибыль;
- уровень риска;
- соответствие проекта действующему законодательству и т.п. [Электронный ресурс, 2015а; Электронный ресурс, 2015b]

Для простоты в качестве обобщающего критерия можно применить какое-либо правило свертки приведенных критериев, например:

$$\tilde{Q}_j = \prod_{l=1}^w (\mu_{q_l}(m_j)) \rightarrow 1, \quad (9)$$

где $\mu_{q_l}(m_j)$ – функция принадлежности, определяющая степень соответствия инвестиционного проекта m_j объективному критерию q_l , $l = \overline{1, w}$; w – число критериев.

Экспериментально установлено, что при выполнении (9) и $r \in [0,5; 0,8]$ инвестор принимает проект к полному финансированию.

3. Пример

Потенциальный инвестор может находиться в одном из пяти эмоциональных состояний (табл. 1).

Таблица 1 – Основные эмоциональные состояния инвестора при переговорах

№ состояния	Условное обозначение	Лингвистическое описание
1	s_1	«совсем не доверяет»
2	s_2	«скорее не доверяет, чем доверяет»
3	s_3	«скорее доверяет, чем не доверяет»
4	s_4	«доверяет, но сомнения все же есть»
5	s_5	«полностью доверяет и готов сотрудничать»

Нечеткая цель предпринимателя – убеждение инвестора принять как можно более полное участие в инвестиционном проекте. Пусть агенты планируют три встречи и расплывчатая цель предпринимателя в момент $t = 3$ определяется функцией принадлежности μ_{G_3} , принимающей значения: $\mu_{G_3}(s_1) = 0,1$, $\mu_{G_3}(s_2) = 0,3$, $\mu_{G_3}(s_3) = 0,5$, $\mu_{G_3}(s_4) = 0,8$ и $\mu_{G_3}(s_5) = 1$. Т.е., в первом случае значение 0,1 говорит о том, что находясь в состоянии полного недоверия инвестор не будет принимать участие в проекте (вероятность его участия составляет 10%). Если же инвестор находится в состоянии s_5 , вероятность его участия

в проекте (или полнота финансирования) составляет 100%.

Предприниматель может оказывать на инвестора следующие воздействия (табл. 2).

Таблица 2 – Типовые воздействия предпринимателя на инвестора

Расплывчатые ограничения в моменты функции принадлежности $t = 0$, $t = 1$ и $t = 2$ в зависимости от начального представления предпринимателя о поведенческом типе (r_{inv}) потенциального инвестора (или о состоянии s_0) задаются функцией принадлежности μ_L . Так, для компромиссного типа инвестора значения ограничений приведены в табл. 3.

Таблица 3 – Значения ограничений в модели поведения предпринимателя для компромиссного типа инвестора

	μ_0	μ_1	μ_2
v_1	0,5	0,4	0,6
v_2	0,7	0,8	0,6
v_3	0,6	0,7	0,9
v_4	0,3	0,5	0,7
v_5	0,2	0,3	0,4

Таблица изменения состояний, задающая функцию f в формуле (3), соответствует табл. 4.

Таблица 4 – Функция изменения состояний инвестора компромиссного типа при переговорах

	s_1	s_2	s_3	s_4	s_5
v_1	s_1	s_3	s_4	s_5	s_5
v_2	s_2	s_3	s_4	s_5	s_5
v_3	s_3	s_4	s_5	s_5	s_5
v_4	s_2	s_2	s_4	s_3	s_4
v_5	s_1	s_1	s_2	s_4	s_3

Применяя выражение (7), можно найти функцию принадлежности расплывчатой цели в момент $t = 2$ (табл. 5):

Таблица 5 – Значения функции принадлежности μ_{G_2} при $t = 2$

	s_1	s_2	s_3	s_4	s_5
v_1	0,1	0,5	0,6	0,6	0,6
v_2	0,3	0,5	0,6	0,6	0,6
v_3	0,5	0,8	0,9	0,9	0,9
v_4	0,3	0,3	0,7	0,5	0,7
v_5	0,1	0,1	0,3	0,4	0,4

По каждой графе выбирается максимальное значение: $\mu_{G_2}(s_1) = 0,5$, $\mu_{G_2}(s_2) = 0,8$, $\mu_{G_2}(s_3) = 0,9$, $\mu_{G_2}(s_4) = 0,9$ и $\mu_{G_2}(s_5) = 0,9$, тогда соответствующее оптимальное решение имеет вид: $\Psi_2(s_1) = \Psi_2(s_2) = \Psi_2(s_3) = \Psi_2(s_4) = \Psi_2(s_5) = v_3$.

Если предприниматель начинающий, т.е. ранее не осуществлявший никакие проекты, его максимизирующая модель поведения имеет вид: $\Psi_2(s_1) = v_2$ и v_4 , $\Psi_2(s_2) = v_1$ и v_2 , $\Psi_2(s_3) = v_4$, $\Psi_2(s_4) = v_1$ и v_2 , $\Psi_2(s_5) = v_4$.

Аналогичные действия выполняются при $t = 1$ (табл. 6) и при $t = 0$ (табл. 7).

Таблица 6 – Значения функции принадлежности μ_{G_2} при $t = 1$

	s_1	s_2	s_3	s_4	s_5
v_1	0,4	0,4	0,4	0,4	0,4
v_2	0,8	0,8	0,8	0,8	0,8
v_3	0,7	0,7	0,7	0,7	0,7
v_4	0,5	0,5	0,5	0,5	0,5
v_5	0,3	0,3	0,3	0,3	0,3

Максимальное значение по каждой графе равно 0,8 и соответствующее оптимальное решение имеет вид:

$$\Psi_2(s_1) = \Psi_2(s_2) = \Psi_2(s_3) = \Psi_2(s_4) = \Psi_2(s_5) = v_2.$$

Таблица 7 – Значения функции принадлежности μ_{G_1} при $t = 0$

	s_1	s_2	s_3	s_4	s_5
v_1	0,5	0,5	0,5	0,5	0,5
v_2	0,7	0,7	0,7	0,7	0,7
v_3	0,6	0,6	0,6	0,6	0,6
v_4	0,3	0,3	0,3	0,3	0,3
v_5	0,2	0,2	0,2	0,2	0,2

Максимальное значение по каждой графе равно 0,7 и соответствующее оптимальное решение имеет вид:

$$\Psi_2(s_1) = \Psi_2(s_2) = \Psi_2(s_3) = \Psi_2(s_4) = \Psi_2(s_5) = v_2.$$

Таким образом, получено подтверждение того, что если инвестор относится к компромиссному типу поведения, то независимо от его начального состояния s_0 оптимальная модель поведения предпринимателя выглядит так: $\mu_D(v_0^*, \dots, v_{T-1}^*) = (v_2, v_2, v_3)$, причем соответствующее значение функции принадлежности $\mu_{G_0} = 0,7$.

Если предприниматель начинающий, то возможны следующие нечеткие модели его поведения (табл. 8).

Таблица 8 – Нечеткие модели поведения начинающего предпринимателя при обсуждении инвестиционного проекта за три встречи

Начальное состояние (s_0)	Стратегия: $t = 0, t = 1, t = 2$	
	Условные обозначения	Лингвистическое описание
s_1	v_2, v_2, v_2	Предпринимателю лучше сразу начать не со стандартного

	v_4	описания проекта, а использовать дополнительные доказательства его важности, в т.ч. – общественной значимости; при этом целесообразно продемонстрировать готовность пересмотра проект, его изменения в соответствии с пожеланиями инвестора
S_2	v_2, v_2, v_1 и v_2	Важным представляется «мягкое» убеждение инвестора в значимости проекта, используя дополнительные доказательства важности проекта, но и не забывая о слабых сторонах проекта, если таковые имеются. Необходимо продемонстрировать потенциальному инвестору правдивость и доверие
S_3	v_2, v_2, v_4	Практически не отличается от модели при состоянии S_1
S_4	v_2, v_2, v_1 и v_2	Совпадает с моделью при состоянии S_2
S_5	v_2, v_2, v_4	Совпадает с моделью при состоянии S_3

Заключение

Очевидно, что модели поведения предпринимателя при переговорах с инвестором других поведенческих типов будут отличаться от рассмотренных другими управляющими воздействиями и их последовательностями. Можно также представить вариант такой нечеткой ситуации, когда предприниматель должен убедить инвестора выделить на реализацию проекта всю необходимую сумму, а инвестор в зависимости от выполнения (9) и своих предпочтений может полностью профинансировать проект, частично или вовсе отказаться от участия в нем. Если предпринимателю известен поведенческий тип потенциального инвестора, то он может в некоторой степени предсказать, какую реакцию от инвестора следует ожидать при переговорах. Конечно, точность такого прогноза зависит от поведенческого типа самого предпринимателя. Поэтому в интересах предпринимателя вести переговоры с инвестором на оптимистичной ноте, соблюдая все правила делового этикета.

Теория нечетких множеств является одной из наиболее эффективных математических концепций, направленных на формализацию и обработку неопределенной информации и во многом интегрирующей известные подходы и методы. Модели поведения участников инвестиционного процесса, построенные на основе нечеткой логики, имеют ряд преимуществ перед игровыми моделями на основе вероятностного подхода. Вероятностный подход предполагает повторяемость опытов, т.е. осуществление многих однотипных проектов на достаточно большом временном интервале при неизменных условиях. На практике это

невыполнимо в отношении инвестиционных проектов в силу уникальности каждого из них. Нечеткие множества являются инструментом расчета возможностей, поэтому их применение может оказаться весьма полезным при составлении разного рода прогнозов, особенно – прогнозов поведения социально-экономических систем.

Работа выполнена при поддержке гранта РФФИ 13-01-00118-а «Нечеткие модели поведения лиц и групп, принимающих решения».

Библиографический список

- [Матвеев и др., 2005] Матвеев, А.А. Модели и методы управления портфелями проектов / А.А. Матвеев, Д.А. Новиков, А.В. Цветков. М.: ПМСОФТ, 2005. 206 с.
- [Мутовкина и др., 2014] Мутовкина, Н.Ю. Влияние целеустремленного поведения агентов на устойчивость многоагентной системы / Н.Ю. Мутовкина, В.Н. Кузнецов, А.Ю. Ключин // Системы управления и информационные технологии: науч.-техн. журнал. Воронеж: Изд-во «Научная книга», № 2 (56), 2014. С. 43 – 48.
- [Мутовкина и др., 2013] Мутовкина, Н.Ю. Семантическое определение типа агента в многоагентной системе. Проблема межагентного взаимодействия / Н.Ю. Мутовкина, А.Ю. Ключин, В.Н. Кузнецов // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): материалы III Междунар. науч.-техн. конф. (Минск, 21-23 февраля 2013г.) / редкол.: В.В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР, 2013. 592 с., С. 309 – 316.
- [Беллман и Заде, 1976] Беллман, Р., Заде, Л. Принятие решений в расплывчатых условиях / Р. Беллман, Л. Заде. В кн.: Вопросы анализа и процедуры принятия решений. М.: Мир, 1976. 230 с., С. 172 – 215.
- [Электронный ресурс, 2015a] Критерии и методы оценки эффективности инвестиционного проекта [Электронный ресурс] Дата обращения: 22.07.2015. Режим доступа: <http://www.financekind.ru/dlcs-327-1.html>.
- [Электронный ресурс, 2015b] Основные критерии отбора коммерческих проектов [Электронный ресурс] Дата обращения: 24.07.2015. Режим доступа: http://asi.ru/docs/nb_crit_upd.pdf.

A FUZZY MODEL OF THE BEHAVIOR OF THE ENTREPRENEUR IN NEGOTIATIONS ABOUT INVESTING IN THE CONDITIONS OF ACCURATE KNOWLEDGE OF NATURE A POTENTIAL INVESTOR

Palyukh B.V., Klyushin A.Yu., Mutovkina N.Yu.

*Tver State Technical University,
Tver, Russian Federation*

Pboris@tstu.tver.ru

klalex@inbox.ru

letter-boxNM@yandex.ru

The article discusses the behaviors of the entrepreneur during negotiations with the investor, the object of which is investment project. When choosing an investor is guided not only by objective evaluation criteria, but also their subjective preferences. The entrepreneur must take into account this fact and to build a conversation so as to convince potential investors to participate in financing the proposed project. How successful is the strategy of the entrepreneur depends on the result of negotiations.



УДК 681.5+004.896+004.8

СЕНСОРНАЯ МОДЕЛЬ ПОДРАЖАТЕЛЬНОГО ПОВЕДЕНИЯ РОБОТОВ

Карпов В.Э.

НИИ “Курчатовский институт”, Москва, Россия

karpov-ve@yandex.ru

В работе рассматриваются вопросы реализации моделей подражательного поведения робота. Предложена модель, основанная на наблюдении субъектом за поведением других членов группы. Показано, что результат наблюдения должен оказывать воздействие на поведенческую мотивацию агента. Приведены результаты имитационного моделирования предложенной схемы на примере решения простой поведенческой задачи. Показано, что целесообразность применения подражательного поведения далеко не однозначна и определяется сложностью решаемых задач и степенью когнитивных способностей субъекта.

Ключевые слова: роевая робототехника; модели социального поведения; подражательное поведение; языковое общение; семантическая сеть.

Введение

Вопросы применимости моделей социального поведения животных применительно к задачам организации управления в робототехнических системах обсуждается уже достаточно долгое время. Обзор [Fong, Nourbakhsh, Dautenhahn, 2003] уже представлял широкий спектр перспективных и уже решенных задач в этой области. Особый интерес к этой области проявляется в групповой робототехнике. Однако поставленные вопросы по-прежнему остаются весьма актуальными. Не в последнюю очередь это связано с отсутствием серьезных, концептуальных моделей или стратегических подходов, позволяющих группе роботов решать задачи, более сложные, нежели согласованное движение. Характерно, что в [Navarro, Matia, 2013] проблема распределения задач (Task Allocation) фигурирует как “опциональная”, вспомогательная задача, а речь идет в основном о задачах агрегации, распределения в пространстве, создания форм, согласованном движении и проч.

Таким образом, был сформулирован следующий тезис: реализация эффективных способов управления группами роботов может быть инспирировано моделями социального поведения в животном мире. Вопросы социального поведения применительно к созданию технических систем поднимались еще в классических работах Д.А. Поспелова, М.Г. Гаазе-Раппопорта, В.И. Варшавского [Варшавский, Поспелов, 1984], [Гаазе-Раппорт, Поспелов, 1987]. Однако

предлагаемые в этих работах модели относятся к категории “черных ящиков”, в них феномены социального взаимодействия рассматривались как есть, в феноменологическом смысле, т.е. вопросы составляющих этих моделей базисных элементов, равно как и генезис механизмов социального поведения, в них не рассматривался. Более конструктивным представляется рассмотрение моделей социального поведения с позиций этологии и зоопсихологии, которые активно занимаются типизацией и интерпретацией феноменов животного мира.

Считается, что типология важнейших видов социального поведения включает в себя три компоненты (см, например, [Мак-Фарленд, 1988], [Гудолл, 1992], [Тинберген, 1993], [Зорина, Полетаева, Резникова, 1999]):

1. Контагиозное поведение (т.н. “симпатическая индукция” или “заразное” поведение). Иницируется одним членом группы и охватывает всю группу, приводя к координированным действиям или поведению.

2. Агонистические поведение (“драка” у Тинбергена [Тинберген, 1993]). Считается, что поддержание структуры сообщества осуществляется именно на основе агонистического поведения (нападение, угрозы, подчинение и бегство).

3. Репродуктивное поведение. Основные составляющие – это брачное и родительское поведение, связанное прежде всего с процессом ухода и обучением (важнейший аспект с точки зрения групповой робототехники).

Существует некоторый ряд механизмов, реализующих эти формы поведения, обеспечивающий различные уровни социального поведения, и условий, при которых группу особей можно определить как социум. Это:

1. Наличие закрепленной за группой территории.
2. Постоянный состав группы (устойчивость во времени, прочность связей, агрессия к чужакам).
3. Когезия (стремление членов группы держаться вместе).
4. Персонификация в группе.
5. Система коммуникаций для взаимной координации и синхронизации деятельности членов группы.
6. Социальное доминирование.
7. Специализация (разделение функций).
8. Образование коалиций (кооперация).
9. Усвоение опыта старших поколений (обучение).

Реализация различных комбинаций этих механизмов позволяет получать различные уровни социализации у группы роботов. При этом одной из важнейших компонент, фигурирующей во многих этих механизмах, является механизм подражательного поведения. Именно феномен подражательного поведения ответственен за т.н. систему специальных адаптаций. Эта система включает в себя врожденный стереотип поведения, выражающийся в непрерывной ориентации на соседних особей, а также ярко выраженные подражательные реакции [Сотская, 2003]. Подражательное поведение служит для поддержания стада или стаи как устойчивого целого и обеспечения определенной синхронизации деятельности особей. То же самое касается реализации процедур обучения и в какой-то мере симпатической индукции (контагиозного поведения).

Подражательное поведение – это один из наиболее сложных для технической реализации механизмов. Основная проблема заключается в том, что система должна понимать, чему, собственно, следует подражать. Т.е. уметь выделять некий комплекс действий, совершаемых объектом для подражания, рассматривать этот процесс во времени и т.д. Частично эти проблемы созвучны тем, что имеются в процедуре обучения как такового. Например, даже в механизме простейшего стимул-реактивного поведения сакраментальным является вопрос о том, за что именно, т.е. за какое действие или поведения система получила сигнал поощрения или наказания.

В работе [Breazeal, Scassellati, 2002] ставятся следующие вопросы, касающиеся реализации подражательного поведения: (1) когда подражать? (2) чему подражать? (3) как отобразить наблюдаемые действия в поведенческие ответы? И (4) как роботу оценить свои действия, скорректировать ошибки, и определить, когда он достиг своей цели?

Ниже будет рассмотрен один из возможных механизмов реализации подражательного поведения, построенный на основе наблюдений и сопоставления наблюдаемых признаков с некоторой моделью.

1. Языковые схемы подражательного поведения

Уже говорилось, что основная проблема при реализации подражательного поведения – это понять, чему надо подражать. Фактически, это означает необходимость определения состояние объекта подражания и последующий переход субъекта в такое же состояние. Очень часто решение этой задачи основано на реализации некоторой системы коммуникаций между агентами. Например, используется следующий сугубо технический трюк: каждое действие или выполнение некоторой фиксированной поведенческой процедуры сопровождается выдачей (генерацией) некоторого символа или фразы. Иными словами, робот постоянно оповещает свое окружение разъяснением того, что он сейчас делает или в каком состоянии находится. Это – коммуникационный уровень общения.

Существуют подходы, основанные на реализации уже языкового взаимодействия между роботами. Например, в [Карпов, 2015] описывается знак-ориентированная система управления роботом (агентом), позволяющая реализовывать, в частности, феномен контагиозного поведения (примером которого является действие сигнала тревоги, заставляющего всю группу обратиться в бегство). Здесь следует отметить, что контагиозное поведение может рассматриваться как пример более общего феномена подражательного поведения.

Рассмотрим пример семантической сети, реализующей реакцию агента на некоторую опасность. Важной особенностью системы является наличие эмоциональной компоненты: выполнение действия агентом сопровождается некоторой эмоциональной оценкой – положительной или отрицательной. В основе эмоциональной компоненты СУ лежит т.н. потребностно-информационная теория эмоций П.В. Симонова [Симонов, 1982]. В этой теории предполагается, что эмоции являются оценкой текущей потребности (ее качества и ценности) и возможности ее удовлетворения. В предложенной знак-ориентированной системе управления именно эмоциональное состояние ответственно за генерацию фраз, т.е. робот начинает “говорить”, лишь испытывая сильные отрицательные эмоции.

Пусть в семантической сети системы управления имеется вершина “Опасность”, возбуждаемая неким комплексом сенсорных, вершин. Агент, получив сигнал опасности, в течение некоторого времени будет испытывать отрицательные эмоции (в силу цепочки “получен сигнал – надо убежать – опасность еще близка”). Это приведет к тому, что

будет сгенерирована фраза “Опасность”. Остальные члены группы воспринимают этот сигнал, т.е. для них происходит внешняя инициация вершины “Опасность” соответствующих сетей, несмотря на то, что соответствующие входные сенсорные вершины не возбуждены. Далее возбуждение передается на связанные с вершиной “Опасность” элементы, что в конечном итоге приведет к выполнению тех или иных двигательных функций. Эта ситуация приведена на (Рисунок 1).

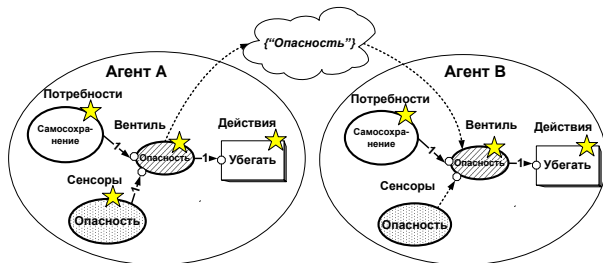


Рисунок 1 - Генерация и восприятие фразы

Здесь агент-инициатор *A* формирует фразу {“Опасность”}. Фраза воспринимается агентом-реципиентом *B*, при этом у реципиента *B* вершина “Опасность” возбуждается при отсутствии подтверждающего сигнала от соответствующего сенсора. Далее реципиент *B* выполняет действие “Убегать”.

Как видно, при такой языковой организации подражательное поведение формируется достаточно естественным образом. Однако в этой работе нас интересует подражательное поведение, основанное исключительно на сенсорном восприятии.

2. Сенсорная модель

Далее мы рассмотрим модель подражательного поведения, в которой не используются коммуникационные каналы, а реализуется схема непосредственного наблюдения, которую назовем сенсорной моделью.

2.1. Схемы поведения

Будем считать, что поведение агента (робота) описывается множеством продукций вида

$$C_1 \& C_2 \& \dots \rightarrow A$$

Подробности устройства таких правил не существенны. Это могут быть нечеткие правила, это могут быть варианты правил с использованием коэффициентов уверенности и проч. Важно, что имеется множество условий C_i и некоторое заключение *A*. Конъюнкт C_i может рассматриваться как некий наблюдаемый сенсорами робота объект. Причем в простейшем случае можно полагать, что чем ближе этот объект, тем выше уровень сигнала соответствующего сигнала C_i . В свою очередь заключение *A* может интерпретироваться как сигнал, запускающий некоторую поведенческую реакцию или процедуру – некий аналог командного нейрона.

Рассмотрим простую поведенческую схему, которая определяет проявление агрессии агента, направленную на некоторый объект *O*.

Правило агрессии (атаки) выглядит так:

$$O \& C_A \rightarrow A \quad (1)$$

Здесь *O* – наблюдаемый сенсорной системой агента объект, а C_A – некий контекст, мотивация агрессивного поведения агента. Условно эта схема изображена на (Рисунок 2).

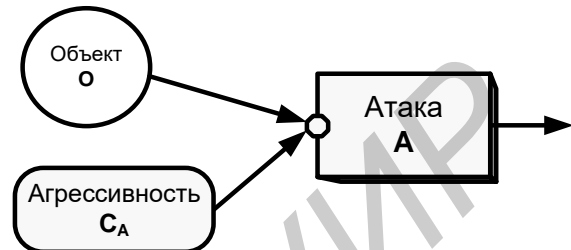


Рисунок 2 - Правило атаки

Выражение (1) определяет индивидуальное поведение, и подражательный аспект в нем отсутствует. Рассмотрим далее некоторое общее правило поведения некоторого субъекта *S*, трактуемое следующим образом: если некий субъект *S* наблюдает объект *O* и при этом объект находится рядом с субъектом, то *S* атакует *O*. Здесь также присутствует некий контекст агрессивности C_S субъекта.

Формально это можно записать так:

$$O \& S \& R(S, O) \& C_S \rightarrow A_S \quad (2)$$

Здесь $R(S, O)$ – некая функция, определяющая близость между субъектом и объектом. На (Рисунок 3) изображена схема соответствующего поведения.

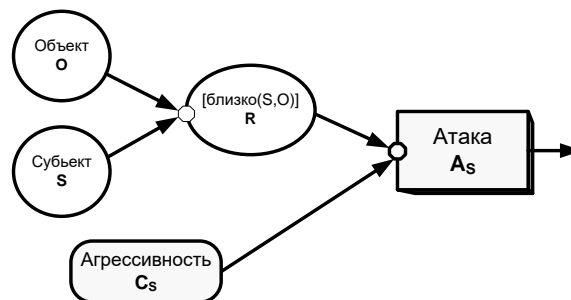


Рисунок 3 - Общее правило поведения субъекта

Структура, представленная на (Рисунок 3), определяет некую возможную схему наблюдения. Действительно, если сенсорная система агента-наблюдателя может зарегистрировать некий объект *O* и субъекта *S*, то этот агент может определить близость наблюдаемых *O* и *S* и сделать вывод о том, что *S* атакует *O*.

2.2. Подражательное поведение

Возникает вопрос: насколько общее правило поведения (2) применимо к нашей основной задаче реализации подражательного поведения. Непосредственное отождествление субъекта *S* и

некоторого субъективного “Я” не приведет к конструктивным результатам, несмотря на кажущуюся естественность этого шага. Здесь под субъективным “Я” будем понимать некоторый комплекс параметров, характеризующий состояние агента, включая его координаты, направление, скорость и т.п. Это несколько отличается от понятия субъективного “Я”, представленного, например, в [Карпов, 2012]

Рассмотрим следующую задачу. Пусть агент наблюдает некую удаленную пару O и субъекта S . Отождествление субъективного “Я” и S формально может рассматриваться как замена $R(S, O)$ на $R(\text{self}, O)$. Однако становится непонятно, на кого должно быть направлена процедура нападения. Объект O может и не находиться рядом с агентом-наблюдателем. Было бы логично, если бы агрессия агента была направлена не на удаленный объект O , а на тот объект, который обнаружен рядом. Этого можно добиться, если результат наблюдения (наблюдаемая сцена) будет оказывать влияние на поведенческую мотивацию агента – его контекст C_A .

Если принять, что результатом подражания поведения должна быть тождественность поведения, то, исходя из (1) и (2), мы получим следующую цепочку правдоподобных рассуждений.

Пусть $A=A_S$. Тогда

$$O \ \& \ C_A = O \ \& \ S \ \& \ R(S, O) \ \& \ C_S$$

Откуда получаем основное соотношение:

$$C_A = S \ \& \ R(S, O) \ \& \ C_S \quad (3)$$

Иными словами, наблюдаемая близость стороннего субъекта S и объекта O воздействует на контекст наблюдателя (мотивацию агрессивности в нашем примере). При этом важно, что действие агента будет направлено именно на наблюдаемый им – агентом – объект. Итоговая схема представлена на (Рисунок 4).

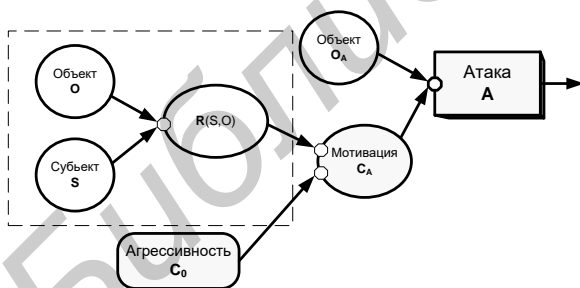


Рисунок 4 - Схема подражательного поведения

Итак, особенность схемы заключается в том, что агент не видит непосредственно действия другого субъекта, т.е. атаку. Агент наблюдает, что кто-то (субъект) взаимодействует с объектом. Именно это наблюдаемое взаимодействие (близость субъекта и объекта) расценивается как атака в силу имеющейся у агента модели такого поведения. Иными словами, агент сопоставляет наблюдаемую сцену с некоторой схемой.

3. Эксперименты

В качестве иллюстративного примера была рассмотрена следующая типичная задача. На тороидальной клеточной поверхности, на которой обитают агенты, размещалось некоторое количество “корма”. Агенты оснащены датчиками, регистрирующими в некоторой окрестности кормовые участки и способными обнаружить своих “собратьев”. Кроме того, агенты обладают способностью генерировать некий сигнал, также воспринимаемый другими агентами. При этом агент способен определить направление на источник сигнала. Задача заключается в том, чтобы, используя локальные правила поведения, как можно более эффективно собрать корм. На (Рисунок 5) приведен пример такого поля.

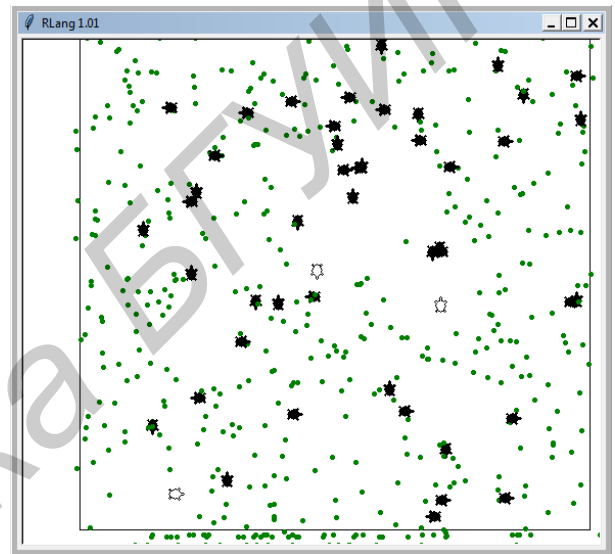


Рисунок 5 - Игровое поле

Были проведены серии экспериментов, в которых сравнивалась эффективность модели подражательного поведения с тестовой – той, у которой агенты при поиске корма использовали только случайное блуждание. Подражательность реализовывалась следующим образом: когда агент находил кормовой участок, он генерировал некое сообщение (сигнал), которое заставляло прочих агентов направляться к источнику сигнала. Эта упрощенная схема, в отличие от рассуждений, приведенных выше, необходима была исключительно с точки зрения получения сугубо статистических результатов.

На (Рисунок 6) приведены усредненные графики зависимости количества съеденного корма от времени. Серия R1 – это популяция, в которой реализован механизм подражательного поведения, R0 – поведение без подражания.

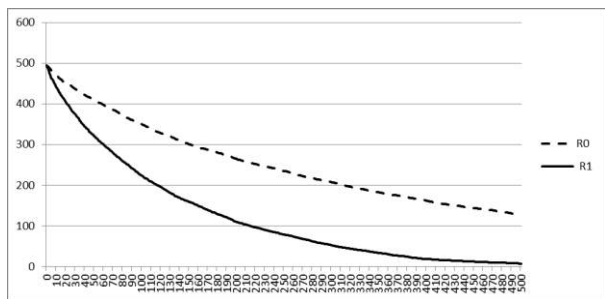


Рисунок 6 - Зависимость количества съеденного корма от времени

Из (Рисунок 6) видно, что эффективность схемы подражательного поведения значительно выше той, в которой агенты игнорируют состояние и поведение других членов группы.

На самом же деле эти эксперименты не доказывают ровным счетом ничего. Варьирование параметров среды или агентов для рассмотренной задачи может привести к результатам, когда обе схемы дадут и практически одинаковые результаты, и те, в которых подражательность ухудшает эффективность поиска. Причина заключается в том, что подражание – это механизм, присущий высокоорганизованному поведению, и его преимущества могут проявляться лишь при решении достаточно специфических, сложных задач. Рассмотренная же выше задача является сугубо модельной, иллюстративной лишь со статистической точки зрения. По сути, такие задачи сводятся лишь к тому, что у агентов увеличивается радиус окрестности обнаружения цели.

Более реальной выглядит схема поведения, представленная на (Рисунок 7).

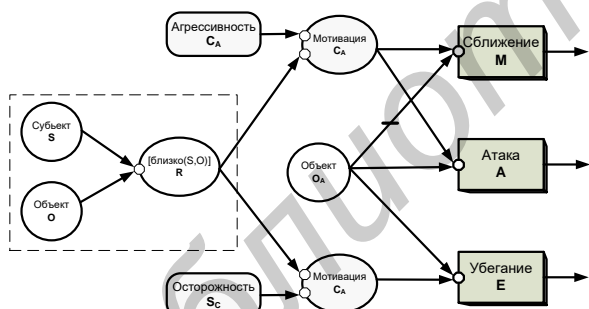


Рисунок 7 - Схема поведения, учитывающая мотивации агрессивности и осторожности

На этой схеме агент обладает двумя мотивациями – агрессивностью и осторожностью. Обнаружив некоторый объект, агент может проявить как агрессию (сближение, если объект далеко и атака, если объект близко), так и осторожность, т.е. убежать от объекта. Результаты моделирования такой системы становятся еще более неоднозначными и требуют уже иной формулировки задачи.

Заключение

Подражательное поведение – это один из механизмов, присущий высокоорганизованным

особям и являющийся основой для реализации феномена социальной организации. Возможно ли подражательное поведение для особей с ограниченными когнитивными способностями – это один из открытых вопросов. Равно как и вопрос, связанный с формальным определением условий, при которых подражательное поведение становится выгодным.

Зачастую термин “подражательное поведение” используется не совсем оправданно. Например, в [Chatty et al., 2011] описывается схема организации колонии муравьев, основанная на использовании т.н. когнитивных карт. Когнитивные карты являются как средством представления феромонных следов, так и способом реализации подражательного поведения. Суть механизма подражания заключается в том, что вероятность выбора того или иного действия агентом зависит от того, какое действие было выбрано другими агентами. А в [Chernova, Veloso, 2007] описывается механизм обучения на основе демонстраций. При этом суть обучения сводится к тому, что реализуется механизм классификации, построенный на модели гауссовой смеси (Gaussian mixture model). Целью обучения является сопоставление каждой точке признакового пространства одного из ограниченного числа действий агента. Декларируемое подражание по большому счету здесь заключается исключительно как подача серии обучающих примеров.

Проблема не в последнюю очередь заключается в необходимости определения того, что такое подражательное поведение и, как следствие, в чем его отличие от процедуры обучения. Не берясь за формальное определение, повторим, что очень важными аспектами подражательного поведения является выявление того, чему и когда, собственно, следует подражать.

В [Гаазе-Рапопорт, Поспелов, 1987] отмечается, что важнейшая задача подражательного поведения заключается в обучении, в формировании двигательных стереотипов, а реализация подражательного поведения требует сложной организации неровной системы. В частности, в памяти должны сохраняться наблюдаемая ситуация, представление о самом себе, представление о другом и мотивах его поведения. Механизм, идеологически близкий к этому подходу, и был предложен в данной работе. Основным допущением при этом было воздействие наблюдаемой сцены на мотивацию субъекта.

Работа выполнена при частичной финансовой поддержке гранта РФФИ № 15-01-07900. “Разработка и исследование моделей и методов непосредственной языковой коммуникации на основе семиотических моделей для реализации социального поведения в групповой робототехнике”.

Библиографический список

- [Breazeal, Scassellati, 2002] Breazeal C., Scassellati B. Challenges in Building Robots That Imitate People, in: K. Dautenhahn, C. Nehaniv (Eds.), *Imitation in Animals and Artifacts* // MIT Press. Cambridge, MA. 2002. С. 363–390.
- [Chatty et al., 2011] Chatty A. et al. Emergent complex behaviors for swarm robotic systems by local rules // IEEE SSCI 2011 Symp. Ser. Comput. Intell. - RIIS 2011 IEEE Work. Robot. Intell. Informationally Struct. Sp. 2011. С. 69–76.
- [Chernova, Veloso, 2007] Chernova S., Veloso M. Multiagent Collaborative Task Learning through Imitation // 4th International Symposium on Imitation in Animals and Artifacts. , 2007.
- [Fong, Nourbakhsh, Dautenhahn, 2003] Fong T., Nourbakhsh I., Dautenhahn K. A survey of socially interactive robots // Rob. Auton. Syst. 2003. Т. 42. № 3-4. С. 143–166.
- [Navarro, Matia, 2013] Navarro I., Matia F. An Introduction to Swarm Robotics // ISRN Robot. 2013. Т. 2013. С. 10.
- [Варшавский, Поспелов, 1984] Варшавский В.И., Поспелов Д.А. Оркестр играет без дирижера: размышления об эволюции некоторых технических систем и управлении ими. М.: Наука, 1984. 208 с.
- [Гаазе-Рапопорт, Поспелов, 1987] Гаазе-Рапопорт М.Г., Поспелов Д.А. От амёбы до робота: модели поведения, -М., 1987.
- [Гудолл, 1992] Гудолл Д. Шимпанзе в природе: поведение. Пер. с англ. М.: Мир, 1992. 670 с.
- [Зорина, Полегаева, Резникова, 1999] Зорина З.А., Полегаева И.И., Резникова Ж.И. Основы этологии и генетики поведения. М.: Изд-во МГУ, 1999.
- [Карпов, 2012] Карпов В.Э. Частные механизмы лидерства и самосознания в групповой робототехнике // XIII национальная конференция по искусственному интеллекту с международным участием КИИ-2012 (16-20 октября 2012 г., Белгород) Труды конференции. 2012. Т. 3. С. 275–283.
- [Карпов, 2015] Карпов В.Э. Об одной реализации знак - ориентированной системы управления мобильного робота // Искусственный интеллект и принятие решений. 2015. Т. 3. С. 53–61.
- [Мак-Фарленд, 1988] Мак-Фарленд Д. Поведение животных: Психобиология, этология и эволюция: Пер. с англ. М.: Мир, 1988. 520 с.
- [Симонов, 1982] Симонов П.В. Потребностно-информационная теория эмоций // Вопросы психологии. 1982. Т. 6. С. 44–56.
- [Сотская, 2003] Сотская М.Н. Зоопсихология: Учебно-методический комплекс в электронной форме (электронный учебник) по курсу “Зоопсихология и сравнительная психология” // Хрестоматия по зоопсихологии и сравнительной психологии: Учебное пособие МГППУ / Сост. М.Н. Сотская [Электронный ресурс]. URL: http://www.ido.edu.ru/psychology/animal_psychology/index.html.
- [Тинберген, 1993] Тинберген Н. Социальное поведение животных – *Social Behavior in Animals*, 1953 / Пер. с англ. Под ред. акад. РАН П. В. Симонова. М.: Мир, 1993.

SENSORY MODEL OF IMITATIVE BEHAVIOUR OF ROBOTS

Karpov V.E.

*National Research Centre “Kurchatov Insitute”,
Moscow, Russia*

Karpov-ve@yandex.ru

In work questions of realization of models of imitative behavior of the robot are considered. The model based on supervision by the subject over behavior of other members of group is offered. It is shown that the result of supervision has to make impact on behavioral motivation of the agent. Results of imitating modeling of the offered scheme on the example of the solution of a simple behavioral task are given. It is shown that expediency of application of imitative behavior isn't unambiguous and is defined by complexity of the solved tasks and degree of cognitive abilities of the subject.

Keywords: swarm robotics; social behavior modeling; imitative behavior; language interaction; semantic network.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89;94

ОНТОЛОГИЧЕСКИЙ ПОДХОД К ПОСТРОЕНИЮ ИНТЕРАКТИВНОЙ ВИРТУАЛЬНОЙ СРЕДЫ ДЛЯ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ ПЛАНИРУЕМЫХ ДЕЙСТВИЙ ПРИ ДИАЛОГОВОМ УПРАВЛЕНИИ РОБОТОМ- ПОМОЩНИКОМ КОСМОНАВТА НА МКС

Крючков Б.И.^{*}, Усов В.М.^{*}, Карпов А.А.^{**}

^{}Научно-исследовательский испытательный центр подготовки космонавтов имени Ю.А. Гагарина,
Звёздный городок Московской области, Россия*

b.kryuchkov@gctc.ru
v.usov@gctc.ru

*^{**}Санкт-Петербургский институт информатики и автоматизации Российской академии наук
(СПИИРАН), г. Санкт-Петербург, Россия*

karpov@iias.spb.su

Для обеспечения безопасности пилотируемых космических полетов применительно к внутрикорабельной деятельности (ВнуКД) экипажа на борту МКС предлагается использование интерактивных виртуальных сред на базе онтологического подхода для визуального представления планируемых действий робота-помощника космонавта (Р-ПК) во время речепроизводства голосовых команд человека-оператора (Ч-О). Представлен результат анализасовременных информационных технологий и средств для поддержки эргономического проектирования системы «космонавт – робот-помощник – рабочая среда».

Ключевые слова: робот-помощник космонавта (Р-ПК), диалоговое управление, голосовые команды, человек-оператор (Ч-О), виртуальные среды, 3D-визуализация, онтологии.

Введение

Практика подготовки космонавтов к выполнению пилотируемых космических полётов показывает выраженную тенденцию расширения применения виртуальных интерактивных сред (*англ.*: Virtual Interactive Environments, VIE) и технологий виртуальной реальности (*англ.*: Virtual Reality Technology, VRT): от настольных компьютерных обучающих систем до тренажеров. В русле задач дистанционного управления роботом-помощником космонавта (Р-ПК) при внутрикорабельной деятельности (ВнуКД) на МКС в работах [Крючков и др., 2014а; 2014б] рассматривается использование: 1) многомодальных человеко-машинных интерфейсов [Юсупов, 2013а; Карпов, 2012], и, в том числе, для управления Р-ПК в режиме диалога [Ющенко, 2009; Загорюлько, 2013; Крючков и др., 2014б] и 2) предложенной в работах [Тимофеев и др., 2006; 2008] технологии «внешнего наблюдателя» на базе VIE и VRT.

Применение указанных технологий является важным элементом обеспечения безопасности проектируемых полетных операций на МКС, поскольку при этом расширяются возможности Ч-О по выполнению мониторинга состояния рабочей

среды и контролю действий Р-ПК при внутрикорабельной деятельности (ВнуКД) экипажа. На современном этапе развития космической робототехники технологии VIE могут сыграть ключевую роль в составе исследовательских стендов для эргономического проектирования сложных систем «Ч-О – Р-ПК – рабочая среда».

1. Применение интерактивных визуальных сред при диалоговом управлении Р-ПК

1.1. Построение модели мира Р-ПК на основе технологий "обучение показом движения" для диалогового управления

Подход к построению «модели внешнего мира» Р-ПК является определяющим с точки зрения наделения искусственного интеллекта способностями к рациональному взаимодействию с Ч-О при выполнении сложных видов ВнуКД. Необходимо выполнять ряд условий, чтобы была обеспечена единая трактовка голосовых команд участниками диалога. Предполагается, что Р-ПК должен быть известен (то есть заранее запрограммирован) способ реализации

исполнительного действия, которое предписано командой. Вопросы предварительного формирования способа действий исследованы в работах Ф.М.Кулакова по методу «обучения показом движения» [Крючков и др., 2013]. Особый интерес представляет [Cantrell et al., 2011], в которой продемонстрировано создание новых сценариев взаимодействия на основе одновременного формирования голосовых инструкций роботу демонстрации способов выполнения действий.

1.2. Когнитивные составляющие взаимодействия интеллектуальных агентов при диалоговом управлении Р-ПК

Необходимость создания и исследования искусственных сред визуализации при управлении Ч-О динамическим объектом и при контроле сложно организованной рабочей среды с психологической точки зрения определяется визуализированным характером образных представлений человека, которые лежат в основе регуляции когнитивной активности Ч-О и формирования управляющих действий [Завалова и др., 1986]. Это положение справедливо для выполнения Ч-О контроля активности Р-ПК в рабочем пространстве МКС при Внутренней Экипажа. В работе [Ющенко, 2009] также указывается на «когнитивную составляющую, связанную с диалоговым управлением» (когнитивного робота). Сопоставление этих взглядов на когнитивный характер задачи контроля текущего состояния рабочей среды и планирование активности робота позволяет говорить о когнитивно-регулятивной основе построения взаимодействия в системе «Ч-О – Р-ПК – рабочая среда». При антропоцентрическом подходе к распределению функций в этой человеко-машинной системе диалоговое управление предлагается строить с использованием такой «модели внешнего мира» Р-ПК (как исполнительного звена системы), которая позволяет идентифицировать и выполнить все командные предписания Ч-О. Однако при этом космонавт (как центральное звено системы) должен быть способен сформировать цель действия, выбрать объект воздействия, спрогнозировать результат действия и проконтролировать на этой основе не только активность Р-ПК, но и правильность выдачи им самим предустановленных голосовых команд (в режиме самопроверки качества речепроизводства). Это важный элемент системы обеспечения безопасности пилотируемого полета. В этом заключается суть предлагаемой модели распределения функций в человеко-машинной системе и требуемый уровень искусственного интеллекта Р-ПК, достаточный для ведения диалога.

1.3. Применение многомодальных интерфейсов и речевых команд при управлении Р-ПК

Реализации диалогового управления Р-ПК в приведенных выше предположениях строится на основе речевого интерфейса, ориентированного на

использование Ч-О проблемно-ориентированного языка общения с роботом, который является основой для передачи команд (в том числе, в виде командных скриптов) в «верхний контур управления» Р-ПК, который собственно и предназначен для интерфейса с Ч-О [Ющенко, 2009; Загоруйко, 2013]. Авторы работ [Карпов, 2014; Ющенко, 2009; Крючков и др., 2014b] видят преимущество этого типа интерфейса в том, что он наиболее естественен для Ч-О, задача построения органов управления Р-ПК упрощается, поскольку она практически не требует формирования сложных двигательных навыков Ч-О и применения специальных конструкций органов управления типа многостепенных ручек, носимого костюма – «экзоскелета» и других устройств [Ющенко, 2009]. Методическая схема построения и реализации диалогового управления может быть условно разделена на две фазы (рис.1).

Для улучшения «включенности» Ч-О в текущую ситуацию при построении диалогового управления в общем случае необходимо помимо визуального представления виртуальной среды сохранить возможность предъявления Ч-О символично-графической информации, соответствующей уровню декларативного описания задач предметной деятельности по схеме: «полетная операция – варианты построения сценариев ее выполнения – состав объектов воздействия и циклограммы действий – начальное и конечное состояние реального мира до- и после выполнения операции».

Одновременно необходимо осуществлять текстовое отображение распознанной голосовой команды Ч-О с указанием найденного в базе данных соответствия для нее некоторой скриптовой команды для выполнения заранее сформированного действия. Другими словами, выполняется контроль полноты «модели внешнего мира» Р-ПК с точки зрения состава когнитивных задач, решаемых Ч-О, а именно: должно выполняться требование, чтобы структура голосовой команды учитывала онтологическое описание внешнего мира, включающего как «описание объектов для выполнения операции, так и пространственных отношений между объектами внешнего мира, включая сам робот» [Ющенко, 2009; Загоруйко, 2013]. На этом же уровне обеспечения надежности взаимодействия Ч-О с Р-ПК необходимо предусматривать средства контроля ошибок распознавания речи Ч-О при неблагоприятной акустической обстановке на МКС или из-за сбоев в процессе речепроизводства (оговорок, пропуска слогов, нечеткого произношения, неправильных сокращений и др., в условиях стресса).

В фазе ведения диалога с Р-ПК в распоряжении Ч-О должен иметься программно-аппаратный комплекс, предназначенный для распознавания голосовой команды Ч-О, тестового отображения на средствах электронной индикации результатов ее идентификации и соответствующего голосовой команде скрипта.

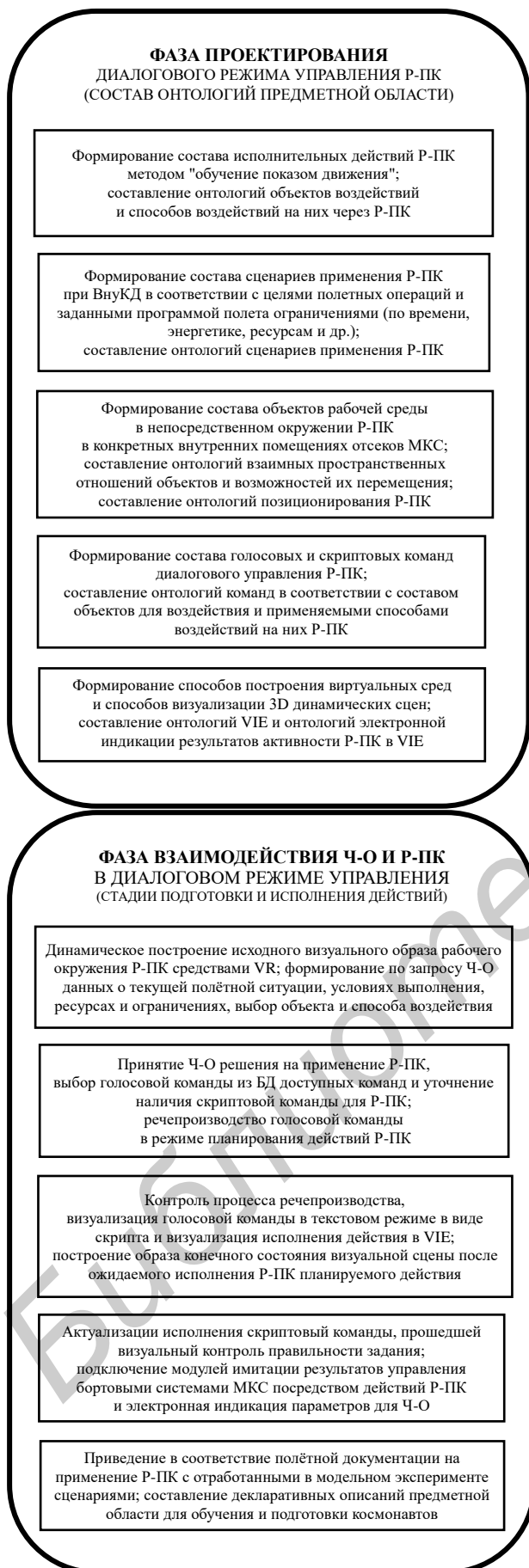


Рисунок 1 – Схема проектирования (вверху) и реализации (снизу) диалогового управления Р-ПК на базе онтологического подхода

В качестве системы автоматического распознавания русской речи можно использовать, например, систему СИРИУС, представленную в работах [Кипяткова и др., 2011; Karpov et al., 2014].

1.4. Построение онтологий VIE для исследования сценариев диалогового взаимодействия Ч-О с Р-ПК

Необходимо иметь в виду, что человеко-машинная система "Ч-О – Р-ПК" входит в состав более сложной организационно-технической системы (СОТС) «экипаж – МКС – ЦУП», а потому активность Р-ПК необходимо рассматривать не только в аспектах непосредственного взаимодействия с элементами окружения рабочей среды, но и ожидаемых последствий воздействий на органы управления бортовыми системами МКС.

Для выполнения этого условия могут использоваться различные средства информационной поддержки в рамках концепции «интеллектуального окружения» [Юсупов, 2013b, Крючков, 2014d], а также доступная телеметрическая информация, получаемая с высоко реалистической модели имитации функционирования МКС, например, с комплексного тренажера орбитальной станции. В этой связи в работах [Афанасьев, 2005; Подшивалов, 2005] вводится понятие «индуцированной виртуальной среды» (англ.: Induced Virtual Environment, IVE), функционирующей в режиме реального времени и воспроизводящей поведение моделируемой системы на основе поступающих данных от средств измерения и телеметрии от компонентов СОТС.

Именно технологии IVE наиболее интересны с точки зрения анализа безопасности выполнения полетных операций, когда в текущую ситуацию включена активность нового агента, Р-ПК.

В составе эргатических систем «Ч-О – Р-ПК – рабочая среда» Ч-О принимает решения на основе анализа всех поступающих данных из имитационной модели и передает команды управляемой системе (в данном случае, в контур управления верхнего уровня Р-ПК), и при этом Ч-О должен учитывать не только результат исполнения команды Р-ПК, но и последствия для состояния той реальной имитируемой системы, на звенья которой осуществлено воздействие. Можно пояснить этот нюанс, указав на значимые различия ситуаций в случаях:

- 1) перемещения/размещения Р-ПК внутри МКС какого-то груза;
- 2) воздействий манипуляторами робота на органы управления бортовыми системами (кнопки и тумблеры на пультах управления, включение и выключение, настройка контуров аппаратуры и др.) при ВнукД.

Для этого виртуальная на исследовательском стенде и (имитируемая на комплексном тренажере) реальная среды рассматриваются в единстве

текущего информационного обеспечения операторских задач контроля ситуации, а моделирование виртуальных объектов и явлений осуществляется в едином масштабе времени и в жесткой связи с теми явлениями и процессами, которые происходят в реальном мире.

Применение IVE преследует цель интеграции отображаемых Ч-О данных от датчиков и сенсоров, получающих информацию о процессах в реальном физическом мире, и визуализации 2-х типов данных, а именно, тех, которые:

- 1) синтезированы путем цифрового моделирования;
- 2) доступны наблюдению из реальной среды оптическими средствами.

Применение VIE в тех ограничениях, которые перечислены выше, требует [Крючков и др., 2014с]:

- 1) Разработки 3D-модели объектов виртуального мира.
- 2) Описания сценариев поведения для моделей объектов и возможного изменения отображения в виртуальной среде.
- 3) Формирования для контролируемых Ч-О объектов виртуального окружения Р-ПК «видимости» при конкретных положениях «внешнего наблюдателя» – определяется положение объектов относительно друг друга, их размер, повороты и другие необходимые атрибуты, исходя из разработанных онтологий.
- 4) Описания возможных сценариев влияния объектов друг на друга, в том числе коллизий и контактного взаимодействия.
- 5) Описания возможных изменений объектов виртуального мира при воздействии на них активных агентов (как самих членов экипажа, так и при задействовании манипуляторов Р-ПК).
- 6) Включения описаний действий каждого активного агента при его взаимодействии с объектами среды для реализации свойств интерактивности, исходя из соответствующего декларативного описания сценариев.

Очевидно, что для варианта ВнукД важную часть исходной информации для синтеза моделей объектов среды и индикации их позиционирования даёт априорно построенная цифровая карта всех внутренних помещений, в которой планируется осуществлять передвижения мобильного робота [Крючков и др., 2014а].

2. Возможности применения облачных виртуальных сред для проектирования интерфейсов системы «Ч-О – Р-ПК»

В настоящее время отмечается большой интерес к разработке инструментария для создания VIE, чтобы упростить их разработку и сопровождение. Для достижения этой цели может быть использован онтологический подход, в соответствии с которым «разработчики виртуальных сред формируют проект в терминах онтологии, который затем

интерпретируется в готовое приложение» [Грибова и др., 2012]. Данный подход может упростить решение ряда практически важных методических вопросов, ускорить разработку онтологий.

2.1. Поддержка командной работы специалистов различного профиля

Ввиду обширности предметной области профессиональной подготовки космонавтов, многоаспектности профессиональных задач экипажа при ВнукД и разнообразия состава объектов профессиональной рабочей среды, представляется актуальным при разработке и развитии комплекса технических средств подготовки космонавтов обеспечить коллективную работу специалистов многих специальностей в единой команде и на едином инструментарии, упрощающего создание и сопровождение VIE. Возможные решения в этой области показаны в работах ряда авторов в русле тематики разработки клиент-серверных облачных сервисов и приложений [Клещев и др., 2012; Афанасьев, 2005; Диков и др., 2012; Шукшунов и др., 2013].

Инструментарий для разработки виртуальных сред может быть реализован на облачной платформе IASaaS [Клещев и др., 2012], которая представляет собой программно-информационный интернет-комплекс для обеспечения поддержки разработки, управления и удаленного использования прикладных и инструментальных мультисервисных облачных сервисов (прежде всего интеллектуальных) и их компонентов. «Средствами платформы должен обеспечиваться контролируемый доступ всем участникам разработки к той функциональности, которая им необходима, а также предоставляются соответствующие средства разработки» [Грибова и др., 2012]. Исходя из потребностей развития комплекса технических средств подготовки космонавтов, а также, средств эргономического проектирования, рассмотрим далее преимущества такого инструментария.

2.2. Включение в проектирование VIE специалистов различного профиля и разделение работ между ними

Идеология создания готовых приложений с виртуальной реальностью, предложенная в [Грибова и др., 2012], предполагает технологию разработки VIE, в которой каждый этап разработки осуществляется специалистами различного профиля, и при этом каждому профилю специалистов предлагается соответствующий инструмент разработки. Этой работе выделены следующие основные участники процесса разработки: эксперты предметной области, графические дизайнеры, дизайнеры пользовательского интерфейса и программисты. При этом подчеркнуто, что отдельной задачей является обеспечение согласованности (полноты и непротиворечивости) всех компонентов VIE.

В ходе проектирования сценариев использования Р-ПК и человеко-машинного диалога необходимо получить единообразные описания состава объектов в модулях МКС, их свойств, отношений, возможных действий и команд Ч-О в VIE, а также результаты, к которым могут привести эти действия. На этой основе формируется визуальное представление VIE и определяется логика ее работы.

В соответствии с современным подходом к разработке интеллектуальных систем, база знаний должна разрабатываться и сопровождаться экспертами предметной области с помощью редакторов знаний [Грибова и др., 2012]. В этой работе предлагается выделить знания экспертов в отдельную компоненту и предусмотреть программное средство (редактор), управляемый унифицированной онтологией виртуальной среды для управления экспертными знаниями. Декларативное представление знаний, отделенное от логики работы технической системы, по мнению авторов, позволяет на протяжении жизненного цикла их модифицировать и уточнять, так как экспертные знания являются самым «подвижным и живым» компонентом любой системы.

2.3. Интеграция инструментария платформы IASaaS внешними программными системами

Как следует из работы [Клещев и др., 2012], платформа IASaaS имеет механизм работы с внешними (удаленными) программными системами, включая программный интерфейс (API), который может быть использован для интеграции программного обеспечения, например, такие как сервис распознавания речи ([Карпов, 2014; Ющенко, 2009; Крючков и др., 2014b]), сервис аппаратного управления роботами и др.

Заключение

По результатам анализа проблемы выбора возможных решений в данной работе предлагается:

- 1) применение для диалогового управления роботом управляющих данных, сформированных заранее на основе предопределенных голосовых команд, которые соответствуют набору предусмотренных сценариев использования Р-ПК, описаниям типовых ситуаций активности Р-ПК в рабочей среде и объектов для воздействия;
- 2) предварительное построение виртуальной сцены с визуализацией для Ч-О в виде 3D динамического образа планируемого действия Р-ПК и сцены рабочего окружения;
- 3) построение модели внешнего мира Р-ПК на основе онтологий, посредством задания отношений работа с объектами рабочей среды, в том числе, в категориях безопасности для Ч-О и СОТС в целом;
- 4) динамическое построение визуального образа рабочего окружения Р-ПК (в части взаимодействия с объектами) на основе мультисенсорного измерения параметров, которые доступны из реальной среды,

применения средств телеметрии и оптического наблюдения.

При разработке онтологий предметной области предполагается использовать базы данных декларативных описаний полетных операций, представленных в бортовой документации МКС с тем существенным уточнением, что для интерактивного управления Р-ПК необходим тщательный отбор фрагментов инструкций и описаний, которые непосредственно относятся к сфере диалогового управления Р-ПК и допускают его применение в конкретных динамических режимах пилотируемого полета [Соловьёв и др., 2012; Кравченко и др., 2013].

Благодарности

Исследование выполняется при частичной финансовой поддержке Совета по грантам Президента РФ (проект № МД-3035.2015.8) и фонда РФФИ (проект № 15-07-04415-а).

Авторы также выражают признательность В.В.Грибовой и ее сотрудникам за внимание к данной работе, конструктивную критику и содействие в написании статьи, в частности, предоставленные материалы для раздела 2.

Библиографический список

- [Афанасьев, 2005] Афанасьев, В.О. Модели структур и данных для решения задач управления поведением и взаимодействием объектов в индуцированной виртуальной среде / В.О. Афанасьев // Космонавтика и ракетостроение, №2(39), Изд-во ЦНИИМаш, 2005. С.168-179.
- [Грибова и др., 2011] Грибова, В.В. Разработка виртуального мира медицинского компьютерного обучающего тренажера / В.В. Грибова, М.В. Петряева, Л.А. Федорищев // Дистанционное и виртуальное обучение. №9, 2011. С.56-66.
- [Грибова и др., 2012] Грибова, В.В. Обучающие виртуальные системы и средства их создания / В.В. Грибова, Л.А. Федорищев // Вестник информационных и компьютерных технологий. 2012. №3. С. 48-51.
- [Диков и др., 2012] Диков, Д.А. Подход к проблеме удаленной визуализации сложных 3D-моделей на базе поVNC-решений / Д.А.Диков, Г.И.Радченко // Вестник Нижегородского университета им. Н.И. Лобачевского 2012. №5(2). С.328-331.
- [Завалова и др., 1986] Завалова, Н.Д. Образ в системе психической регуляции деятельности. / Н.Д. Завалова, Б.Ф.Ломов, В.А. Пономаренко. М.: Наука, 1986, 173 с.
- [Загорулько, 2011] Загорулько, Ю.А. Моделирование работа, управляемого речевыми сигналами. / Ю.А. Загорулько // Известия Томского политехнического университета. 2011. Т. 319. №5. С.98-102.
- [Карпов, 2012] Карпов, А.А. Когнитивные исследования ассистивного мультимодального интерфейса для бесконтактного человеко-машинного взаимодействия / А.А. Карпов // Информатика и ее применения. 2012. Т.6. № 2. С.77-86.
- [Карпов, 2013] Карпов, А.А. Ассистивные информационные технологии на основе аудиовизуальных речевых интерфейсов // Труды СПИИРАН, 2013, Вып. 27, С.114-128.
- [Кипяткова и др., 2011] Кипяткова, И.С. Опыт разработки модели распознавания речи со сверхбольшим словарем / И.С. Кипяткова, А.А. Карпов // Материалы Международной научно-технической конференции «Открытые семантические технологии

проектирования интеллектуальных систем» OSTIS-2011, Минск, Беларусь, 2011, С. 409-412.

[Клещев и др., 2012] Клещев, А.С. Облачная платформа для разработки и управления интеллектуальными системами / Клещев А.С., Грибова В.В. [и др.]; // OSTIS-2011: Тр. конф. Минск, 2011. С.5-14.

[Кравченко др., 2013] Кравченко, С.И. Виртуальные руководства научными космическими экспериментами на борту международной космической станции / Е.И. Жук, С.С. Обыденков, С.И. Кравченко, В.В. Степанов, А.С. Потоцкая // Программные продукты и системы. 2013 год, №3 С.72-77.

[Крючков и др., 2013] Крючков, Б.И. Создание моделей виртуальной реальности, как способ обучения космонавтов взаимодействию с роботом – помощником экипажа и как условие определение потенциальных областей его полезного применения / Б.И. Крючков, В.М. Усов // Труды международной научно-технической конференции «Экстремальная робототехника», СПб: Политехника-сервис. 2013. С.230-244.

[Крючков и др., 2014а] Крючков, Б.И. Перспективные подходы к применению сервисных роботов в области пилотируемой космонавтики / Б.И. Крючков, А.А. Карпов, В.М. Усов // Труды СПИИРАН. 2014. Вып. 32. С.125-151.

[Крючкови др., 2014б] Крючков, Б.И. Организация речевого взаимодействия человека-оператора с антропоморфным мобильным роботом для ведения пространственной ориентировки в условиях невесомости / Б.И. Крючков, А.А. Карпов, В.М. Усов // Проблемы управления и моделирования в сложных системах. Сб. трудов XVII Международной конференции – Самара, Самарский научный центр РАН, 2015 С.522-527.

[Крючков и др., 2014с] Крючков, Б.И. Технологии моделирования для эргономического проектирования системы «космонавт – манипуляционный робот – рабочая среда» / Крючков Б.И., Михайлюк М.В., Усов В.М. // Материалы конференции «Управление в морских и аэрокосмических системах» (УМАС-2014). СПб.: ОАО «Концерн «ЦНИИ «Электронприбор», 2014. С.367-377.

[Крючков и др., 2014д] Крючков, Б.И.. Перспективные направления робототехники для поддержки социально значимых сфер активности человека (на примере сервисных роботов в пилотируемой космонавтике) / Б.И.Крючков, А.А.Карпов, В.М.Усов // Электронный научно-практический журнал «ИнноЦентр» Вып. № 3(4) ноябрь 2014 // Электронный ресурс, URL: <http://innoj.tversu.ru/number4.html>

[Подшивалов, 2005] Подшивалов, А.Ю. Использование индуцированной виртуальной среды для развития концепции управления с обратной связью. / А.Ю. Подшивалов // Электронный журнал "Исследовано в России", 2005, № 38, С.411-418.

[Соловьёвы др., 2012] Соловьёв, В.А. Текущее состояние и перспективы развития системы управления полетами космических аппаратов (окончание). / В.А. Соловьёв, В.Е. Любинский, Е.И. Жук // Пилотируемые полеты в космос 2012, №3(5). С. 29-33.

[Тимофеев и др., 2006] Информационные технологии и мехатронные устройства для интеллектуальных медицинских систем. / А.В. Тимофеев [и др.]; // Информационно-управляющие системы, 2006, № 4. С.45-49.

[Тимофеев и др., 2008] Тимофеев, А.В. Медицинские аспекты разработки систем человеко-машинного взаимодействия с использованием моделей виртуальной реальности для нейрохирургии. / А.В. Тимофеев [и др.]; // Труды СПИИРАН. 2008. Вып. 6. С.184-196.

[Шукшунуви др., 2013] Шукшунув, В.Е. Основы разработки и создания интегрированных учебно-тренажерных комплексов для оснащения центров подготовки специалистов по управлению динамическими объектами / Шукшунув В.Е., Янюшкин В.В. // Программные продукты и системы. № 3, 2013. С.3-10.

[Юсупов и др., 2013а] Юсупов, Р.М. Возможности применения многомодальных интерфейсов на пилотируемом космическом комплексе для поддержания коммуникации космонавтов с мобильным роботом – помощником экипажа. / Р.М. Юсупов, Б.И. Крючков, А.А. Карпов, А.Л. Ронжин, В.М. Усов // Пилотируемые полеты в космос. 2013. № 3. С.23-34.

[Юсупов и др., 2013б] Юсупов, Р.М. Создание «интеллектуального окружения» на пилотируемом космическом комплексе для позиционирования мобильного робота – помощника экипажа / Юсупов Р.М. [и др.] // Актуальные проблемы психологии труда, инженерной психологии и эргономики. Вып. 5. / Под ред. А.А.Обознова, А.Л. Журавлева. М.: Институт психологии РАН. 2013. С. 397-422.

[Ющенко, 2009] Ющенко, А.С. Диалоговое управление роботами с использованием нечетких моделей // Интегрированные модели и мягкие вычисления в искусственном интеллекте. / А.С. Ющенко // Сборник трудов V-й международной научно-практической конференции (Коломна, 28-30 мая 2009 г.). М.: Физматлит, 2009. Т.1. С.97-108.

[Cantrell et al., 2011] Cantrell, R. Learning actions from human-robot dialogues / by Rehj Cantrell, J. Benton, Kartik Talamadupula, Subbarao Kambhampati, Paul Schermerhorn, Matthias Scheutz // In: Proc. IEEE, 2011, URL: http://www.researchgate.net/publication/224256250_Learning_actions_from_human-robot_dialogues

[Karpov et al., 2014] Karpov, A. Large vocabulary Russian speech recognition using syntactico-statistical language modeling / A.Karpov, K.Markov, I.Kipyatkova, D.Vazhenina, A.Ronzhin // Speech Communication. Elsevier, 2014, Vol. 56, pp. 213-228.

AN ONTOLOGICAL APPROACH FOR DESIGNING INTERACTIVE VIRTUAL ENVIRONMENTS FOR A VISUAL REPRESENTATION OF PLANNED ACTIONS DURING DIALOGUE CONTROLLING A ROBOT-ASSISTANT ON-BOARD OF THE ISS

Kryuchkov B.I. *, Usov V.M. *, Karpov A.A. **

* Yu. A. Gagarin Research & Test Cosmonaut Training Center, Star city, Moscow region, Russia

b.kryuchkov@gctc.ru
v.usov@gctc.ru

** St. Petersburg Institute for Informatics and automation of the Russian Academy of Sciences (SPIIRAS), Saint-Petersburg, Russia
karpov@ias.spb.su

To ensure the safety of manned space flights for purposes of IVA of cosmonauts on the ISS is proposed the use of virtual interactive environments (VIE) on the basis of the ontological approach for visual representation of the planned actions of the robot-assistant of cosmonauts (R-A) during the speech production of voice commands of a human operator (H-O). An analysis of available information technologies to support the ergonomic design of the systems "cosmonaut – robot-assistant – working environment" is presented.

Keywords: robot-assistant (R-A) in intra-vehicular activity (IVA) of cosmonauts, dialog management, voice commands, Human-Operator (H-O), virtual interactive environment (IVE), 3D-visualization, ontology.



УДК 004.822:514

ПРИМЕНЕНИЕ ДАННЫХ ТЕЛЕМЕТРИИ ДЛЯ ЦЕЛЕЙ УПРАВЛЕНИЯ НЕШТАТНЫМИ СИТУАЦИЯМИ НА БОРТУ СПУТНИКОВЫХ СИСТЕМ

Данилин Н.С.^{*}, Димитров Д.М.^{**}, Сабиров И.Х.^{**}

^{*}ОАО «Российские космические системы»,

г. Москва, Россия

n_danilin@mail.ru

^{**}ЗАО «Космос Комплект»,

г. Москва, Россия

office@dimitrov.ca

В работе предлагается подход к формированию пространства признаков на базе телеметрических данных спутниковой системы с применением ROC-анализа для нужд ситуационного управления, которое реализуется бортовой системой интеллектуального мониторинга программно-аппаратной среды.

Ключевые слова: интеллектуальный мониторинг; ситуационное управление; анализ данных телеметрии.

1. Введение

Современные спутниковые системы и их миссии занимают важное место в бизнесе, науке и военном деле. Требования к 10-, 15-летнему сроку активного существования аппаратов ставят остро вопрос о разрешении нештатных ситуаций в режиме реального времени без участия оператора. В связи с этим ведущие производители космических аппаратов начали эффективно применять интеллектуальные системы мониторинга и управления нештатными ситуациями в реальном времени на борту [Staraub, 2009], [Girimonte, 2012]. Проводимые ЗАО «Космос Комплект» работы по этой тематике привели к задаче формирования пространства признаков на потоке данных телеметрии, при помощи которого возможно с достаточной точностью выявление нештатных ситуаций бортовой системой мониторинга в режиме реального времени.

2. Обработка потока телеметрических данных

Для выборки данных из потока телеметрической информации необходимо использовать некоторые новые механизмы. В связи с этим выбрана следующая модель обработки [Gama, 2007], [Bonneti, 2001], [Abadi, 2003]:

- Поток представляет собой последовательность кадров с фиксированной

реляционной схемой (список атрибутов - F). Последовательность кадров в потоке не ограничена.

- Каждый кадр связан с временем появления t в потоке. Кадр может быть сохранен или пропущен.

- Для каждого потока данных можно определить «окно», т.е. множество кадров, которые появились в определенном интервале времени T (t и $t-T$), либо окно может покрывать каждый кадр. Окно (W) можно представить как $W[F, n]$, где F описание кадра, а n – число последних поступивших с телеметрии кадров, либо как $W[F, T, m]$, где T, m - время выборки кадров в интервале m единиц времени. Например, выражение $W[F, 1000]$ описывает окно в потоке телеметрии, в котором происходит выборка последних 1000 кадров, а $W[F, T, 10]$ – окно при котором выборка кадров происходит в течении 10 единиц времени.

Механизм «окно» позволяет преобразовать поток телеметрических данных в набор кадров [Димитров, 2009]. Система управления данными (СУД), как часть интеллектуальной мониторинговой системы, принимает поток данных из телеметрии и вместе с этим и два типа заявки (W) – постоянные и «горячие». Постоянные заявки сохраняются как формализованные описания нештатных ситуаций, а горячие – как запросы со стороны оператора на Земле [Димитров, 2014]. Представленная модель обработки потока телеметрических данных положена в основе разработанной в рамках проекта «Сколково» распределенная интеллектуальная программно-аппаратная среда «БИС».

2.1. Формирование пространства признаков и проверка нормального распределения данных

Формирование пространства признаков - это операция, которая упрощает анализ выбранных данных. Она позволяет подобрать только те кадры, которые содержат наиболее полезную, с точки зрения определения внештатной ситуации, информацию [Семерджиева, 2013].

Другая существенная проблема это редукция пространства признаков. Существует множество методов такой редукции как метод главных компонент, вейв-трансформации и др [Начев, 2012]. Эти методы дают возможность определить главные направления представления множества данных, что позволяет сформировать их собственное подпространство (компрессия данных) и тем самым уменьшить размерность признакового пространства.

Практика работы с данными телеметрии показала, что они являются случайными величинами (измерения параметров бортовой программно-аппаратной среды) с нормальным распределением. Известно, что при нормальном распределении данных 99.7% их значений находятся в интервале $(m-3\sigma, m+3\sigma)$ и тем самым можно принять, что все значения вне этого интервала можно удалить из выборки.

2.2. ROC анализ

ROC анализ (Receiver Operating Characteristic Analysis) - это классическая методология в теории сигналов [Egan, 1975], которая широко применяется при анализе данных в диагностических системах [Fawcett, 2006].

Предполагается, что располагаем двумя классами объектов P и N (штатные и нештатные множества ситуаций соответственно). Задача в том, что бы классификатор мониторинговой системы определял правильно к какому множеству P или N относится каждый элемент кадра F.

ROC анализ дает возможность сравнивать различные варианты классификаторов и при выборе подходящей точки ROC кривой также определить индекс чувствительности и точность классификатора. На этом этапе разработана программа, которая работает с двумя диагностическими модальностями. Следующая версия программы будет работать с несколькими диагностическими модальностями, что позволит расширить значительно точность распознавания нештатных ситуаций.

3. Система БИС

БИС (бортовая интеллектуальная среда) является программно-аппаратной средой на которой реализована обработка данных телеметрии для нужд ситуационного управления.

3.1. Программная среда БИС

Программная среда БИС включает три подсистемы (рис.1):

- Обработка потока телеметрических данных (DMP).
- Формирование пространства признаков (INFP).
- Управление взаимодействием с системой мониторинга, с базовыми процессами БИС и с каналами связи (IOP).

Обобщенная блок-схема программной среды БИС показана на рис.1.

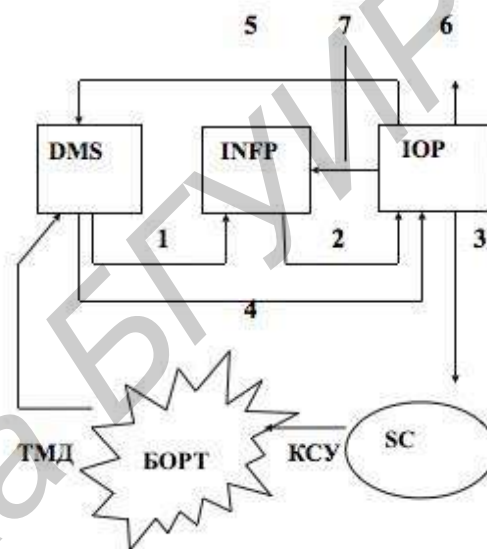


Рисунок 1- Блок-схема программной среды БИС

Обозначения на рис.1

DMS – Процесс управления обработки данных из телеметрического потока на базе механизма «Окно» в одном из двух режимов W [F,n] или W[F,Tm] по заявкам от системы мониторинга (1). Выбор режима зависит от скорости потока и объема выборки данных по каждому окну и может проводится предварительно, либо по команде с мониторинговой системы (5).

INFP – Процесс формирования пространства диагностических признаков на базе ROC-анализа (2) и (7).

IOP – Процесс выполняет: формирование заявок с атрибутами по команде от системы мониторинга (7), поддерживает обменные взаимодействия с системой управления ситуациями (3), фильтрует поток телеметрических данных (4), осуществляет обмен информации по радиоканалам (6).

SC – Подсистема бортового ситуационного управления.

КСУ – Команды от системы ситуационного управления.

БОРТ – Программно-аппаратная среда на борту спутника.

ТМД – Телеметрические данные.

3.2. Аппаратная реализация БИС

Аппаратная реализация БИС включает три специализированных процессора (системы в корпусе) и коммутатор с интерфейсом SpaceWire (рис.2)

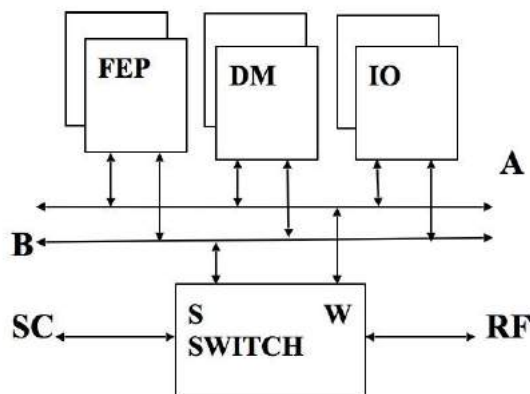


Рисунок 2 – Блок-схема аппаратной реализации БИС

Обозначения на рис. 2

FEP – Процессор ARM9 для работы с потоком данных и формирования окон (W).

DM – Процессор ARM9 для формирования пространства диагностических признаков и анализа потока кадров.

IO – Процессор ARM9 для реализации обменных взаимодействий в системе обработки потока данных телеметрии, а также для взаимодействия с системой ситуационного управления и радиоканалами.

SW SWITCH – Коммутатор 16 x 16, поддерживающий интерфейс Space Wire.

Процессорные модули подключены к общей шине и к коммутатору.

Для отладки функционирования БИС и для проведения экспериментов с различными потоками телеметрических данных, на базе существующего у компании SSE Congregation симулятор, был разработан имитатор потоков телеметрической информации (ПОТОК-С) [SSE Corp, 2013]. Имитатор также позволяет моделировать потоки информации, которые трудно практически воспроизвести.

Поток телеметрической информации имитатора генерируется в соответствии со стандартом IRIG 106-13 [IRIG Standard, 2013]. ПОТОК-С позволяет имитировать различные сбои в формировании кадровой информации и системы передачи данных. Применение указанного стандарта вызвано его

широким применением не только в космической промышленности, но и в других смежных областях. Структура кадра достаточна универсальная, чтобы можно было реализовать широкий класс алгоритмов обработки потока данных [Douglas, 2003].

Заключение

Одним из первых и необходимых шагов при проектировании спутников со сроком эксплуатации 10- 15 лет является создание бортовой системы управления нештатными ситуациями. Источником информации о состоянии борта является телеметрическая информация. Распознавание нештатной ситуации определенного класса на основе анализа потока телеметрических данных сегодня решается многими мировыми производителями и уже запущено на орбите несколько реализаций таких проектов. ЗАО «Космос Комплект» также реализовала первую версию своих систем обработки потока данных и ситуационного мониторинга на программно-аппаратной среде БИС.

Для отладки системы БИС и создания среды для разработки и тестирования соответствующего программного обеспечения, дополнительно к системе БИС разработан симулятор потока телеметрических данных ПОТОК-С.

БИС можно применять на любом спутнике, в т.ч. и на микро- и нано спутниках, благодаря своим потребительски ориентированным модулям, которые выполнены по технологии «система в корпусе». К моменту публикации доклада проводятся лабораторные испытания на реальном симуляторе микроспутника EyaSat X4.

Библиографический список

- [Staraub, 2009] Staraub, J. A review of spacecraft AI control systems/J.Straub//JPL, Pasadena, 2009.
- [Grimonte, 2012] Grimonte, J.AI for space applications/D.Grimonte, D.Izzo//ESA, ESTEC, 2012.
- [Gama, 2007] Gama, J. Learning from data streams/ J.Grama, M.Gater//Springer, New York, 2007.
- [Bonneti, 2001] Bonneti, J.Towards sensor data base systems/P.Bonneti, J.Gehrke, P.Seshadri// Proceedings of the 2nd International Conference on Mobile Data Management, January, 2001.
- [Abadi, 2003] Abadi, d. Aurora: a data stream management system/D.Abadi, D.CXarney, U.Cetintemel// Proceedings of the ACM SIGMOD International Conference on Management of Data, San Diego< CA, 2003.
- [Димитров, 2009] Димитров, В. Модел на потоци от данни / В.Димитров // Научни трудове на Русенския университет,2009,том 48, серия 6.1.
- [Димитров, 2014] Димитров, Д.М. Интеллектуальная система управления на борту отечественных спутников:фантастика или реальность/ Н.С.Данилин, Д.М.Димитров, И.Х.Сабилов // Материалы международной научно-технической конференции OSTIS-2014, БГУИР, Минск, С 217-220.
- [Семерджиева, 2013] Семерджиева, В. Анализ на данни от диагностични тестове / В.Семерджиева, Б.Георгиев, Ч.Дамянов // Научни трудове на УХТ, том LX, Пловдив, 2013.
- [Начев, 2012] Начев В. Интелигентни системи/ В.Начев, Ч.Дамянов, Т.Титова // Академично издателство на УХТ, Пловдив, 2012.
- [Egan, 1975] Egan, J. Signal detection theory and ROC analysis/ J.Egan // Academic Press, New York, 1975.

[Fawcett, 2006] Fawcett, T. An introduction to ROC analysis/ T.Fawcett // Pattern recognition Letters, 27, 861-874, New York, 2006.

[CCE Corp , 2013] CCE Corporation. TELSVM Telemetry Simulator// Ostin, 2013.

[IRIG Standard, 2013] IRIG Standard 106-13//US Army White Sand Missile Range, 2013.

[Douglas, 2003] Douglas, C. Virtual telemetry for dynamic data driven application simulations/C. Douglas//ICCS Proceedings, Springer, Berlin, 2003.

TELEMETRY DATA USAGE IN ON-BOARD EMERGENCY SITUATIONAL CONTROL

Danilin N.S., Dimitrov D.M., Sabirov I.H.

JSC «Russian Space Systems», Moscow, Russia

N_danilin@mail.ru

SC «Cosmos Complect», Moscow, Russia

office@dimitrov.ca

An approach to diagnostic vector sets generation based on the telemetry ROC data analysis is suggested in this paper. The generated diagnostic vector sets are used by on-board intellectual monitoring system for emergency situation recognition and control in real time.

Introduction

The satellite customers today require 10-15 years active life. The serious problem is on-board emergency situation overcoming. Some of them could not be solved during the limited time period by the operators. Additionally, Earth control is too expensive. To improve on-board efficiency an intelligent on-board monitoring system is needed. One of the main problems for design and implementation of such systems is the question of telemetry data flow using in order to generate diagnostic vector sets for emergency case recognition and control. The possible solution discussion of that problem is the goal of the presented paper.

Main Part

An approach to analysis of telemetry data flow is presented. The main steps in it are defined as follows:

- Flow data extraction and diagnostic vector sets generation (windows and frames)/
- Normal distribution of real time data checking.
- Minimization of extracted data.
- ROC analysis.

The architecture (block diagrams) and connected building blocks of the corresponded software-hardware environment (BIS) are described.

A telemetry data flow simulator POTOC-S was designed in order to debug and improve system BIS and also for different variants of telemetry flow generation. POTOC-S implements the telemetry standards IRIG 106-13.

Conclusion

The recognition of wide range of on-board emergency situations is a priority task for new satellite projects. Company “Cosmos Complect” prepared its first variant of data telemetry analysis and intelligent monitoring system BIS.

A telemetry data flow simulator POTOC-S was additionally designed for the needs of BIS debugging and application software implementation.

The hardware product called BIS can be used with any satellite, including micro and nano satellites, because of its specialized building modules which are designed as a systems-in-package.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ СОСТОЯНИЯ КОСМИЧЕСКИХ АППАРАТОВ

Юрыгина Ю.С.* , Лахин О.И.* , Донсков А.В.** , Мишурова Н.В.** , Сысоев Д.В.** , Скорюпина Е.Г.*

*ООО «Научно-производственная компания «Разумные решения»

г. Самара, Россия

yurygina@smartsolutions-123.ru

**ОАО Ракетно-космическая корпорация «Энергия» имени С.П. Королева,

г. Королёв, Россия

post@rsce.ru

Рассматриваются методы и средства по созданию системы прогнозирования нештатных ситуаций на борту пилотируемых космических аппаратов. Для решения задач система использует концептуальную и математическую модели космического аппарата. Применение системы поможет специалистам главной оперативной группы управления своевременно обнаруживать отклонения от штатного режима работы и принимать соответствующие меры.

Ключевые слова: концептуальная модель, онтология, мультиагентные технологии, кибер-физическая модель, нештатная ситуация, ПКА «Союз».

Введение

Совершенствование пилотируемых космических аппаратов (ПКА) включает расширение диапазона контролируемых параметров, что увеличивает сложность оперативного контроля состояния и движения ПКА. Аномальное состояние ПКА может возникнуть вследствие длительного медленно протекающего процесса на борту. В таких ситуациях параметры контроля могут находиться в допустимых диапазонах, поэтому распознавание подобных отклонений от нормального состояния тяжело идентифицировать. Для повышения точности определения процессов, происходящих на борту, достоверной оценки будущего состояния ПКА необходимо создание автоматизированной системы оперативного контроля состояния ПКА.

1. Обзор существующих решений

Среди решений в ракетно-космической отрасли, можно выделить интеллектуальную систему, разрабатываемую в Санкт-Петербургском научно-инженерном центре [НИЦ ЭТУ, 2015], которая осуществляет мониторинг и контроль состояния космодронов. В задачи системы входит определение состояния оборудования по показаниям датчиков, прогноз изменения значений параметров для выявления опасных тенденций, предпосылок

развития нештатных ситуаций. В случае обнаружения отклонений в техническом состоянии объекта формируются и выдаются рекомендации для принятия решений. В настоящее время система осуществляет комплексный мониторинг ракетно-космической техники и наземной космической инфраструктуры космодрома Плесецк.

В промышленной среде широко используются решения, осуществляющие мониторинг факторов, влияющих на безопасность производства. Наиболее известными являются: ProcessVue Alarm Management Suite компании MAC Solutions, Великобритания [MAC Solutions, 2015], DynAMO Alarm Suite компании Honeywell, США [Honeywell, 2015], Alarm Management System компании Dräger, Германия [Dräger, 2015], Nellcor™ SatSeconds компании Covidien, Ирландия [Covidien, 2015]. Среди недостатков данных систем можно выделить подробное документирование событий, возникающих в системе, что вызывает необходимость хранения большого количества избыточных данных. Обмен информацией осуществляется по единой шине данных, которая в случае возникновения непредвиденной ситуации является потенциальным узким местом системы и др.

Предлагаемый в данной статье подход отличается способом представления модели

сложного технического объекта (ПКА). В отличие от рассмотренных ранее решений, в предлагаемом подходе модель сложного технического объекта описывается при помощи семантической сети (онтологии). Это обеспечивает гибкость и легкость масштабирования системы, не требует перепрограммирования при внесении изменений: добавлении новых технических средств, новых типов нештатных ситуаций, изменении предметной области.

Следующее отличие состоит в методе построения рекомендаций при возникновении нештатных ситуаций. Многие системы работают на основе регламентированных инструкций, внесенных в базу данных. Некоторые системы сохраняют для дальнейшего использования алгоритм решения, информацию о событиях, произошедших на контролируемом объекте. Это ограничивает область решаемых задач кругом ситуаций, имеющихся в базе системы. Данное ограничение устраняется за счет использования мультиагентных технологий, когда на базе имеющихся в системе знаний агент может получить новые знания и построить алгоритм выхода из непредвиденной ситуации. Взаимодействие между агентами децентрализовано, что свидетельствует об отсутствии узких мест в системе.

2. Постановка проблемы

В процессе выполнения основных динамических режимов - выведение, манёвр, сближение и спуск - высока вероятность отказов элементов бортовых систем, последствия которых могут привести к срочному спуску ПКА. Причинами несвоевременного выявления предпосылок возникновения нештатных ситуаций при оперативном управлении могут стать:

- необходимость совершенствования средств контроля и воздействия на ход полета;
- большое количество разнородных знаний;
- необходимость определения гарантированного запаса времени;
- необходимость оперативного принятия решений в условиях ограниченного времени;
- сложность планирования полета.

Основной целью разработки системы является сокращение времени на анализ состояния космического аппарата, причин возникновения аномальных процессов, подготовка рекомендаций по дальнейшей работе космического аппарата. Это предоставит специалистам главной оперативной группы (ГОГУ) Центра управления полетами помощь в систематическом наблюдении, анализе и своевременном принятии решений по предупреждению и парированию нештатных ситуаций.

Основными задачами, которые будет решать автоматизированная система, являются:

- распознавание отклонения от штатного состояния;
- наблюдение тенденции к возникновению нештатной ситуации, распознавание нештатных ситуаций;
- формирование рекомендаций по предотвращению и парированию нештатных ситуаций;
- отслеживание, фиксирование и сохранение выполненных действий по ликвидации нештатных ситуаций (НШС) на основе анализа телеметрической информации, докладов специалистов);
- прогнозирование состояния ПКА на следующем витке или на заданный период при штатной работе;
- анализ риска отказа взаимосвязанных систем при отклонении от штатной работы или тенденции к возникновению нештатной ситуации;
- прогнозирование развития НШС на ближайшие витки;
- моделирование и оценка состояния ПКА в реальном времени.

3. Предлагаемый подход

Для решения поставленных задач система использует концептуальную и математическую модель ПКА. Концептуальная модель отражает физическую, функциональную структуру корабля, штатные и нештатные состояния, правила по предупреждению и парированию нештатных и аварийных ситуаций. При помощи математической модели на основе нечетких множеств выполняется оценка текущего состояния ПКА в условиях постоянного изменения параметров.

Задачи анализа, распознавания и прогноза возникновения нештатных ситуаций, а также формирование рекомендаций по их предупреждению и парированию выполняются мультиагентной системой, которая использует концептуальную и математическую модели для получения данных о состоянии ПКА.

3.1. Концептуальная модель

Основными элементами концептуальной модели являются:

- физическая модель (объекты: «ТПК Союз», «СОЖ», «СОТР», «Гидросистема», «Автоматика СТР», «Средства вентиляции» «Влагосборник», «Блок управления приводом вентилятора» и др.; отношения: «состоит из систем», «состоит из оборудования»);
- модель штатного состояния электроприборов («Штатное состояние уровня влажности для работы электроприборов», «Штатное состояние температуры для работы электроприборов», и др.);
- модель нештатных ситуаций;
- правила по предупреждению и парированию нештатных ситуаций («Откачка

конденсата», «Проверить скафандры и частично разгерметизировать СА», «Досрочный спуск» и др.)

Для распознавания отклонения от штатного состояния используется описание ожидаемых штатных и допустимых диапазонов значений параметров. Следовательно, значения, не входящие в данный диапазон, будут попадать в нештатный диапазон значений.

Используя описания штатных, допустимых и нештатных диапазонов значений показателей телеметрической информации (ТМИ), система должна отслеживать тенденции к выходу из штатного диапазона, а также рассчитывать время достижения нештатных границ (рисунок 1).

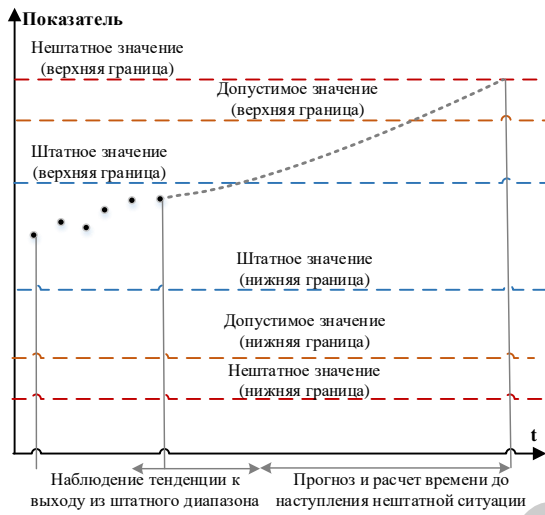


Рисунок 1 – Наблюдение изменения показателя ТМИ

3.2. Математическая модель

Для прогноза состояния космического аппарата может использоваться математическая модель, основанная на применении нечетких множеств. Для этого выделяются основные показатели, определяются диапазоны значений и вычисляются взаимосвязи между различными показателями ТМИ.

В таблице 1 представлены диапазоны значений основных показателей (температура, уровень влажности, временной отрезок с момента доклада экипажа об откачке конденсата, объем конденсата).

Таблица 1 – Диапазоны значений показателей

Лингв. переменная	Лингвистическая шкала	Числовой диапазон
Температура атмосферы внутреннего объема спускаемого аппарата (СА)	Низкая	10÷13 °С
	Ниже средней	14÷17 °С
	Средняя	18÷23 °С
	Выше средней	24÷27 °С
	Высокая	> 28 °С
Уровень влажности во внутреннем	Малый	0÷2 мм. рт. ст.
	Ниже среднего	3÷7 мм. рт. ст.
	Средний	8÷11 мм. рт. ст.
	Выше среднего	ст.

объеме СА	Большой	12÷15 мм. рт. ст. > 16 мм. рт. ст.
Временной отрезок с момента доклада экипажа об откачке конденсата	Малый	0÷0.5 ч
	Ниже среднего	0.5÷2.5 ч
	Средний	2.5÷3.5 ч
	Выше среднего	3.5÷5 ч
Количество конденсата	Большой	> 5 ч
	Малое	0÷5 мл
	Ниже среднего	6÷11 мл
	Среднее	12÷17 мл
	Выше среднего	18÷24 мл
	Большое	> 25 мл

На основе таблицы 1, составляется матрица отношений, при помощи которых определяются зависимости между различными показателями:

$$M_{ij}^1 = \begin{matrix} H \\ HC \\ C \\ BC \\ B \end{matrix} \begin{pmatrix} H & HC & C & BC & B \\ t_{11} & t_{12} & t_{13} & t_{14} & t_{15} \\ t_{21} & t_{22} & t_{23} & t_{24} & t_{25} \\ t_{31} & t_{32} & t_{33} & t_{34} & t_{35} \\ t_{41} & t_{42} & t_{43} & t_{44} & t_{45} \\ t_{51} & t_{52} & t_{53} & t_{54} & t_{55} \end{pmatrix}$$

Рисунок 2 – Пример матрицы отношений

Определенные зависимости позволяют составлять прогноз значения показателя, например, количества конденсата (таблица 2).

Таблица 2. Прогнозируемые значения показателя образовавшегося конденсата в зависимости от значений других показателей среды

Температура атмосферы внутреннего объема СА	Уровень влажности во внутреннем объеме СА	Время с момента старта (временной отрезок)	Количество образовавшегося конденсата	
			Лингв. оценка	Числ. диапазон
Н	Н	В	Н	0÷0.5 мл
Н	С	В	НС	6÷11 мл
Н	В	С	НС	
НС	ВС	С	НС	
НС	ВС	С	С	12÷17 мл
С	ВС	С	ВС	
С	ВС	НС	ВС	
ВС	В	НС	ВС	18÷24 мл
В	ВС	М	В	
В	ВС	В	В	> 25 мл

Отклонение одного показателя может повлиять на ряд связанных систем ПКА, которые могут также выйти из штатного состояния через определенное время. Так, уровень влажности влияет на следующие системы (рисунок 3):

- СОТР (система обеспечения теплового режима);
- СУС (система управления спуском);
- СЭП (система электропитания);
- СОЖ (система обеспечения жизнедеятельности);
- ПК Нептун (пульт космонавтов).

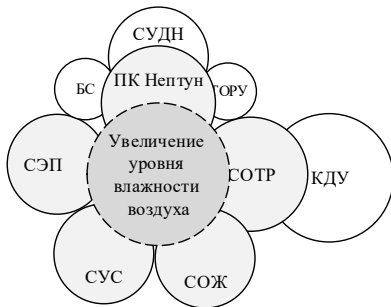


Рисунок 3 – Пример связанных подсистем ПКА

3.3. Кибер-физическая модель

Для упрощения восприятия и анализа текущего состояния ПКА предлагается использовать кибер-физическую модель или 3D модель космического аппарата. Здесь отражается не только физическая структура, но и результаты анализа, выполненного системой: узлы корабля, задействованные в нештатной ситуации, и узлы, находящиеся в зоне риска.

3.4. Мультиагентная модель

Для каждого узла ПКА назначается программный агент [Матюшин и др., 2013]. Агент узла отслеживает показатели среды, влияющие на штатную работу узла. При регистрации динамики, приближающей выход показателя из штатного диапазона, агент узла выполняет расчет располагаемого времени и отправляет сообщения агентам функционально связанных узлов.

Для формирования рекомендаций по нормализации показателей агент узла анализирует правила и процессы, влияющие на изменение показателей, оценивает возможность выполнения действий по запуску данных процессов и формирует рекомендации по выполнению необходимых действий [Вакурина и др., 2014].

4. Основные модули автоматизированной системы

На рисунке 4 представлены основные модули системы в информационном контуре управления космическими аппаратами: модуль анализа ТМИ, модуль прогнозирования, модуль формирования рекомендаций по предотвращению и парированию НШС, модуль моделирования и оценки состояния КА после предполагаемых действий.

В реальном времени на вход модуля анализа ТМИ поступают данные ПКА. Модуль анализа ТМИ регистрирует предпосылки к возникновению НШС, либо отклонения от штатного режима

работы, модуль прогнозирования рассчитывает возможные последствия, модуль моделирования оценивает состояние КА после выполнения рекомендаций, сформированных системой.

Из других систем, входящих в контур управления (например, Автоматизированная система планирования (АСП)), поступают данные о программе полета, зонах видимости и т.д.

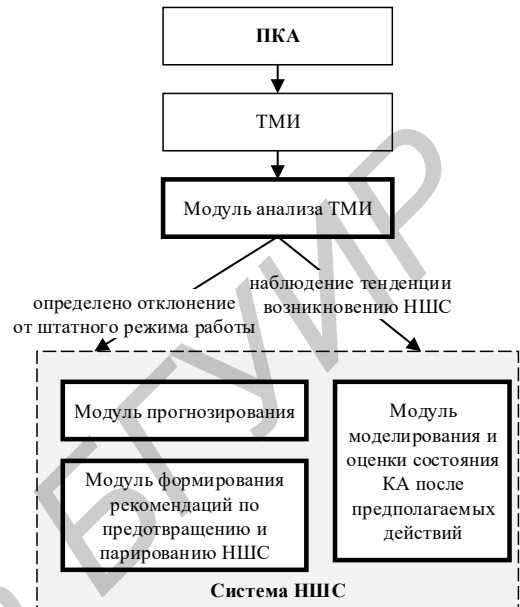


Рисунок 4 – Основные модули системы

Обработанные системой данные поступают специалисту для дальнейшего принятия решений и анализа текущей ситуации. На рисунке 5 приведен общий сценарий работы системы.



Рисунок 5 – Сценарий работы системы

5. Описание примера использования

Рассмотрим основные принципы использования системы на примере возникновения нештатной ситуации останов основного вентилятора В1 холодно-сушильного агрегата– телеметрическое обозначение РВ3, во время автономного полета ПКА «Союз 224» 27 марта 2009 г. Для упрощения и

наглядности в ходе рассмотрения примера некоторые детали были опущены.

5.1. Анализ нештатной ситуации наземными специалистами

По рекомендации ГОГУ экипаж отключает питание вентиляторов В1, В2 и выдает команду на включение вентилятора В1. Устойчивой работы вентилятора В1 не зафиксировано. На витке 21 было проведено тестовое включение вентилятора В1 на фоне работающего вентилятора В2. Вентилятор В1 включился, вышел на нормальный режим работы, однако, обороты ротора стали уменьшаться, поэтому экипажу была выдана рекомендация «отключить вентилятор В1».

По возвращении ПКА на Землю специалистами был проведен анализ ситуации для определения причины возникновения НШС. Причиной отказа вентилятора В1 ХСА во время автономного полета ПКА «Союз 224» стало попадание конденсата атмосферной влаги в расточку электродвигателя. Исследование состояния электродвигателя выявило совпадение причины отказа с отказом электродвигателя БК1425 во время полета корабля «Союз 223». Наиболее вероятной причиной попадания конденсата в полости электродвигателя является нерегулярность откачки экипажем конденсата из влагосборника ХСА.

5.2. Сценарий работы системы по предотвращению возникновения НШС

Для предотвращения возникновения НШС система сигнализирует экипажу о необходимости откачки конденсата из влагосборника ХСА. Ожидаемый результат требует выполнения следующих задач:

- фиксирование отклонений от штатного состояния – повышение концентрации влажности;
- прогноз возможных последствий – отказ электрических приборов;
- расчет располагаемого времени – время до отказа электрических приборов;
- формирование рекомендаций по нормализации показателей – уменьшению концентрации влажности;
- формирование рекомендаций по парированию НШС.

Перед началом полета в систему поступают следующие исходные данные:

- текущее состояние ПКА;
- программа полета, сеансы связи.

Во время полета в систему поступает телеметрическая информация, а также доклады экипажа и специалистов. На основании данных система формирует онтологическую модель ситуации, которая позволяет выполнять комплексный контроль состояния КА.

Рассмотрим решение поставленных задач для достижения ожидаемого результата:

- фиксирование отклонений от штатного состояния – повышение уровня влажности.

На основе ТМИ ПКА на витке 19 модуль анализа ТМИ регистрирует изменение параметров:

- температура. Т65, Т66: снижение температуры с 21 до 17 °С в отсеке СА; Т64: стабильное состояние 23-24 °С;
- уровень влажности. УВСА: повышение с 14,3 до 15,9 мм.рт.ст.

Штатный диапазон показателей СОЖ во время режима «маневр»:

- температура в СА: [20, 23] °С;
- уровень влажности: [3, 15] мм.рт.ст.

Штатный диапазон показателей СОТР во время режима «маневр»:

- температура Т65, Т66 [18, 23] °С;
- температура Т64 [20, 23] °С.

Примечание. Системы СОТР и СОЖ имеют дополнительный набор параметров, который для упрощения не рассматривается.

Результат работы модуля анализа ТМИ:

- уровень влажности превысил штатные значения, необходимые для системы СОЖ;
- температура стала ниже штатного значения, необходимого для системы СОТР.

- прогноз возможных последствий – отказ электрических приборов;

Для штатной работы электроприборов параметры температуры и уровня влажности должны лежать в определенных диапазонах. При регистрации тенденции повышения уровня влажности система выполняет прогноз возможных последствий: выход из строя электроприборов, пороговые значения которых могут быть превышены.

На основе данных о штатном режиме работы электродвигателя вентилятора (температура, уровень влажности) модуль прогнозирования устанавливает возможные последствия НШС: выход из строя вентилятора в составе СОТР, используемого для откачки конденсата.

- расчет располагаемого времени – время до отказа электрических приборов;

На основе динамики повышения уровня влажности и пороговых значений штатного функционирования электродвигателя вентилятора система выполняет расчет располагаемого времени до возникновения нештатной ситуации.

- формирование рекомендаций по нормализации показателей – уменьшению концентрации влажности;

Для предотвращения возникновения НШС модуль формирования рекомендаций анализирует правила по предотвращению роста показателей

уровня влажности и требует выполнить откачку конденсата (в результате выполнения операции показатель концентрации влажности снизится). Другой способ нормализации концентрации влажности - «проверка скафандров и частичная разгерметизация СА» указан в системе как крайняя мера предотвращения НШС, поэтому данная рекомендация будет предложена только при отсутствии других возможных способов решения.

Если располагаемого времени не хватит для предотвращения возникновения НШС, система рекомендует продолжение полета с контролем температурно-влажностного режима.

Заключение

Система, разработанная на основе предлагаемого подхода, позволит повысить оперативность реагирования на изменение текущего состояния ПКА, повысит эффективность управления полетом, как в штатном режиме, так и в случае возникновения НШС. Результаты обработки ТМИ поступают специалисту для дальнейшего принятия решений и анализа текущей ситуации, что уменьшает влияние человеческого фактора.

Ожидается получение экономического эффекта за счет снижения рисков отказа ПКА и повышения показателей надежности.

Система поможет специалистам ГОГУ Центра управления полетами решать задачи комплексного контроля и анализа состояния ПКА, а также своевременно принимать решения по предупреждению и парированию нештатных ситуаций.

Библиографический список

[НИЦ ЭТУ, 2015] Научно-инженерный центр электротехнического университета. Интеллектуальные системы мониторинга и контроля состояния технически сложных объектов. <http://www.nicetu.ru/resheniya-i-produkty/intellektualnye-sistemy-monitoringa-i-kontrolja-sostojanija-tekhnicheski-slozhnykh-obektov/> (дата обращения: 20.11.2015).

[MAC Solutions, 2015] ProcessVue Suite Overview <http://www.processvue.com/en/products/processvue-suite-overview> (дата обращения: 13.11.2015).

[Honeywell, 2015] HoneyWell Alarm Management. https://www.honeywellprocess.com/en-US/online_campaigns/alarm-management/Pages/alarm-management.html (дата обращения: 13.11.2015).

[Dräger, 2015] Dräger Alarm Management System. http://www.draeger.com/sites/en_seeur/Pages/Applications/Alarm-Management-System.aspx?navID=1877 (дата обращения: 14.11.2015).

[Covidien, 2015] Covidien Nellcor™ SatSeconds. <http://www.covidien.com/rms/clinical-solutions/alarm-management> (дата обращения: 14.11.2015).

[Матюшин и др., 2013] Матюшин М.М., Потоцкий С.И., Скобелев П.О., Потапов В.И., Лахин О.И. Автоматизированная система поддержки принятия решений в аварийных ситуациях // Программные продукты и системы. – 2013. №3.

[Вакурина и др., 2014] Вакурина Т.Г., Матюшин М.М. и др. Интеллектуальная система предупреждения нештатных ситуаций в бортовых системах ТК «Прогресс» // XII Всероссийское совещание по проблемам управления ВСПУ-2014, Москва, 16 – 19 июня 2014 г.: Труды. [Электронный ресурс]. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2014.

FEATURES OF REALIZATION OF INTELLIGENT SYSTEM OF ANALYSIS AND FORECASTING OF SPACE VEHICLES

Iurygina I. S., Lakhin O. I., Donskov A.V.,
Mishurova N. V., Sysoev D.V.,
Skoryupina E.G.

Software Engineering Company «Smart Solutions», Samara, Russian Federation

yurygina@smartsolutions-123.ru

OJSC «S. P. Korolev Rocket and Space Corporation «Energia», Korolev, Russian Federation

post@rsce.ru

The purpose of the study is designing an intelligent system for supporting safety and reliability at the operation stage of a manned space vehicle.

Introduction

Modern space flight control requires quick decision-making based on data analysis, reflecting the current condition of the vehicle. Execution of this laborious task largely depends on the proficiency of the specialist. Existing solutions for managing a complex technical object are limited by their knowledge domain, which prevents them from being used in related fields. The primary objective of the system development is to assist Flight Control Team specialists in monitoring, analyzing and decision-making for quick prevention and elimination of emergencies.

Main Part

A basis for the suggested approach is using multi-agent technologies and semantic networks (ontologies) as well as a conceptual model of a complex technical object's conceptual model and forming a cyber-physical model. Results:

There have been developed a universal solution for designing an intelligent system supporting safety and reliability a manned space vehicle: units of the complex technical object described in the ontology, each subsystem of the object has a correspondent software agent. Agents analyze the system's state based on the model of normal states and emergencies, signal in case of detection for an emergency occurrence.

Recommendations for an emergency elimination are formed using the emergency elimination rules and the model of physical processes described in the ontology.

Conclusion

Application of the suggested approach will help the Flight Control Team specialists to locate deviations from the normal conditions in time and take appropriate measures. It is expected that the economic benefit will be obtained by reducing risks of failure and improving the object reliability parameters.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

ОСОБЕННОСТИ ПОСТРОЕНИЯ ОНТОЛОГИИ ДЛЯ АДАПТИВНОГО ПЛАНИРОВАНИЯ ПРОГРАММЫ ПОЛЕТА И ГРУЗОПОТОКА РОССИЙСКОГО СЕГМЕНТА МЕЖДУНАРОДНОЙ КОСМИЧЕСКОЙ СТАНЦИИ

Лахин О.И., Полников А.С., Редькина К.В.

ООО «НПК «Разумные решения», г. Самара, Россия

lakhin@smartsolutions-123.ru

polnikov@smartsolutions-123.ru

redkina@smartsolutions-123.ru

В работе рассматривается задача адаптивного планирования программы полёта и грузопотока Российского сегмента Международной космической станции (РС МКС) с применением интерактивной мультиагентной системы построения программы полета, грузопотока и расчета ресурсов РС МКС. Описывается подход к формированию онтологии МКС, используемой агентами при переговорах в процессе планирования. Приводятся основные концепты, атрибуты и отношения, задействованные в процессе планирования, и примеры их влияния на конечный результат.

Ключевые слова: российский сегмент международной космической станции; программа полета, поблочный грузопоток; мультиагентные технологии; адаптивное планирование; онтология.

Введение

Для обеспечения жизнедеятельности экипажа Международной космической станции (МКС) требуется сформировать и реализовать множество взаимосвязанных планов по различным аспектам её функционирования. Построение плана полетов, а также доставки, возврата, утилизации и размещения на МКС грузов (свыше 7000 тыс. наименований) — сложная и трудоемкая задача, при решении которой необходимо учитывать множество противоречивых факторов, ограничений и предпочтений, таких, например, как изменяющиеся потребности в топливе, воде и продовольствии, баллистика и солнечная активность, типы кораблей и стыковочных модулей и т.д. [Бидеев, 2014].

Для решения данной задачи используется интерактивная мультиагентная система построения программы полета, грузопотока и расчета ресурсов РС МКС (МАС «Программа полета и грузопоток»), внедренная в промышленную эксплуатацию в ОАО «РКК «Энергия» в декабре 2012 г. Для описания основных компонентов МКС, а также правил, критериев и ограничений планирования, в системе используются взаимосвязанные онтологии, применяемые при автоматическом планировании. Онтологии могут быть представлены в виде древовидного справочника, семантической сети или карточек концептов.

1. Краткое описание системы

Система состоит из набора программных модулей по каждой из областей обслуживания МКС:

- Модуль «**Программа полета**» представляет план стартов, стыковок и расстыковок транспортных средств (ТС) к стыковочным узлам модулей РС МКС.
- Модуль «**Время экипажа**» предназначен для расчёта затрат времени по категориям операций экипажа на МКС для каждой экспедиции.
- Модуль «**Поблочный грузопоток**» представляет собой детальную информацию для каждого полета о доставляемых на МКС грузах.
- Модуль «**Размещение**» предназначен для автоматического планирования размещения грузов, доставляемых на транспортных кораблях в зоны хранения модулей РС МКС.
- Модуль «**Возврат**» предоставляет детальную информацию о возвращаемых с МКС грузах для каждого полета.
- Модуль «**Утилизация**» предназначен для формирования детальных перечней грузов для последующей утилизации.
- Модуль «**Расчёт ресурсов**» предназначен для прогнозирования расходов топлива, воды, пищи и прочих расходуемых ресурсов на станции в зависимости от полетных операций, состава экипажа, а также доставок грузов на РС МКС.

В основе используемого в планировании мультиагентного подхода лежит создание программных агентов, которые могут действовать от лица и по поручению каждого ТС, договариваясь с другими агентами о времени стыковки к портам станции или сроках доставки грузов. Особенностью данного подхода является возможность построения программы полета и плана грузопотока в результате функционирования динамически формируемой сети потребностей и возможностей (ПВ-сети) ресурсов МКС. Конфигурация агентов в такой сети может трансформироваться в любой момент вследствие изменения текущей сцены в результате внешних или внутренних событий, информация о которых поступает из центра управления полетами (ЦУП) и кураторов [Ивашенко, 2010].

2. Описание онтологии МКС

Онтология МКС формирует базис для описания предметной области планирования и содержит основные концепты, их атрибуты и отношения между ними. Онтология МКС содержит только классы концептов, экземпляры объектов ссылаются на родительские концепты и наследуют от них все имеющиеся атрибуты и отношения [Вакурина, 2014]. Выделены следующие основные классы концептов:

- *Экспедиция* – концепт, определяющий условия начала новой экспедиции на МКС. Данный концепт связан отношениями «Зависит» с типами ТС, способными к транспортировке космонавтов, такими как «Союз» и «Шаттл».
- *Экипаж* – концепт, связанный отношениями «Состоит из» с концептами класса «Космонавт».
- *Космонавт* – класс концептов для описания космонавтов и астронавтов, имеющих отношения с концептами «Страна» и «Организация».
- *Операции космонавтов РС* – концепты категорий операций космонавтов, учитывающихся при расчёте рабочего времени экипажа, включая операции при стыковке ТС, внекорабельной деятельности (ВнеКД) и др.
- *Тип ВнеКД* – концепты различных типов ВнеКД, в зависимости от точки выхода космонавтов на внешнюю поверхность МКС.
- *Этап сборки МКС* – концепты различных этапов сборки, на каждом из которых МКС имела различную конфигурацию.
- *Полёт* – концепты для различных типов полётов ТС: полётов со стыковкой к МКС, либо автономных полётов без стыковки.
- *Операции* – категории полётных операций РС МКС, включает такие концепты как «Стыковка», «Отстыковка», «ВнеКД», «Коррекция орбиты» и др. Каждый концепт обладает различным набором отношений, так, например, концепт «Перестыковка» имеет отношения «От» и «К» с концептом «Порт МКС», а концепт «ВнеКД» связан отношениями «Требует» и «Зависит» с концептом «Космонавт РФ».

- *Сегмент МКС* – концепты «Российский сегмент» и «Американский сегмент» описывают разграничение сфер ответственности между обслуживающими МКС организациями. Сегменты связаны отношениями «Состоит из» с концептами модулей МКС.

- *Модуль* – класс, описывающий концепты различных модулей МКС, таких как служебные модули и функционально-грузовые блоки. Модули связаны отношениями «Имеет» с концептами класса «Порт МКС».

- *Порт МКС* – концепты стыковочных портов модулей МКС. Порты обладают атрибутами «Дата стыковки» и «Порядок», которые характеризуют даты и этапы сборки конфигурации МКС.

- *Бак* – отношение с данным концептом характеризует наличие у ТС баков для транспортировки жидких грузов. Концепт обладает атрибутами «Вместимость» и «Использовать для утилизации» (значение данного атрибута определяет, могут ли при планировании утилизации на указанном ТС утилизироваться жидкие отходы).

- *Отсек* – концепты, описывающие отсеки для сухих грузов различных типов ТС. Аналогично бакам для жидких грузов, концепты отсеков обладают атрибутами «Вместимость» и «Использовать для утилизации».

- *Двигатель* – класс концептов двигателей МКС и пристыкованных к ней ТС, участвующих в операциях стыковки/отстыковки и коррекции орбиты.

- *Транспортное средство* – основной класс концептов, отношения и значения атрибутов которых определяют условия планирования.

Рассмотрим подробнее атрибуты и отношения концептов класса «Транспортное средство», экземпляры которых (отдельные космические корабли) играют основную роль при формировании планов во всех модулях и системах. Отношения «Баки» и «Отсеки» связывают экземпляр ТС с экземплярами отсеков и баков, характеризуя его возможности по доставке, спуску и утилизации сухих и жидких грузов. Количество космонавтов на борту ТС указывается атрибутом «Количество человек для перевозки».

О возможности использования типов ТС для различных операций свидетельствуют атрибуты вида «Использовать для построения плана доставки», «Использовать для утилизации» и т.п. Максимальная и допустимая вместимость топлива и полезной нагрузки характеризуется атрибутами «Полезный груз», «Максимальный объём топлива в рамках полезного груза», «Минимальный объём топлива» и «Минимальный запас топлива для выполнения манёвров». Атрибут «Очередность использования бака транспортного средства» указывает, в каком порядке выполняется расход воды и топлива и баков при наличии нескольких пристыкованных ТС.

3. Использование онтологии при планировании программы полёта

В процессе автоматизированного создания программы полета агент порта предоставляет агентам стыковок информацию о свободных интервалах на портах или информацию о том, кто занял интересующий их отрезок. Концепты типов ТС имеют отношения «Лучше стыковать»/«Хуже стыковать» со стыковочными портами, определяющие возможность и приоритеты стыковок к различным портам МКС. Агент полета предоставляет агентам стыковок время, которое они могут занять. При этом учитываются атрибуты минимальных интервалов времени между стартами, стыковками и отстыковками ТС (как на одном, так и на нескольких стыковочных портах). Агент полета создает и контролирует агенты своих стыковок и устанавливает для них правила планирования на основании значений атрибутов – расстояний между полётами ТС (для стартов, стыковок и отстыковок). Также учитываются атрибуты «Количество дней полёта до стыковки» для всех типов ТС и «Допустимые месяцы для старта» для Транспортно-Пассажирского Корабля (ТПК) «Союз».

Использование онтологии позволяет пользователям системы самостоятельно указывать и модифицировать приоритеты планирования, без необходимости доработки и модификации системы, путём изменения значений атрибутов и отношений [Кузьмин, 2014]. Так, в большинстве случаев ТПК «Союз» стыкуются к узлам модулей МИМ1 и МИМ2, а Транспортно-Грузовые Корабли (ТГК) «Прогресс» – к портам модулей СО1 и СМ, однако при отсутствии места на предпочитаемых портах допускается использовать менее приоритетные. В онтологии концепт «Союз» имеет отношения «Лучше стыковать» с концептами стыковочных портов МИМ1 и МИМ2 и «Хуже стыковать» с портами СМ и СО1. Ситуация для концепта «Прогресс» обратная, т.е. имеются отношения «Лучше стыковать» с портами СМ и СО1, и «Хуже стыковать» с портами МИМ1 и МИМ2.

Рассмотрим пример появления нового транспортного средства «SpX». Поскольку SpX является подклассом концепта «Транспортное средство», то он наследует все его значения атрибутов с условиями планирования, такими как «Минимальный интервал между стартами» и «Минимальный интервал между стыковками на одном порту». С помощью создания нового отношения «Лучше стыковать» с концептом «Американский порт (Nadir)» для нового типа ТС задаются условия планирования, после чего экземпляры данного концепта (отдельные космические корабли) будут планироваться в соответствии с указанными пользователем требованиями.

4. Описание онтологии грузопотока

Задачу планирования доставок на борт МКС грузов можно свести к ПВ-сети, где возможностями является набор полётов к РС МКС с фиксированными датами, а потребностями – грузы с набором различных параметров и предпочтений. При формировании ПВ-сети посредством онтологии полёты представляются в виде экземпляров класса «Транспортное средство», имеющих свойство «Дата старта», а грузы описываются онтологией грузопотока МКС. Кроме основных атрибутов, таких как масса и габариты, интерес при планировании представляют следующие атрибуты и отношения концепта единицы груза:

- *Ресурс* – атрибут с численным значением ресурса работы груза. Планировщик будет стремиться доставить такой груз на ближайшем полёте перед истечением срока годности уже имеющейся на борту МКС единиц.
- *Доставки для ВнеКД* – атрибут грузов, расходуемых при проведении ВнеКД. Планировщик будет планировать такие грузы на ближайший полёт перед осуществлением очередного ВнеКД.
- *Минимальный остаток* – атрибут количества дней, в течение которых космонавты должны быть обеспечены грузом в случае пропущенного цикла доставки. При планировании доставок расходуемых грузов планировщик будет учитывать, что величина текущего запаса на борту МКС ни в какой момент не должна оказаться меньше минимального остатка.
- *Норма расхода, кг на чел. в сутки* – атрибут, значение которого указывается только для расходуемых грузов. Планировщик учитывает нормы расхода при определении доставок для поддержания минимального остатка.
- *Приоритет* – численный атрибут, показывающей приоритетность груза при планировании; по умолчанию равен приоритету родительской системы.
- *Регулярные доставки на ТГК* – атрибут, показывающий минимальное обязательное количество единиц груза, доставляемых на грузовом корабле «Прогресс».
- *Регулярные доставки на ТПК* – атрибут, показывающий минимальное обязательное количество единиц груза, доставляемых на транспортном корабле «Союз».
- *Резервный* – атрибут для грузов, используемых при нештатных ситуациях. Такие грузы учитываются планировщиком в зависимости от уже имеющегося их запаса на борту МКС
- *Предпочитаемый тип ТС* – отношение с концептом класса «Транспортное средство», на котором предпочтительна доставка груза. При отсутствии такого отношения планировщик не будет делать различий между ТС различных типов при планировании груза. Если же такое отношение имеется, то планировщик будет стремиться сначала запланировать груз на экземпляре ТС с наиболее выгодной датой старта, затем на других ТС того же

типа, и только в последнюю очередь на ТС прочих типов.

- Отношения «Груз-укладка» и «Груз-вложение» между двумя или более грузами свидетельствуют о наличии между ними пространственной связи.

Грузом-укладкой считается груз, который имеет массу и сложную внутреннюю структуру, при планировании грузопотока учитывается масса самой укладки, а не грузов в её составе.

Грузом-вложением считается груз, который входит в состав груза-укладки, при этом масса груза-вложения не учитывается при планировании грузопотока. В составе одного груза-укладки может находиться несколько разных грузов-вложений.

5. Использование онтологии при планировании грузопотока

Начальными условиями для формирования грузопотока является планируемый период с набором полётов ТПК «Союз» и ТГК «Прогресс». При формировании грузопотока, значения атрибутов «Регулярные доставки на ТГК» и «Регулярные доставки на ТПК» определяют число единиц груза, в обязательном порядке планирующиеся на ТС каждого типа. Для прочих грузов адаптивное планирование выполняется посредством взаимодействий агентов грузов и улучшений ими своей текущей позиции в соответствии с принципами гомеостатической оптимизации. [Бидеев, 2014]

Рассмотрим появление нового типа ТС – концепта «Прогресс МС/Союз-2.1а», у которого отсутствуют отношения «Предпочитаемый тип ТС» с грузами онтологии грузопотока. По этой причине планировщик первоначально будет стремиться игнорировать ТС этого типа, планируя на них доставки в самую последнюю очередь. Изменить такое поведение посредством расширения онтологии возможно за счёт переноса в онтологию МКС данного концепта с верхнего уровня иерархии типов ТС на уровень дочернего элемента для уже существующего ТГК «Прогресс».

В этом случае для грузов с отношениями «Предпочитаемый тип ТС» с концептом «Прогресс» будут распространяться те же правила планирования на новый концепт «Прогресс МС/Союз-2.1а». Для тех грузов, условиям доставки которых удовлетворяет только новая модификация ТГК, следует создать отношения «Предпочитаемый тип ТС» с элементов «Прогресс МС/Союз-2.1а» напрямую, минуя его родительский концепт. Такие грузы планировщик будет стремиться распределить только на ТС соответствующего типа, не допуская их перераспределения на соседние полёты.

Заключение

Использование онтологий для адаптивного построения программы полета и грузопотока на РС

МКС позволило обеспечить гибкость системы планирования, а также расширить базу знаний без необходимости перепрограммирования системы. Полученные от специалистов ОАО «РКК «Энергия» оценки позволяют заключить, что система положительно зарекомендовала себя не только при штатной работе, но и при необходимости срочного согласования и корректировки планов по причине возникших нештатных и аварийных ситуаций.

Библиографический список

[Бидеев, 2014] Бидеев А.Г., Метод адаптивного планирования грузопотока в интерактивной мультиагентной системе расчета программы полета, грузопотока и ресурсов российского сегмента Международной космической станции / Бидеев А.Г. [и др.] // Космическая техника и технологии. – 2014. №1(4). – С. 29-38.

[Иващенко, 2010] Иващенко, А.В. Мультиагентная подсистема построения программы полета Международной космической станции / А.В. Иващенко [и др.] // Перспективные информационные технологии для авиации и космоса (ПИТ-2010). Труды Международной конференции с элементами научной школы для молодежи, Самара, 29 сентября – 1 октября 2010 г. Самара: СГАУ, 2010. – С.98-101.

[Вакурина, 2014] Вакурина, Т.Г. Онтология российского сегмента Международной космической станции и ее практическое использование в интеллектуальных аэрокосмических приложениях / В.В. Котеля, О.И. Лакхин, М.М. Матюшин, П.О. Скобелев // Материалы IV Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» Open Semantic Technologies for Intelligent Systems (OSTIS-2014). - Минск: БГУИР, 20 – 22 февраля 2014 г. – С. 221-226.

[Кузьмин, 2014] Кузьмин В.В., Результаты внедрения интерактивной мультиагентной системы построения программы полета, грузопотока и расчета ресурсов МКС / В.В. Кузьмин [и др.] // Материалы XX Научно-технической конференции молодых ученых и специалистов, ОАО «Ракетно-космическая корпорация «Энергия» имени С.П. Королёва», 10 – 14 ноября 2014 года. – С. 607-609.

DEVELOPING ONTOLOGIES FOR USE IN ADAPTING SCHEDULING OF FLIGHT PROGRAM FOR THE INTERNATIONAL SPACE STATION

Lakhin O., Polnikov A., Redkina K.

Software Engineering Company Smart Solutions,
Samara, Russian Federation

lakhin@smartsolutions-123.ru

polnikov@smartsolutions-123.ru

redkina@smartsolutions-123.ru

The task of adaptive scheduling of flight program and cargo flow of the Russian Orbital Segment of the International Space Station is solved using the Multi-Agent System for Scheduling of Flight Program, Cargo Flow and Resources of International Space Station. This paper describes the approach to building the ontology used by agents during the flight program planning stage and describes major concepts, attributes and relations involved. Included is an example of how they affect the result of the planning process.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891

НЕКОТОРЫЕ ОСОБЕННОСТИ РАЗРАБОТКИ И ПРИМЕНЕНИЯ ОБУЧАЮЩИХ ИНТЕГРИРОВАННЫХ ЭКСПЕРТНЫХ СИСТЕМ В УЧЕБНОМ ПРОЦЕССЕ НИЯУ МИФИ

Рыбина Г.В., Рыбин В.М., Сергиенко Е.С.

*Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация*

galina@ailab.mephi.ru

Анализируется опыт разработки и использования в учебном процессе НИЯУ МИФИ обучающих интегрированных экспертных систем, созданных на основе задачно-ориентированной методологии и интеллектуальной программной среды комплекса АТ-ТЕХНОЛОГИЯ. Делается акцент на особенностях реализации некоторых задач интеллектуального обучения, связанных с выявлением знаний и умений обучаемых решать неформализованные задачи.

Ключевые слова: обучающие интегрированные экспертные системы, интеллектуальное обучение, повторно-используемые компоненты, инструментальный комплекс АТ-ТЕХНОЛОГИЯ.

Введение

В обширном спектре проблематики интеллектуальных обучающих систем (ИОС), рост интереса к которым возник на рубеже XX и XXI веков, к настоящему времени предложено значительное число подходов и системных архитектурных решений для интеллектуализации, индивидуализации и веб-ориентации процессов обучения и подготовки специалистов. Не претендуя на полноту обзора публикаций в области ИОС, укажем только некоторые современные отечественные [Рыбина Г.В., 2011], [Рыбина Г.В., 2014а], [Юрков Н.К., 2010], [Тельнов Ю.Ф., 2014], [Тельнов Ю.Ф., 2015], [Трембач В.М., 2013], [Швецов А.Н. и др., 2014], [Смирнова Н.В. и др., 2012] и зарубежные [Nye B.D., 2015], [Bonner D. и др., 2015], [Allu M., 2008], [Keleş A. и др., 2010] работы, отражающие широкую типологию архитектур ИОС.

В лаборатории «Интеллектуальные системы и технологии» кафедры «Кибернетика» НИЯУ МИФИ за последние годы накоплен опыт разработки и использования в учебном процессе целого ряда обучающих интегрированных экспертных систем (ИЭС), созданных на основе задачно-ориентированной методологии и мощного современного инструментария – комплекса АТ-ТЕХНОЛОГИЯ [Рыбина Г.В., 2008]. Обучающие ИЭС и веб-ИЭС являются полнофункциональными ИОС нового поколения, обеспечивающими

реализацию всех базовых моделей ИОС (модель обучаемого, модель обучения, модель проблемной области, онтологии курсов/дисциплин и др.), а также решение комплекса задач интеллектуального обучения, основными из которых являются [Рыбина Г.В., 2014а]: индивидуальное планирование методики изучения учебного курса/дисциплины; интеллектуальный анализ решения учебных задач; интеллектуальная поддержка принятия решений.

Как показано в [Рыбина Г.В., 2011], [Рыбина Г.В., 2014а], [Трембач В.М., 2013] и др. работах, объединяющей основой задач интеллектуального обучения путем использования различных архитектур ИОС являются процессы выявления *знаний* (декларативные знания по конкретному курсу/дисциплине) и *умений* (процедурные знания, позволяющие демонстрировать, как на практике применяются полученные декларативные знания). Например, для реализации этих процессов в обучающих ИЭС, сетевая компетентностно-ориентированная модель обучаемого [Рыбина Г.В., 2008] динамически формируется на основе анализа ответов на вопросы из специальных веб-тестов, генерация которых осуществляется с помощью генетических алгоритмов, а методика оценивания знаний базируется на вычислении результирующей оценки за полный тест. Затем текущая модель обучаемого сравнивается с онтологией курса/дисциплины, в результате чего выявляются так называемые «проблемные зоны» в знаниях обучаемого по отдельным разделам и темам курса/дисциплины. Есть и другие подходы к

выявлению уровня знаний обучаемого, описанные, в частности, в работах [Юрков Н.К., 2010], [Трембач В.М., 2013], [Смирнова Н.В. и др., 2012], [Durlach P.J. и др., 2012], [Conati C., 2012], [Feng M. и др., 2010], однако, с методической, алгоритмической и технологической точек зрения реализация данных процессов не представляет особой сложности.

Что касается методов компьютерного выявления умений обучаемых решать учебные задачи, то здесь возникают проблемы, связанные со спецификой конкретного курса/дисциплины. Как показано в [Рыбина Г.В. и др., 2015а], в образовательном процессе выделяются два основных класса курсов/дисциплин – сильно и средне формализованные (математика, физика, теоретическая механика и др.) и слабо формализованные (неформализованные) гуманитарные, инженерные и специальные курсы/дисциплины. Для первого из них вопросы создания компьютерных средств, отражающих методику преподавания, специфику требований к учебным задачам, способы оценивания и др., достаточно хорошо проработаны и опираются на уже готовые решения и стандарты.

Относительно второй группы курсов/дисциплин, ситуация складывается намного сложнее, поскольку для подобных неформализованных курсов/дисциплин процесс выявления умений решать учебные задачи связан с моделированием рассуждений человека (обучаемого), и уже требуются совершенно другие подходы, которые связаны с искусственным интеллектом, в частности, с методами и средствами традиционных экспертных систем (ЭС) и ИЭС.

Например, преподавание специальных курсов/дисциплин по направлениям подготовки «Прикладная математика и информатика» и «Программная инженерия» («Введение в интеллектуальные системы», «Экспертные системы», «Интеллектуальные информационные системы», «Интеллектуальные диалоговые системы» и др.) связано с привитием навыков и умений обучаемых решать такие задачи, как [Рыбина Г.В., 2011], [Рыбина Г.В., 2014а], [Рыбина Г.В. и др., 2015а], [Рыбина Г.В., 2014б]: умение строить по принципу «сам себе эксперт» модели простейших ситуаций проблемной области (Про) на основе фреймов и семантических сетей, моделирование стратегий прямого/обратного вывода в ЭС, построение компонентов лингвистической модели подязыка деловой прозы и другие. Перечисленные выше учебные задачи базируются на неформализованных экспертных методиках (НФ-методиках), опыт реализации которых накоплен в технологиях традиционных ЭС и ИЭС, в частности, в инженерии знаний.

С другой стороны, преподавание специальных курсов/дисциплин по направлению подготовки «Электроника и автоматика физических установок» («Автоматизация физических установок и научных исследований») связано с привитием навыков и

умений сочетать строгие формализованные методы с НФ-методиками, например, в тех случаях, когда [Рыбин В.М., 2011]: невозможно построить точную математическую модель объекта управления; возникают дополнительные источники внешних возмущений (например, эксплуатационные параметры в системах управления электрофизическими комплексами); требования пользователей оказываются противоречивыми, неполными, приближенными и т.д. Поэтому, для поддержки построения обучающих ИЭС на основе задачно-ориентированной методологии (комплекс АТ-ТЕХНОЛОГИЯ) были созданы и апробированы на практике в учебном процессе НИЯУ МИФИ и других вузов специальные средства, реализующие «ручные» методики решения различных НФ-задач, в частности, представленных в [Рыбина Г.В., 2014а], [Рыбина Г.В., 2014б].

Цель данной работы – описать полученный опыт реализации некоторых задач интеллектуального обучения именно в контексте НФ-методик, использующихся в обучающих ИЭС и веб-ИЭС по конкретным курсам/дисциплинам.

1. Общая характеристика компонентов интеллектуальной программной среды комплекса АТ-ТЕХНОЛОГИЯ

Комплекс АТ-ТЕХНОЛОГИЯ – это современный инструментарий типа WorkBench, поддерживающий интеллектуальную программную технологию автоматизированного построения ИЭС различной типологии и уровня сложности. Концептуальной основой для интеграции методов инженерии знаний, онтологического инжиниринга, интеллектуального планирования и традиционного программирования является понятие «интеллектуальной программной среды», впервые введенной в работе [Рыбина Г.В., 2008] и экспериментально исследованное в процессе разработки целого ряда прикладных ИЭС, в том числе обучающих ИЭС [Рыбина Г.В., 2008], [Рыбина Г.В. и др., 2013], [Рыбина Г.В., 2014а], [Рыбина Г.В. и др., 2015б]. Базовая роль в интеллектуальной программной среде принадлежит *интеллектуальному планировщику*, управляющему проектами по разработке ИЭС и веб-ИЭС. Различные версии планировщика детально описаны в [Рыбина Г.В., 2014а], [Рыбина Г.В., 2008], [Рыбина Г.В. и др., 2013], [Рыбина Г.В. и др., 2015б] и др. работах. Поэтому в фокусе внимания данной работы находятся вопросы, связанные с методами реализации рассмотренных выше задач интеллектуального обучения с помощью других, не менее важных компонентов интеллектуальной программной среды комплекса АТ-ТЕХНОЛОГИЯ.

Как показано в [Рыбина Г.В., 2008], основное место в составе компонентов интеллектуальной программной среды, использующихся для построения и исполнения планов разработки прототипов прикладных ИЭС, отводится *типовым*

проектным процедурам (ТПП) и повторно-используемым компонентам (ПИК). В частности, основным значимым алгоритмическим элементом, используемым для построения обучающих ИЭС и веб-ИЭС, является ТПП «Построение обучающих ИЭС», под которой в целом понимается набор типовых действий (инструкций), традиционно совершаемых инженером по знаниям на каждой стадии жизненного цикла разработки ИЭС при решении каких-либо проектных задач.

Сложность данной ТПП связана с необходимостью поддержки двух режимов функционирования обучающих ИЭС – режима Design Time, ориентированного на работу преподавателей и/или инженеров по знаниям (построение онтологии курсов/дисциплин, построение обучающих воздействий различных типов и т.д.) и режима RunTime, предназначенного для работы с обучаемым (построение текущих компетентностно-ориентированных моделей обучаемых, включая психологические портреты, динамическое формирование индивидуальных планов обучения на основе использования различных обучающих воздействий и т.д.).

Относительно роли ПИК следует отметить, что поскольку архитектура инструментального комплекса АТ-ТЕХНОЛОГИЯ построена по принципу распределенной функциональности, которая «разносится» по компонентам, зарегистрированным в комплексе и действующим под управлением интеллектуальной программной среды, то данные компоненты являются ПИК комплекса и реализуются по правилам, определенным для ПИК [Рыбина Г.В., 2008]. Поэтому при прототипировании обучающих ИЭС в настоящее время используется значительное число *операционных* и *информационных* ПИК, отражающих базовые процессы, связанные с решением задач интеллектуального обучения. Именно в контексте ПИК ниже представлено описание отдельных аспектов реализации интеллектуальной технологии обучения на основе применения в учебном процессе НИЯУ МИФИ обучающих ИЭС и веб-ИЭС. Данные вопросы обсуждались также в [Рыбина Г.В., 2008], [Рыбина Г.В. и др., 2013], [Рыбина Г.В., 2014а], [Рыбина Г.В. и др., 2015а].

2. Особенности реализации некоторых задач интеллектуального обучения на основе использования операционных и информационных ПИК

В соответствии с задачей-ориентированной методологией построения ИЭС [Рыбина Г.В., 2008], одним из важных компонентов обобщенной модели *типовой задачи обучения* является сетевая модель обучаемого, построение и обновление которой осуществляется динамически с помощью проведения контрольных мероприятий,

предусмотренных учебным планом каждого курса/дисциплины.

Для этих целей в составе подсистемы поддержки построения обучающих ИЭС/веб-ИЭС, функционирующей в базовой и веб-версии инструментального комплекса АТ-ТЕХНОЛОГИЯ, предназначены, в том числе, специальные средства построения модели обучаемого, представляющие из себя совокупность программных компонентов, использующихся для выявления как теоретических знаний, так и практических умений обучаемых решать учебные задачи по конкретным курсам/дисциплинам. Используя [Рыбина Г.В., 2014а], [Рыбина Г.В. и др., 2015а], рассмотрим более детально примеры реализации некоторых задач интеллектуального обучения.

2.1. Индивидуальное планирование методики изучения учебного курса

Основным операционным ПИК для данной задачи являются средства построения онтологии курса/дисциплины. В настоящее время компонентами интеллектуальной программной среды комплекса АТ-ТЕХНОЛОГИЯ поддерживаются *обобщенная* онтология «Интеллектуальные системы и технологии» (кафедра «Кибернетика») и онтология «Автоматизация физических установок и научных исследований» (кафедра «Автоматика»). Кроме того используется порядка десяти информационных ПИК, связанных с фрагментами гипертекстовых электронных учебников (ГТ-учебники) по конкретным курсам/дисциплинам, и несколько информационных ПИК для построения обобщенной онтологии «Интеллектуальные системы и технологии».

В целом, для построения текущих моделей обучаемого предназначены несколько ПИК [Рыбина Г.В. и др., 2015а], перечень которых приводится ниже.

1. Операционный ПИК «Компонент выявления уровня знаний» и несколько информационных ПИК, описывающих тестовые задачи для различных фрагментов онтологии конкретного курса/дисциплины.

2. Для обобщенной онтологии «Интеллектуальные системы и технологии» используются четыре операционных ПИК, связанных с выявлением умений обучаемых решать задачи на основе НФ-методик («Компонент выявления умений обучаемых моделировать стратегии прямого/обратного вывода», «Компонент выявления умений обучаемых строить компоненты лингвистической модели подязыка деловой прозы», «Компонент выявления умений обучаемых моделировать простейшие ситуации ПрО с помощью семантических сетей», «Компонент выявления умений обучаемых моделировать простейшие ситуации ПрО с помощью фреймов»).

3. Для онтологии «Автоматизация физических установок и научных исследований» используется

операционный ПИК, связанный с выявлением умений обучаемых разрабатывать системы автоматического управления (САУ) физическими установками («Компонент САУ физическими установками»).

4. Два операционных ПИК - «Генератор психологических тестов» и «Компонент выявления личностных характеристик обучаемых», причем процесс генерации психологических тестов осуществляется с использованием информационных ПИК, содержащих фрагменты авторских психологических тестов, направленных на выявление совокупности личностных характеристик обучаемых.

Следует отметить, что компонент отображения текущей модели обучаемого на онтологию курса/дисциплины, оформленный в виде ПИК, позволяет выявить «проблемные зоны» обучаемого, что лежит в основе построения индивидуального плана (стратегии) обучения. В контексте интеллектуальной технологии построения ИЭС, для автоматической генерации индивидуального плана обучения используется операционный ПИК «Компонент формирования планов (стратегий) обучения», а исполнение плана осуществляется специальными ПИК «Компонент управления реализацией обучающих воздействий».

Соответственно, каждая стратегия обучения включает определенную последовательность обучающих воздействий, в качестве которых в настоящее время используются следующие [Рыбина Г.В., 2008], [Рыбина Г.В., 2014a]: чтение ГТ-учебника; решение нескольких типов учебно-тренировочных задач; выполнение обучающего воздействия «Тренинг с ИЭС»; объяснения полученных результатов; подсказки; локализация ошибочных действий; контроль правильности решения и др.

Любая стратегия обучения характеризуется конкретным набором и порядком применения учебных воздействий, содержание которых определяется степенью конкретизации поставленной задачи, зависящей от уровня знаний/умений обучаемого и его психологического портрета. Процесс формирования и реализации всех обучающих воздействий поддерживается двумя специальными операционными ПИК (без учета ПИК, соответствующих базовым средствам комплекса АТ-ТЕХНОЛОГИЯ), а также несколькими информационными ПИК.

2.2. Интеллектуальный анализ решений учебных задач

В соответствии с [Рыбина Г.В., 2011], [Рыбина Г.В., 2014a], для выявления навыков/умений обучаемых решать учебные НФ-задачи из шести курсов/дисциплин, представленных в обобщенной онтологии «Интеллектуальные системы и технологии», используется моделирование процесса рассуждений студентов, решающих четыре типа следующих учебных задач: моделирование

стратегий прямого/обратного вывода в ИЭС; моделирование простейших ситуаций ПрО с использованием фреймов и семантических сетей; построение компонентов лингвистической модели простейшего подязыка деловой прозы. Используя [Рыбина Г.В. и др., 2015a], приведем краткую характеристику операционных ПИК, обеспечивающих реализацию вышеперечисленных задач.

1. ПИК «Компонент выявления умений обучаемых моделировать стратегии прямого/обратного вывода» и несколько информационных ПИК (фрагменты баз знаний) полностью обеспечивают данную функциональность в рамках изучения таких курсов, как «Введение в интеллектуальные системы», «Экспертные системы», «Интеллектуальные информационные системы» и др. Обучаемый проходит следующие этапы: создание фрагмента базы знаний, состоящей из продукционных правил; задание начальных фактов для осуществления прямого вывода; моделирование стратегии прямого вывода с получением оценки; задание начальных фактов и целей для осуществления обратного вывода; моделирование стратегии обратного вывода с получением оценки. Для оценивания умений обучаемого реализован простейший решатель, осуществляющий полный цикл эталонного вывода, а затем на основе «защитых» эвристик, производится сравнение действий обучаемого с эталонными этапами.

2. ПИК «Компонент выявления умений обучаемых моделировать простейшие ситуации ПрО с помощью фреймов» и несколько информационных ПИК (фрагменты фреймов-прототипов на упрощенном языке представления знаний FRL [Рыбина Г.В., 2014b]) обеспечивают объявленную функциональность в рамках курсов «Введение в интеллектуальные системы», «Экспертные системы», «Интеллектуальные информационные системы». Обучаемый выполняет контрольные задания по созданию фреймов-прототипов для ПрО, заданной преподавателем, после чего на основе использования простейшего FRL-интерпретатора производится оценка его умений, причем ведется полная история действий обучаемого, воспроизводящая логику рассуждений обучаемого и его ошибки.

3. Операционный ПИК «Компонент выявления умений обучаемых моделировать простейшие ситуации ПрО с помощью семантических сетей» и несколько информационных ПИК (фрагменты семантических сетей) обеспечивают объявленную функциональность в рамках курсов «Введение в интеллектуальные системы», «Экспертные системы» и «Интеллектуальные информационные системы». Обучаемый выполняет контрольные задания по построению фрагментов семантической сети для заданной ПрО, после чего на основе экспертной методики сравнения с эталонными фрагментами семантической сети производится

оценка уровня умений обучаемого решать данный тип задач.

4. ПИК «Компонент выявления умений обучаемых строить компоненты лингвистической модели подъязыка деловой прозы» и несколько информационных ПИК (словари, фрагменты текстов подъязыка деловой прозы, и т.д.) обеспечивают объявленную функциональность в рамках курса «Интеллектуальные диалоговые системы». Обучаемый выполняет контрольные задания по созданию лексического, синтаксического и семантического компонентов лингвистической модели для конкретного текста подъязыка деловой прозы, а затем с помощью специальной экспертной НФ-методики производится оценка уровня его умений;

Для выявления навыков/умений обучаемых решать как формализованные так и НФ-задачи, представленных в онтологии «Автоматизация физических установок и научных исследований» используется операционных ПИК «Разработка САУ физическими установками», который обеспечивает следующую функциональность в рамках соответствующего курса/дисциплины [Рыбин В.М., 2011]: разработка структурных схем САУ; расчет устойчивости САУ; выбор элементов САУ и др.

2.3. Интеллектуальная поддержка принятия решений

Важно отметить, что при разработке обучающих воздействий типа «Тренинг с ИЭС» для различных курсов/дисциплин, содержащих НФ-методики, наиболее актуальной является задача построения моделей Про (в том числе на основе знаний, содержащих отдельные виды НЕ-факторов знаний [Рыбина Г.В., 2008]), а также реализация режима консультации с ИЭС путем построения сценария диалога с обучаемым, в котором значительное внимание уделяется объяснениям, подсказкам и/или проверке следующего этапа решения задачи и т.д.

Здесь применяются несколько операционных ПИК из базовых средств комплекса АТ-ТЕХНОЛОГИЯ (подсистема общения, универсальный АТ-РЕШАТЕЛЬ, подсистема объяснений и др.), поскольку разработка обучающего воздействия данного типа представляет собой задачу создания полноценной ИЭС. Используются также информационные ПИК (фрагменты баз знаний из ранее созданных обучающих воздействий «Тренинг с ИЭС», фрагменты сценариев диалога пользователя в режиме консультации с ИЭС и др.) и операционный ПИК «Объяснительный компонент», осуществляющий помощь на каждом этапе решения учебных задач, в частности, подсказку следующего этапа, объяснения типа «как?» и «почему?», а также визуализацию трассы вывода.

3. Особенности мониторинга функционирования обучающих ИЭС и веб-ИЭС

Таким образом, важной особенностью разработки и использования обучающих ИЭС и веб-ИЭС является автоматизация практически всех процессов, которые возникают в ходе обучения и контроля знаний/умений обучаемых [Рыбина Г.В., 2008], [Рыбина Г.В., 2011], [Рыбина Г.В., 2014а], [Рыбина Г.В. и др., 2015а]. Вся информация об обучаемых, онтологиях курсов/дисциплин, результатах прохождения обучения, результатах контроля обучаемых, индивидуальных рекомендациях на основании полученных результатов обучения и т.д. находится в единой среде и в любое время доступна обучаемому и/или контролирующему процесс обучения, что обеспечивается за счет специальных средств мониторинга процесса функционирования обучающих ИЭС. Иными словами, удовлетворяются все требования образовательного мониторинга, представляющего собой систему сбора, хранения, анализа и представления информации о состоянии наблюдаемых объектов, явлений, процессов с целью их оценки, контроля или прогноза.

Соответственно, в рамках операционного ПИК «Мониторинг процессов функционирования обучающих ИЭС» проводится аналитико-статистическая обработка всех данных, полученных о моделях обучаемых, что осуществляется на основе специальных параметров статистической обработки информации, используемых в процессе функционирования обучающих ИЭС и веб-ИЭС [Рыбина Г.В., 2014а], [Рыбина Г.В. и др., 2015а] и характеризующих как обучаемого, так и контингент обучаемых (для анализа накопленных данных, с целью улучшения качества обучения и выполнения функций мониторинга).

В [Рыбина Г.В. и др., 2015а] приведены примеры базовых параметров, предназначенных для обработки информации по каждому обучаемому: анализ «проблемных зон» по конкретным курсам/дисциплинам и их кластеризация; эффективность применения индивидуального плана обучения (типология и последовательность обучающих воздействий): оценка влияния обучающих воздействий на повышение уровня знаний и определение наиболее эффективных обучающих воздействий; расчет корреляции между уровнем умений и уровнем знаний по соответствующим темам курса/дисциплины; прогноз оценки на экзамене по результатам успеваемости в семестре; учет психологического портрета обучаемого; степень достижения целевых компетенций по конкретным курсам/дисциплинам и др.

Предусмотрены также параметры, используемые для обработки информации по всему контингенту обучаемых (группа, поток и др.), а именно: совокупный анализ «проблемных зон» по

конкретным курсам/дисциплинам и их кластеризация; оценка и кластеризация индивидуальных планов обучения по конкретным курсам/дисциплинам; прогноз результатов экзаменационной сессии; анализ и кластеризация психотипов обучаемых и др.

Заключение

В настоящее время осуществляются экспериментальные программные исследования, реинжиниринг и дальнейшее развитие всех компонентов интеллектуальной технологии построения обучающих ИЭС. Кроме того проводятся работы по реализации НФ-методик решения учебных задач по другим курсам различных онтологий (в частности «Динамические интеллектуальные системы» и др.)

Работа выполнена при поддержке грантов РФФИ 12-01-00467 «Методы и инструментальные средства построения интеллектуальных обучающих систем на основе интегрированных экспертных систем» и 15-01-04696 «Разработка теоретических основ и инструментальных средств построения динамических интегрированных экспертных систем».

Библиографический список

- [Рыбина Г.В., 2011] Рыбина Г.В. Интеллектуальные обучающие системы на основе интегрированных экспертных систем: опыт разработки и использования // Информационно-измерительные и управляющие системы. – 2011. – №10. – С. 4-16.
- [Рыбина Г.В., 2014а] Рыбина Г.В. Интеллектуальные системы: от А до Я. Серия монографий в 3 кн.: Кн. 1: Системы основанные на знаниях. Интегрированные экспертные системы. – М.: Научтехлитиздат, 2014. – 224 с.
- [Юрков Н.К., 2010] Юрков Н.К. Интеллектуальные компьютерные обучающие системы: моногр./ Н.К. Юрков. – Пенза: Изд-во ПГУ, 2010. – 304 с.
- [Тельнов Ю.Ф., 2014] Тельнов Ю.Ф. Принципы и методы семантического структурирования информационно-образовательного пространства на основе реализации онтологического подхода // Экономика, статистика и информатика. Вестник УМО. – М.: МЭСИ. 2014. №1. С.187
- [Тельнов Ю.Ф., 2015] Тельнов Ю.Ф. Реализация процессов учебно-методического обеспечения в интегрированном информационно-образовательном пространстве на основе сервисной архитектуры// Экономика, статистика и информатика. Вестник УМО. – М.: МЭСИ. 2015. №1. С.198
- [Трембач В.М., 2013] Трембач В.М. Системы управления базами эволюционирующих знаний для решения задач непрерывного образования. – М.: МЭСИ, 2013. – 255 с.
- [Швецов А.Н. и др., 2014] Швецов А.Н., Сибирцев Е.В., Андрианов И.А. Компьютерные обучающие системы: мультиагентный подход // XII Всероссийское совещание по проблемам управления ВСПУ-2014. Москва, 16-19 июня 2014 г.: Труды. [Электронный ресурс] М.: Институт проблем управления им. В.А. Трапезникова РАН, 2014. 9616 с.
- [Смирнова Н.В. и др., 2012] Смирнова Н.В., Шварц А.Ю. Мотивационно-волевой компонент модели обучаемого в следящих интеллектуальных системах. Часть 1 // Искусственный интеллект и принятие решений. 2012. №1. С.65-80.
- [Nye B.D., 2015] Nye B.D. Intelligent tutoring systems by and for the developing world: A review of trends and approaches for educational technology in a global context // International Journal of Artificial Intelligence in Education. Volume 25, Issue 2, 2015, P. 177-203
- [Bonner D. и др., 2015] Bonner D., Walton J., Dorneich M.C., Gilbert S.B., Winer E., Sottolare R.A. The development of a testbed to

assess an intelligent tutoring system for teams // Workshops at the 17th International Conference on Artificial Intelligence in Education, AIED-WS 2015; Madrid; Spain; CEUR Workshop Proceedings. Volume 1432, 2015, P. 31-37

[Ally M., 2008] Ally M. Intelligent tutoring systems for distributed learning // Agent-Based Tutoring Systems by Cognitive and Affective Modeling. 2008, P. 292-306

[Keleş A. и др., 2010] Keleş A., Keleş A. Intelligent tutoring systems: Best practices // Intelligent Tutoring Systems in E-Learning Environments: Design, Implementation and Evaluation. 2010, P. 1-26

[Рыбина Г.В., 2008] Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. Монография. – М.: «Научтехлитиздат», 2008. – 482 с.

[Durlach P.J. и др., 2012] Durlach P.J., Lesgold A.M. Adaptive technologies for training and education. Cambridge University Press. 2012, 360 p.

[Conati C., 2012] Conati C. Student modeling and intelligent tutoring beyond coached problem solving // Adaptive Technologies for Training and Education. 2012, P. 96-116

[Feng M. и др., 2010] Feng M., Heffernan N., Koedinger K. Student modeling in an intelligent tutoring system // Intelligent Tutoring Systems in E-Learning Environments: Design, Implementation and Evaluation. 2010, P. 208-236

[Рыбина Г.В. и др., 2015а] Рыбина Г.В., Сергиенко Е.С. Интеллектуальное обучение на основе интегрированных экспертных систем: моделирование умений обучаемых решать сложные учебные задачи // Информационно-измерительные и управляющие системы. – 2015. – №1. – С.31-39.

[Рыбина Г.В., 2014б] Рыбина Г.В. Основы построения интеллектуальных систем. Учебное пособие. – М.: «Финансы и статистика», 2014. – 432с.

[Рыбин В.М., 2011] Рыбин В.М. Интеллектуальное управление на основе динамических интегрированных экспертных систем // Информационно-измерительные и управляющие системы. 2011, №6, с.16-19

[Рыбина Г.В. и др., 2013] Рыбина Г.В., Блохин Ю.М., Иващенко М.Г. Некоторые аспекты интеллектуальной технологии построения обучающих интегрированных экспертных систем // Приборы и системы. Управление, контроль, диагностика. 2013. №4. С.27-36

[Рыбина Г.В. и др., 2015б] Рыбина Г.В., Блохин Ю.М. Методы и средства интеллектуального планирования: применение для управления процессами построения интегрированных экспертных систем // Искусственный интеллект и принятие решений. 2015. №1. С.75-93

SOME FEATURES OF DEVELOPMENT AND USING OF TUTORING INTEGRATED EXPERT SYSTEMS IN EDUCATIONAL PROCESS MEPHI

Rybina G.V., Rybin V.M., Sergienko E.S.

*National Research Nuclear University MEPHI
(Moscow Engineering Physics Institute), Moscow,
Russia Federation*

galina@ailab.mephi.ru

Introduction

Analyzes the experience of development and use in the educational process MEPHI tutoring integrated expert systems created on the basis of task-oriented methodology and programming intelligent environment of AT-TECHNOLOGY workbench. The emphasis is on the peculiarities of the implementation of certain tasks of intellectual training, related to the identification of knowledge and skills of students to solve the problem unformalized.



МЕТОДЫ И МОДЕЛИ СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ ОБРАЗОВАТЕЛЬНОГО НАЗНАЧЕНИЯ

Шихнабиева Т.Ш.

*Институт управления образованием Российской академии образования,
г. Москва, Россия*

shetoma@mail.ru

Современные информационные технологии и стремительное расширение разнообразия сетевых образовательных услуг вызвало поток инноваций по реорганизации существующих образовательных систем всех уровней образования. В работе представлены некоторые подходы совершенствования существующих систем образовательного назначения с использованием интеллектуальных методов и моделей.

Ключевые слова: системы образовательного назначения, структуризация знаний, интеллектуальные обучающие системы, адаптивные семантические модели.

Введение

Следует отметить, что такие хорошо структурированные области, как математика, физика, теоретическая механика имеют в основе богатый математический аппарат для описания своих закономерностей, который позволяет вводить машинное моделирование с использованием традиционного алгоритмического программирования (без выделения уровня знаний). “Знания важны там, где определения размыты, понятия меняются, ситуации зависят от множества контекстов, где велика неопределенность, нечеткость информации” [Давыдов, 1972]. В качестве примера в нашей работе мы рассматриваем предметную область “Информатика”. Как мы наблюдаем, информатика является такой динамично развивающейся, постоянно пополняющейся новыми знаниями предметной областью. Неопределенность и нечеткость в терминологии, необходимость определения места новых знаний, в системе существующих требует разработки теоретических положений и определение методических основ структуризации знаний в системах обучения.

Статья посвящена теоретико-методологическим основаниям развития интеллектуальных обучающих систем на основе создания адаптивных семантических моделей (АСМ) [Шихнабиева, 2009] слабо структурированных областей знаний.

1. О влиянии современных информационных технологий и сетевых образовательных услуг на формы и методы обучения

Современные информационные технологии и стремительное расширение разнообразия сетевых

образовательных услуг вызвало поток инноваций по реорганизации существующих образовательных систем всех уровней образования - от школы до вуза. Как следствие, меняется характер и динамика взаимодействия обучающийся - преподаватель. Это существенным образом влияет на выбор методов, форм и технологий обучения. Обучение с использованием информационных ресурсов, хранящихся в Интернет и Интранет, является катализатором в становлении новой, прогрессивной теории преподавания и учения, ориентированной на развитие личности учащегося, способного реализовывать свои собственные, в том числе, и образовательные проекты и стремящегося к самоусовершенствованию на протяжении всей жизни.

Новые механизмы передачи информации оказали существенное влияние на средства, методы и формы обучения. Как следствие, возникает острая необходимость в анализе особенностей применения закономерностей общей теории обучения - дидактики, в условиях технологии компьютерного обучения и распределенных в пространстве и времени процессов сетевого обучения. Функции преподавателя и обучающихся в образовательной информационной среде претерпевают коренные изменения по сравнению с традиционной учебной средой. В учебных средах с использованием ИКТ учащийся работает в собственном темпе и без постоянного непосредственного контакта с преподавателем; преподаватель из основного носителя и транслятора знаний превращается в советника и консультанта учащегося. Преподаватель управляет процессом обучения, имея в своем распоряжении мощный инструмент - компьютер с его возможностями доставки, хранения, обработки всех видов информации для демонстрации учебной информации, тренировки и самоконтроля. Учащийся, в свою очередь, получает мощное технологическое средство поддержки самостоятельного интеллектуального труда

и доступа к информационной среде, не ограниченного пространством и способом передачи. Учащемуся становятся доступными знания о содержании и методе обучения, которые до сих пор были прерогативой преподавателя.

Интенсивная работа в условиях информационного комфорта активизирует познавательную деятельность учащихся и усиливает творческие компоненты труда преподавателя. Средства информационных и коммуникационных технологий в обучении освобождают преподавателя от множества функций, ставших рутинными в его повседневной деятельности.

Однако для использования новых широких возможностей образовательной информационной среды настоятельно требуются теоретическое осмысление и технологическая поддержка решения ряда практических задач, связанных с реорганизацией учебного процесса. В связи с этим одной из актуальнейших дидактических задач становится задача эффективного использования компьютера для управления учебным процессом самим учащимся.

Для реализации задач, стоящих перед современным образованием, нужна эффективная, гибкая, модульная система, базирующаяся на наиболее передовых технологиях и средствах обучения.

Отличительная черта современного этапа - поиск педагогами - исследователями способов применения формальных методов для описания процесса обучения с использованием аппаратов системного анализа, кибернетики, синергетики, с учетом, развитием и расширением понятий, принципов и достижений дидактики.

В качестве примера слабо структурированной области знаний рассмотрим предметную область «Информатика», которая неразрывно связана с информационными технологиями и с наиболее динамично развивающимся ресурсом мирового сообщества. В процессе обучения информатике это проявляется в постоянном обновлении версий изучаемых средств информационных технологий, появлении новых пользовательских сред и систем программирования, неизвестных учителю. В связи с этим можно определить, с нашей точки зрения, одну из важнейших проблем подготовки специалистов в области информатики и ИКТ: система подготовки должна обеспечивать такой уровень, который позволил бы специалистам в своей будущей профессиональной деятельности быстро адаптироваться к инновациям в области информационных технологий.

Информатика как научная дисциплина представляет собой стремительно развивающуюся область знаний, некоторые разделы которой уже устоялись и являются общепризнанными, а некоторые находятся в стадии становления.

Кроме того, бурное развитие средств ИКТ и сети Интернет, в последнее время породило ряд проблем, связанных с быстрым ростом объемов слабо структурированной, дублирующей информации, подлежащей хранению и обработке. Указанные недостатки ограничивают возможность смыслового поиска необходимой информации и доступ к ней. Над решением перечисленных проблем работают

многочисленные коллективы ученых и специалистов во всем мире, в частности, консорциум W3C, где реализуется концепция Семантического Web.

Как показывает изучение электронных образовательных средств, используемых в обучении, многие из существующих электронных курсов являются замкнутыми системами с жесткими моделями, не всегда позволяющими адаптировать к конкретному уровню знаний обучаемых. Недостатком существующих электронных образовательных средств также является отсутствие целостного восприятия учебной информации студентами. Использование интеллектуальных методов и моделей при разработке систем обучения позволяет устранить указанные недостатки.

2. Сравнительный анализ моделей представления знаний

Для представления знаний в интеллектуальных обучающих системах (ИОС) существуют различные способы. Наличие различных способов вызвано в первую очередь стремлением с наибольшей эффективностью представить различные типы предметных областей. Обычно способ представления учебного материала в интеллектуальных системах характеризуется моделью представления знаний.

Модели представления знаний обычно делят на *логические* (формальные), *эвристические* (формализованные) и смешанные.

В основе *логических моделей* представления знаний лежит понятие формальной теории. Примерами формальных теорий могут служить исчисление предикатов и любая конкретная система продукций. В логических моделях, как правило, используется исчисление предикатов первого порядка, дополненное рядом эвристических стратегий. Эти методы являются системами *дедуктивного типа*, т.е. в них используется модель получения вывода из заданной системы посылок с помощью фиксированной системы правил вывода. Дальнейшим развитием предикатных систем являются системы *индуктивного типа*, в которых правила вывода порождаются системой на основе обработки конечного числа обучающих примеров [Стюарт, Питер, 2006]. В логических моделях представления знаний отношения, существующие между отдельными единицами знаний, выражаются только с помощью тех небогатых средств, которые предоставляются синтаксическими правилами используемой формальной теории. В отличие от формальных моделей *эвристические модели* имеют разнообразный набор средств, передающих специфические особенности той или иной проблемной области.

Именно поэтому эвристические модели превосходят логические как по возможности адекватно представить проблемную среду, так и по эффективности используемых правил вывода. К эвристическим моделям, используемым в

экспертных системах, можно отнести *сетевые, фреймовые, производные* и *объектно-ориентированные* модели. Следует отметить, что производные модели, используемые для представления знаний в экспертных системах, отличаются от формальных производных систем тем, что они используют более сложные конструкции правил, а также содержат эвристическую информацию о специфике проблемной среды, выражаемую часто в виде семантических структур.

Как правило, в системах, основанных на знаниях, используется не одно, а несколько представлений. Исполняемые утверждения представляются либо в виде производных правил, либо в виде модулей (процедур), вызываемых по образцу. Для представления модели предметной области используются объектный подход или сетевые модели (семантические сети и фреймы).

3. Обоснование выбора многоуровневой иерархической адаптивной семантической модели для представления знаний в интеллектуальных обучающих системах

Применение объектно-ориентированного подхода в системах инженерии знаний выводит на первый план другую его особенность, а именно возможность естественной декомпозиции задачи на совокупность подзадач, представляемых достаточно автономными агентами, работающими со знаниями. На сегодняшний день это единственная практическая возможность работы в условиях экспоненциального роста сложности, характерного для систем, использующих знания.

Мы рассмотрели особенности наиболее распространенных моделей представления знаний. Следует отметить, что модель представления знаний в виде семантической сети структурно представляет собой граф. Как известно, «граф является очень характерным математическим объектом адаптации» [Растрин, 1981].

На основе результатов сравнительного анализа интеллектуальных моделей, в качестве основного способа представления слабо структурированных междисциплинарных областей знаний в ИОС мы выбрали адаптивные семантические модели (АСМ).

Для проектирования интеллектуальных обучающих систем, основанных на семантических моделях мы руководствовались теорией семантических сетей и других эвристических моделей представления знаний, а также основными научными подходами в получении знаний (конструктивный, аксиоматический и т.д.).

Преимуществом адаптивных семантических моделей представления знаний и непосредственно самого процесса обучения является наглядность описания предметной области, гибкость,

адаптивность к цели обучаемого. Однако, свойство наглядности с увеличением размеров и усложнением связей базы знаний предметной области теряется. Кроме того, возникают значительные сложности по обработке различного рода исключений. Для преодоления указанных проблем используют метод иерархического описания сетей (выделение на них локальных подсетей, расположенных на разных уровнях).

Приведенные выше сведения, соображения и рекомендации позволили систематизировать и обобщить основные методологические положения по представлению и контролю знаний в интеллектуальных системах обучения [Шихнабиева, 2009].

4. Пример использования адаптивных семантических моделей для представления знаний

В настоящее время существуют различные виды образовательных средств: учебники, методические пособия, справочники и т.д., в том числе и электронные образовательные средства. Однако существующие электронные учебники по абстрактным дисциплинам существенно не отличаются от учебных пособий в твердом носителе. Чтобы найти связи между понятиями учебной дисциплины приходится многократно листать весь учебник и искать необходимую информацию. Представление учебного материала по абстрактным дисциплинам в виде многоуровневой иерархической адаптивной семантической модели (рис.1) позволяет создать структурированный учебник, показывающий связи между понятиями предметной области, что важно при организации обучения на основе информационных и коммуникационных технологий.

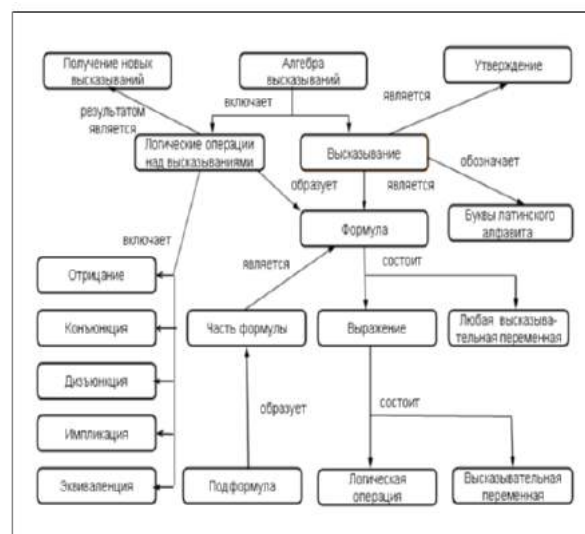


Рисунок 1 - Семантическая модель по теме «Алгебра высказываний»

Как показывает опыт разработки семантических моделей по учебной дисциплине «Математическая

логика», сам процесс построения моделей способствует эффективному приобретению знаний. Поэтому обучение можно вести не только по разработанным преподавателем АСМ, но и давать студентам задания по их разработке, что способствует лучшему усвоению учебного материала. Приведенная на рис.1 модель учебного материала по теме “Алгебра высказываний” представляет основные понятия данного раздела математики и показывает причинно–следственные отношения между ними.

5. Интеллектуальная обучающая система, основанная на АСМ

Предложенные подходы к представлению и контролю знаний заложены в основу интеллектуальной обучающей системы, которая используется в учебном процессе ряда вузов.

Программная оболочка интеллектуальной обучающей системы реализована в объектно–ориентированной среде программирования Delphi. Система программирования Delphi имеет в своем составе инструментальную оболочку с множеством компонентов. Благодаря компонентному подходу к программированию, программу можно собирать как конструктор, настраивая каждый компонент для решения той или иной задачи.

На рис.2. приведена схема связи таблиц базы данных интеллектуальной обучающей системы.

База данных состоит из 16-ти таблиц, схема связи которых показана на рисунке. Как видно из схемы основной таблицей БД является таблица «Сети». Она объединяет в единую структуру все остальные таблицы БД.

В качестве связей таблиц применяются два вида связей: «один-к-одному» и «один-ко-многим».

Исходя из приведенных схем на рис.4, можно отметить сложность структуры базы данных рассматриваемой обучающей системы. Подобная структура базы данных позволяет автоматизировать процесс обучения студентов и контроля их знаний.

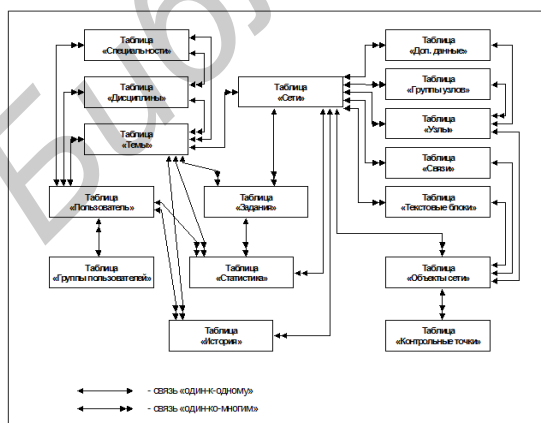


Рисунок 2 - Схема связей таблиц базы данных интеллектуальной обучающей системы

Заключение

Итак, представление учебного процесса в виде адаптивных семантических моделей позволяет обеспечивать: индивидуальный темп обучения при реализации обратной связи; деятельностный подход при выборе решения задачи с учетом учебных ситуаций; связь новых понятий с существующими понятиями и представлениями, что улучшает понимание; осуществление глубокой обработки знаний, что повышает способность применять знания в новых ситуациях.

Преимущества предлагаемой нами модели представления знаний особенно значимы при контроле знаний обучаемых. Разработанная нами методика контроля знаний позволяет также структурировать вопросы и создавать адаптивные тесты.

Такая организация контроля знаний способствует качественному обучению, поскольку обучаемые анализируют базовую структуру изучаемых понятий и представлений, связывая с ними новые понятия.

Библиографический список

- [Давыдов, 1972] Давыдов В. В. Виды обобщения в обучении (логико-психологические проблемы построения учебных предметов)/В.В.Давыдов// М.: Педагогика, 1972. – 424 с.
- [Стюарт, Питер, 2006] Стюарт Рассел, Питер Норвиг. Искусственный интеллект: Современный подход/ Рассел Стюарт, Норвиг Питер// М.: Вильямс, 2006. - 1407 с.
- [Растрингин, 1981] Растрингин Л. А. Адаптация сложных систем / Л. А. Растрингин// Рига: Зинатне, 1981. — 375 с.
- [Шихнабиева, 2009] Шихнабиева Т.Ш. Методические основы представления и контроля знаний в области информатики с использованием адаптивных семантических моделей/ Т.Ш..Шихнабиева // Дисс... д.-ра пед. наук. М., 2009. - 365 с.

METHODS AND MODELS SEMANTIC REPRESENTATION OF KNOWLEDGE IN INTELLECTUAL SYSTEMS EDUCATIONAL PURPOSES

Shikhnabieva T.Sh

*Institute of Education Management
Russian Academy of Education,
Moscow, Russia
shetoma@mail.ru*

Modern information technology and the rapid expansion of the diversity of the network of educational services has caused a stream of innovations on the reorganization of existing educational systems at all levels of education. The paper presents some approaches to improve existing systems of educational appointment using intelligent methods and models.



УДК 004.55

О СОЗДАНИИ АДАПТИВНОЙ СИСТЕМЫ ГЕНЕРАЦИИ МУЛЬТИМОДАЛЬНЫХ ОБУЧАЮЩИХ МАТЕРИАЛОВ

Хусаинов А.Ф. *, Русецкий К.В.**

**Институт прикладной семиотики Академии наук Республики Татарстан
Казанский (Приволжский) федеральный университет,*

Казань, Россия

Khusainov.aidar@gmail.com

***Белорусский государственный университет информатики и радиоэлектроники
Минск, Белоруссия*

rusetski.k@gmail.com

В статье представлены результаты создания обучающей системы, особенностью которой является возможность адаптации под возможности обучаемого. Целью работы является обеспечение оптимального усвоения информации за счет учёта факторов, влияющих на восприятие мультимодальных текстов. Разработана архитектура системы многомодального синтеза текстов, для обеспечения механизма адаптации обучающего материала под особенности восприятия конкретного человека разработана модель представления знаний об одном и том же факте в виде совокупности данных разных модальностей.

Ключевые слова: мультимодальные данные; генерация обучающих материалов; синтез текстов.

Введение

Задача построения интеллектуальных систем обучения является актуальной, множество программных и инструментальных средств внедряются в учебный процесс с целью повышения качества образования. В данной работе мы предлагаем подход, главная идея которого заключается в создании обучающей системы, обладающей возможностью адаптироваться к каждому обучаемому на основе особенностей восприятия им информации в различных модальностях.

Для решения поставленной задачи необходимо определить основные факторы, влияющие на степень восприятия мультимодальных текстов. В частности, важно дать определение понятию модальности. Гюнтер Кресс определяет модальность как результат социально-исторического оформления средств, выбираемых обществом для репрезентации [1] или социально обусловленный семиотический ресурс смыслообразования [1]. В отечественных исследованиях даётся такое определение термина «мультимодальность» – это тип внешнего стимула, воспринимаемого одним из чувств человека, в первую очередь зрением и слухом [2]. При этом

подходе акцент ставится не на системах репрезентации, а на восприятии информации, что подходит для использования в нашем исследовании, направленном на создание практически действующей программной системы. Развитие идей мультимодальности можно найти в работах А. А. Кибрика, который отмечает, что адекватное представление о естественной коммуникации может сложиться только за счет конвергенции подходов к изучению различных каналов обмена информацией. В качестве основных информационных каналов он рассматривает вербальный, просодический и визуальный, каждый из которых дробится на несколько подканалов [2]. Их разнообразие способно обеспечить максимальный охват информации получателем, поэтому исследоваться они должны в совокупности, с учетом взаимозависимости их компонентов и пропорции их представленности.

Так, можно констатировать, что актуальность исследований в русле мультимодальности во многом обусловлена пониманием необходимости задействовать все доступные каналы передачи обучающего материала оптимальным с точки зрения качества восприятия обучаемым способом. Потенциал компьютерных обучающих систем

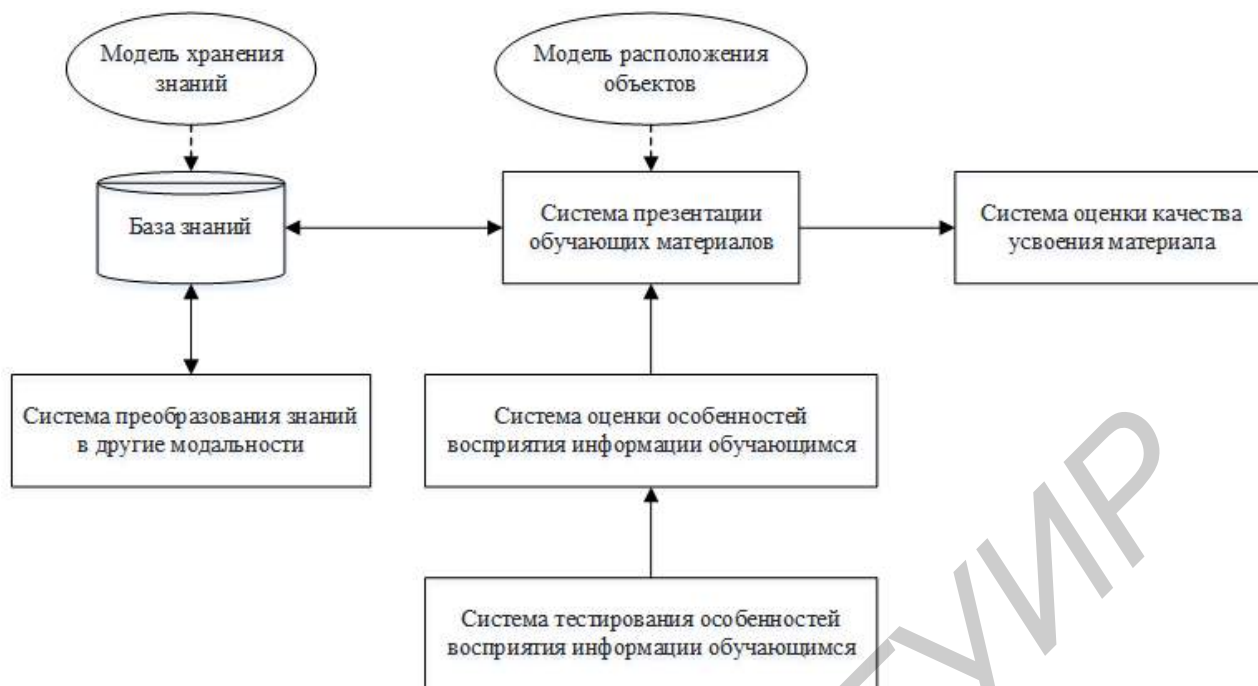


Рисунок 1 – Общая архитектура работы адаптивной системы обучения

позволяет варьировать модальности для максимального усвоения информации, для чего необходимо учитывать ряд факторов, таких как особенности восприятия информации получателем.

Структура данной работы предполагает описание текущих результатов создания системы адаптивной генерации мультимодальных обучающих материалов. Архитектура системы и планы по проведению экспериментальных исследований представлены в пункте 2, основные выводы по результатам исследования приведены в заключении.

1. Архитектура системы адаптивной генерации мультимодальных обучающих материалов

Основываясь на проведенном аналитическом обзоре работ, посвященных проблеме усвоения информации, были выделены факторы, оказывающие наибольшее влияние на качество усвоения информации: модальность данных, размер и расположение данных.

Данные параметры были положены в основу при построении архитектуры системы генерации обучающих материалов. Кроме того, проектируемая система должна обеспечивать возможность адаптации учебного курса под особенности восприятия конкретного обучаемого. Это ведёт к необходимости разработки и внедрения дополнительных подсистем:

- модуль тестирования обучаемого (с целью выявить степень усвоения информации через различные каналы восприятия);

- модуль адаптации курса к особенностям конкретного обучаемого.

Общая архитектура функционирования адаптивной системы обучения на основе многомодальной информации представлена на Рис. 1.

Функциональность работы системы с многомодальной информацией реализуется за счет наличия базы знаний, основанной на модели хранения информации в различных модальностях. Так, каждое знание в учебном курсе должно быть представлено в текстовом, аудио, видео и графическом форматах. Учитывая возможные сложности по созданию учебных курсов большого объема, удовлетворяющих условию полноты представления каждого из знаний, в систему предполагается внедрить модуль преобразования знаний в другие модальности. Например, текстовое представление знаний может быть получено из аудиосигнала, благодаря использованию технологии автоматического распознавания речи [3].

Полный набор взаимосвязей между различными представлениями знаний представлен на Рис. 2, а для его реализации необходимо создание и использование следующих подсистем:

- система автоматического синтеза речи (текст – речь);
- система автоматического распознавания речи (речь – текст);
- система выделения информации из видео файла (видео – изображение, видео – аудио);
- система генерации видео файла (изображения + аудио – видео);

- система генерации изображений (текст – изображение).

В состав разработанных средств входит рабочее место преподавателя, которое позволяет создавать и настраивать учебные курсы для отображения в обучающей системе. Преподаватель вводит необходимые для усвоения курса знания в имеющихся форматах, настраивает веса значимости информации. Далее, система преобразования знаний создаёт новые представления введенной информации в непредставленных модальностях. Преподаватель можно группировать знания для их отображения в стиле слайдов презентации с возможностью указания, какой обучающий материал должен отображаться в конкретной указанной им модальности, а какой должен подстраиваться под особенности восприятия обучаемого.

Для работы разработанной системы на этапе обучения, в первую очередь, требуется провести тестирование студентов на определение степени восприятия через различные каналы передачи информации. Тест состоит из короткой серии вопросов, его прохождение занимает 5-7 минут. На основе результатов данного тестирования и введённых преподавателем весов для знаний система формирует способ отображения обучающего материала для каждого конкретного обучаемого.

Заключение

В данной работе предложен способ построения обучающей системы, способной адаптироваться к возможностям обучаемого воспринимать информацию в той или иной модальности. Создан работающий прототип системы. С помощью созданного прототипа преподавателями Казанского федерального университета осуществляется разработка курса изучения испанского языка для студентов высших учебных заведений. В дальнейшем на основе созданного учебного курса планируется провести оценку качества работы созданных программных средств за счет сравнения степени усвоения студентами предложенного учебного материала.

Работа выполнена при финансовой поддержке РФФИ (проект № 15-57-04085) «Модели и средства мультимодального синтеза текстов для интеллектуальных обучающих систем».

Библиографический список

- [Кибрик, 2010] Кибрик А. А. Мультимодальная лингвистика / А. А. Кибрик // Когнитивные исследования. – 2010. – № 4. – С. 134-152.
- [Khusainov, 2014] Khusainov A. F. Towards Automatic Speech Recognition for the Tatar Language / A. F. Khusainov, Dz. Sh. Suleymanov // Proc. of the International Conf. on Turkic Language Processing. (Istanbul, November 6-7, 2014). – Istanbul: Özkaracan Matbaacılık-Bağcılar, 2014. – P. 33-37.
- [Kress, 2010] Kress G. Multimodality: A Social Semiotic Approach to Contemporary Communication. – 1st ed. – New York: Routledge, 2010. – P.11, 79.

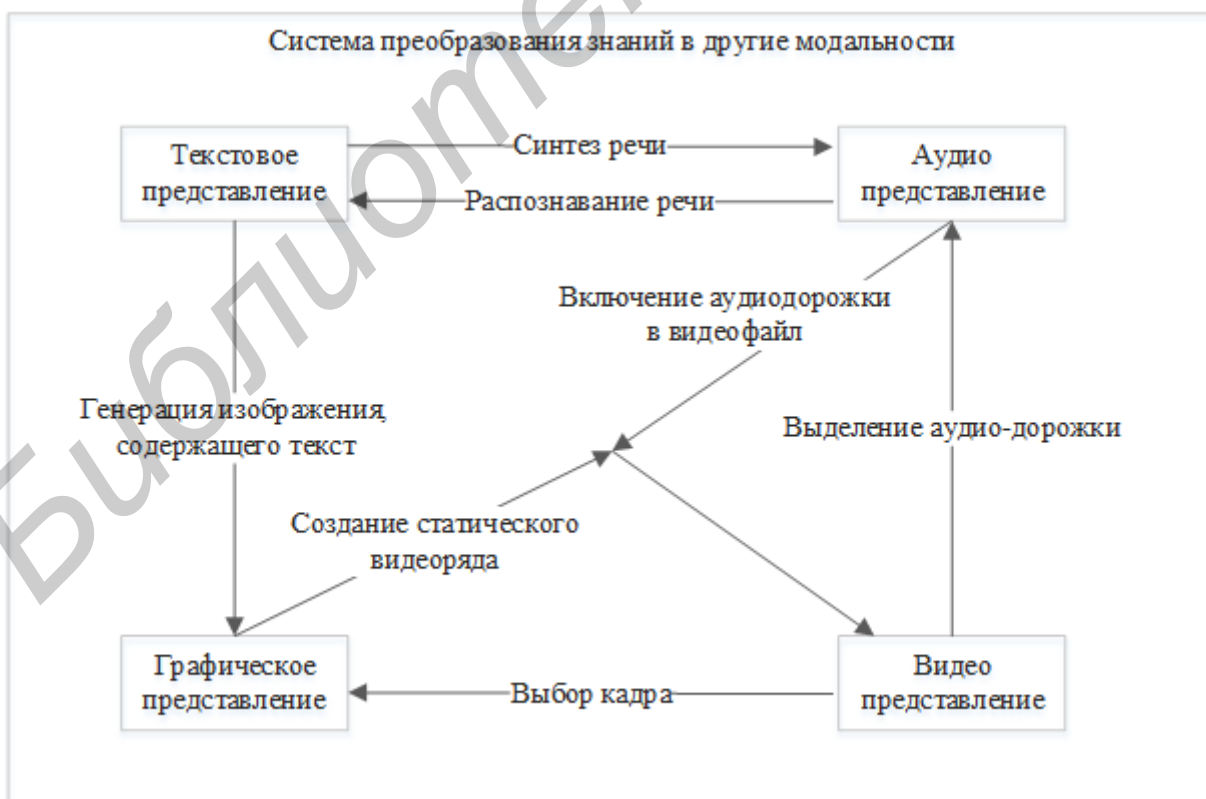


Рисунок 2 – Схема работы системы преобразования знаний в другие модальности

ABOUT THE CREATION OF ADAPTIVE SYSTEM FOR MULTIMODAL LEARNING MATERIALS GENERATION

Khusainov A.F.^{*}, Ruseckiy K.V.^{**}

**Institute of Applied Semiotics of the Tatarstan Academy of Sciences, Kazan, Russia
Kazan (Volga region) federal university, Kazan, Russia*

khusainov.aidar@gmail.com

***Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

rusetski.k@gmail.com

In this paper we describe learning system that can adapt its content depending on a concrete student. The purpose of creating this system is to maximize the amount of information that students can learn based on factors which influence the degree of information perception. We describe the architecture of learning system, the major aspects of multimodal knowledge generating subsystem, the approach to store multimodal teaching materials.

Introduction

The creation of the intellectual systems of education is an urgent issue; many software tools are used in the learning process to improve the education quality. In this paper, we suggest the approach where the main idea is to create the learning system that will be adaptive for each student, based on his or her information perception in different modalities.

It can be stated that the actuality of the research devoted to the multimodality is due to the understanding the necessity of using all the available channels for optimal learning material's transmission concerning the education quality. The potential of computer learning systems allows varying modalities for maximum mastering of information. It is necessary to take into consideration factors such as peculiarities of information perception.

Main Part

Based on the analytical review of the papers devoted to the problem of information perception there has been found some factors that mostly influence the quality of information perception, data modality, size and location.

These features are taken as the basis while creating the architecture of system generation of the learning material. Besides, the projected system should give the opportunity of adaption the curriculum according to the perception of a concrete student. It leads to the necessity of the creation and usage additional subsystems:

- module for testing a student (in order to find out the level of information perception by different channels);

- module for adaptation the curriculum for peculiarities of a concrete student.

The functionality of the system with multimodal information is implemented with the help of having knowledge data, based on the model of keeping information in different modalities. In this way, every knowledge in the curriculum should be in text, audio, video and graphic format. Taking into account the obstacles of creation big courses, satisfying the condition of full knowledge presentation, it is suggested to enlarge the system with a module of knowledge transformation into other modalities. For instance, the text presentation of knowledge can be got from the audio signal due to the usage of the technology of the automatic speech recognition [3].

The developed tools consist of teacher's office, that allows creating and adjusting the courses for depicting them in the learning system. A teacher fills in the necessary knowledge in several formats and then adjusts weight rates. On the first step, the students are suggested to get the test for identifying the degree of perception by different channels of information transmission. After that, the system forms the way of depicting the learning material for a concrete student.

Conclusion

In this paper, we present the system that can be used to create learning materials and to use them in learning process. The main feature of this system is the ability to work with multimodal representations of knowledge and to adapt the way each person see learning materials based on his personal characteristics of perception.

In this paper, the way of creating the learning system is suggested that is able to adapt to student's peculiarities of perceiving information in this or that modality. The working prototype of the system is created.

The teachers of Kazan federal university carry out the production of the course for learning Spanish for students in the institutes of higher education with the help of this prototype. It is planned to estimate the quality of the created software tools based on the made course owing to the comparison of mastery the suggested learning material.

The work has been implemented with the financial support of RFBR (project № 15-57-04085) "Models and resources of multimodal synthesis of texts for intellectual learning systems".



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РАЗРАБОТКА МОДЕЛИ УЧЕТА ВЛИЯНИЯ ЭМОЦИЙ НА ФОРМИРОВАНИЕ ИНДИВИДУАЛЬНОЙ ТРАЕКТОРИИ ОБУЧЕНИЯ СТУДЕНТА НА ОСНОВЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

Козлов О.А., Михайлов Ю.Ф.

*Федеральное государственное бюджетное научное учреждение
«Институт управления образованием Российской академии образования»
г. Москва, Россия*

ole-kozlov@yandex.ru

*ВА РВСН им. Петра Великого
(филиал в г. Серпухове), Россия*

mikhayurij@yandex.ru

В статье приводится описание подхода по формированию модели учета влияния эмоций на формирование индивидуальной траектории изучения студентом некоторой предметной области знаний с помощью искусственной нейронной сети.

Ключевые слова: эмоции, активные нейронные сети, вектор параметров оценки уровня эмоций, усвоения знаний дидактической единицы, интеллектуальная обучающая система.

Введение

Актуальным с точки зрения личностно ориентированного обучения является понятие индивидуальной траектории обучения и изучения. Формирование индивидуальной траектории изучения является задачей со множеством неопределенных параметров, поэтому предлагается для оценки результатов ее решения применять искусственную нейронную сеть. Важной задачей в данном случае является правильный подход к каждому студенту, а также оценка степени понимания материала, заинтересованности в обсуждаемом вопросе студента. Для этого необходимо использование систем, позволяющих моделировать поведение субъектов образовательного процесса в зависимости от текущего состояния учебного процесса и эмоционального состояния студента. Модуль распознавания состояния студента на основе нейронной сети должен включать информацию о различных эмоциональных и психологических состояний человека, их особенности, а также анализ систем и нейронной сети для распознавания образов. Для решения отмеченных проблем чрезвычайно полезным может быть продолжение и развитие компьютерного моделирования активных нейронных сетей.

1. Разработка модели учета влияния эмоций на формирование индивидуальной траектории обучения студента на основе нейросетевых технологий

Сложность современных технологических процессов, переход на ФГОС 3 поколения предъявляет повышенные требования к образовательным технологиям, и требуют адекватных изменений в сфере образования. В этой связи все более актуальными становятся технологии инновационного обучения, основанные на формировании индивидуальной траектории обучения студента.

Важной задачей в данном случае является правильный подход к каждому студенту, а также оценка степени понимания материала, заинтересованности в обсуждаемом вопросе студента.

Распознавание эмоционального состояния системы взаимодействия преподавателя и студента должно включать информацию о различных методиках и языках, позволяющих описывать и моделировать процесс взаимодействия параллельных систем: преподавателя и студента. Модуль распознавания состояния студента на основе нейронной сети должен включать информацию о различных эмоциональных и

психологических состояний человека, их особенности, а также анализ систем и нейронной сети для распознавания образов [Рабинович, Мюезинолу, 2010].

Мы рассматриваем эмоции как один из механизмов для мотивации обучения и планирования учебной деятельности, делая упор на обучении, как основе для организации поведения на занятии. Что касается мотивации, то надо иметь в виду, что мотивация базируется отчасти на инстинктивных процессах, а отчасти, является результатом обучения [Шамис, 2006].

Средству обучения, функционирующему на базе информационных и коммуникационных технологий, при необходимости (по Роберт И.В.) можно частично передать функции обучающего: контроль результатов обучения; предоставления заданий, адекватных уровню обучающегося. Индивидуальная траектория изучения студента выражает цель обучения и содержит информацию о состоянии знаний обучаемого, его эмоциях.

Программа дисциплины строится по модульному принципу, разработанному и представленному в монографии доктора педагогических наук О. А. Козлова «Теоретико-методологические основы информационной подготовки курсантов военных учебных заведений» [Козлов, 2001].

В каждом модуле выделяются элементы знаний – дидактические единицы (ДЕ), представленные в учебной программе, устанавливаются структурные и семантические связи между элементами и разрабатываются семантические модели знаний для этих модулей. Связи между элементами знаний реализуются по гипертекстовой технологии, в виде ссылок. В качестве ссылки берется ключевое слово, терм, понятие, которое присутствует в структурно связанных между собой элементах знаний [Михайлов, 2001].

Влияние эмоций на студента проявляется в ситуации, когда студент должен осуществить переход от усвоения одной дидактической единицы к другой. Студент должен сменить сценарии предъявления и усвоения новых знаний. Сменить процесс усвоения, переключить процесс предъявления. Перейти от усвоения одной дидактической единицы к другой дидактической единице. Переключение процессов или смена дидактических единиц требует активизации гиперссылки, на что уходит определенное время $t_{\text{перекл}}$

Назовем это время в нормальной рабочей обстановке, в обстановке наличия мотивации к изучению, оптимальным временем переключения. В течение этого промежутка времени мы считаем, не происходит потеря знаниевой информации из сознания студента.

Однако, наличие эмоций в сознании студента, вносит изменения в мыслительный процесс. Студент, находясь во власти эмоций, отвлекается от процесса усвоения знаний, прерывает процесс

усвоения знаний. Он находится во власти эмоций, его сознание не воспринимает знаниевую информацию, и не активизирует гиперссылку. Возникает задержка во времени для переключения дидактических единиц. Временная задержка в процессе перехода от изучения, усвоения одной дидактической единицы к усвоению другой ДЕ.

Задержка во времени, ее числовое значение, величина отклонения от оптимального времени переключения сигнализирует о факте появления эмоций в сознании студента.

Временной интервал формирует величину входного вектора, который запускает ИНС формирования эмоционального состояния студента, и оценки этого состояния. Значение величины оценки эмоционального состояния вызывает процедуру управления характером предъявления содержания дидактических единиц.

Управление предполагает внедрение мультимедийного контента в содержание ДЕ [Роберт, 2009]. Использование релаксационных, расслабляющих сценариев урока и после этого, акцентирующих, мобилизационных сценариев.

Характер сценария определяется значением результата работы ИНС.

Мы считаем, что необходимо вывести студента из эмоционального плена, порабощения. Надо нейтрализовать эмоции, влияние эмоций на характер процесса усвоения знаний [Рабинович, Мюезинолу, 2010]. Погасить задержку во времени переключения сценариев предъявления содержания дидактических единиц. Возможные пути преодоления эмоций [<http://www.intuit.ru>]:

- предложить выбор фотографий,
- или цветных изображений,
- выбор музыкальных фрагментов,
- нарисовать геометрическую фигуру.

Варианты ответов формируют четырехэлементный входной вектор для нейронной сети. Предлагается сделать это с целью создания комфортного интерфейса для последующей работы, продолжения изучения учебного материала.

Предлагается применить активные ИНС А-сети для решения поставленной проблемы [Шамис, 2006]. Эти ИНС позволяют сформировать эмоциональную модель студента и вычислить величину эмоционального погружения студента - ЭМП. Эта величина увязана с временем задержки переключения сценария предъявления ДЕ - $t_{\text{задер}}$:

$$t_{\text{задер}} = t_{\text{перкл}} - t_{\text{оптим}} \quad (1)$$

Для активизации работы ИНС можно установить, что для четырех видов эмоций значения вариантов погружения следующие;

- Радость – 0.25.
- Грусть – 0.25.
- Безразличие – 0.25.
- Злость – 0.25.

В дальнейшем, в ходе экспериментальной настройки ИНС эти значения могут быть скорректированы.

Вариант модели учета влияния эмоций на формирование индивидуальной траектории обучения студента на основе нейросетевых технологий приведен на рисунке 1.

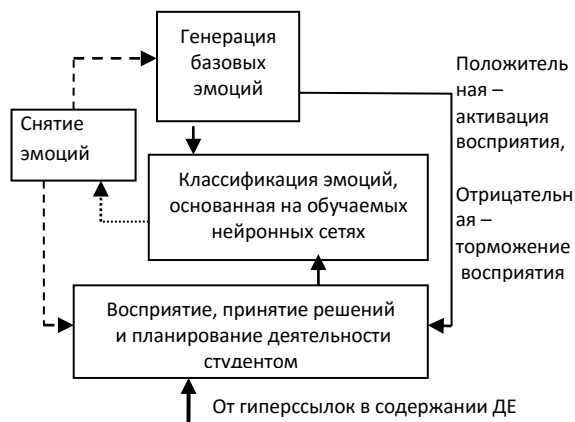


Рисунок 1 – Общая схема предлагаемой модели

В зависимости от текущего уровня знаний по изучаемой дисциплине движение по траектории изучения дидактической единицы можно разбить на три основных направления: возврат назад по траектории к предыдущему дидактическому элементу изучаемой дидактической единицы учебного курса, если обучаемый неудовлетворительно справился с предлагаемыми тестовыми заданиями по изучаемой теме или разделу; движение вперед к новому дидактическому элементу, следующей порции учебного материала, определенной программой курса. В случае если предыдущая тема или раздел учебного курса были освоены на хорошем или отличном уровне выполняется движение вперед к новой дидактической единице учебного материала. Каждое выделенное направление движения по индивидуальной траектории изучения включает несколько вариантов, позволяющих индивидуализировать и дифференцировать процесс обучения.

Модель активной нейронной сети строится, с использованием в ее архитектуре постоянной внутренней задачи, состоящей в необходимости непрерывного поддержания оптимальных функциональных состояний нейронов и требующей их постоянной активности. Внутренняя задача, во-первых, возникает как следствие отклонения от оптимума функциональных состояний не работающих (не возбуждающихся) нейронов за счет появления у студента эмоций. Решается эта задача, при групповом взаимовозбуждении нейронов, за счет того, что эмоции изменяются. Внутренняя задача возникает при постановке внешней задачи восприятия или поведения, сводящейся, в конечном счете, к внешнему возбуждению какой-то группы нейронов сети. Внешняя задача, в виде эмоций, нарушает режим общей равномерной оптимизации

и приводит к отклонению от оптимума функциональных состояний нейронов.

Решение задачи должно приводить к восстановлению режима оптимизации, что должно улучшать функциональное состояние нейронов сети. Таким образом, необходимость решения задачи не задается мозгу извне, а становится его внутренней целью. Направленные на оптимизацию состояния структурные перестройки нейронной сети должны состоять в образовании синергических взаимодействующих групп нейронов [Шамис, 2006]. Эти группы (ансамбли нейронов) становятся основой информационной нейронной модели проблемной среды.

Анализ литературы [Шамис, 2006], [Круглов, 2001], [Рутковская, 2006] позволил предложить для реализации схемы активной нейронной сети с возвратным торможением схему на основе классического трехслойного перцептрона. Так же, как в перцептроне, разделение пересекающихся на входном уровне информационных воздействий происходит на уровне групп нейронов первого слоя и связей, идущих к нейронам от сенсорных элементов. Принципиальное отличие - возвратное торможение и накапливающаяся неустойчивость нейронов, вызванное проявлением эмоций. Последнее лежит в основе обучения и свойства активности. В описанной трехслойной сети возможно запоминание и спонтанное воспроизведение временных последовательностей, т. е. возможны ассоциации по смежности во времени.

Для моделирования процедур возвратного торможения предлагается использовать искусственные нейроглиальные сети (ИНГС), в которых глиальные клетки являются такой же частью сети, как и нейроны [http://aideus.ru], что проиллюстрировано на приведенном ниже рисунке 2

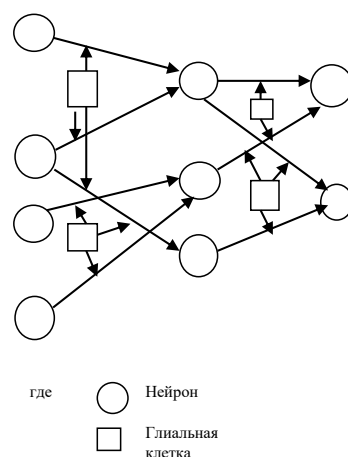


Рисунок 2 – Модель нейронной сети с торможением

Определенные типы глиальных клеток, астроциты, являются активными участниками диффузной передачи информации в мозге. Астроциты способны к генерации кальциевых

осцилляций и образуют сети посредством гап-контактов [Bezzi & Volterra, 2001]. Астроциты влияют на синаптические связи нейронов.

Сбалансированная работа синаптического возбуждения/торможения и нейромодуляторных систем лежит в основе передачи, обработки и сохранения информации в мозге, а также генерации ритмов мозга, которые являются своего рода тактовыми рабочими частотами его структур [Vinogradova, 2001]. Эти же механизмы лежат в основе формирования эмоциональной картины мозга.

Поток эмоциональной информации через глию приводит к модификации весов связей между нейронами, что приводит к ожидаемой блокировке обучения. В частности, с помощью введения специально подобранной системы глиальных клеток, в многослойный перцептрон, он может быть расширен таким образом, чтобы он обучался методом обратного распространения ошибки без всякого внешнего алгоритма.

Итак, прохождение сигнала через некоторые синапсы влияет не на активность нейронов, а на проводимость других синапсов. Это означает, что «веса связей» между нейронами могут меняться не только медленно в процессе обучения, но и динамически, под влиянием текущей информации эмоций.

От нейронов первого слоя на нейроны второго слоя образуются только возбуждающие связи. От глиальных клеток первого слоя на глиальные клетки второго слоя образуются только тормозящие связи. В физиологии подобные схемы взаимодействия нейронов называют структуры с возвратным торможением [Шамис, 2006].

В отсутствие эмоций тормозящая связь, идущая к нейрону первого слоя от элемента второго слоя, ничем не отличается от связей внутри первого слоя, и формируемый ею потенциал входит в алгебраическую сумму потенциалов нейрона, что соответствует стабильному процессу усвоения знаний. При наличии эмоций тормозящая связь, идущая к нейрону первого слоя от элемента второго слоя, полностью запрещает срабатывание нейрона на время уменьшения создаваемого ею тормозного потенциала до некоторой величины. Все это время происходит независимая суммация всех других создаваемых на нейроне потенциалов.

Эмоции и мышление это активные процессы, появляющиеся в результате изменений в организации и силе связей между различными центрами мозга непрерывно во времени, то есть динамический отклик мозга на информацию об окружающем и внутреннем мире. Динамические переменные описывающие динамику эмоций и когнитивных функций и их взаимодействие образуют совместное рабочее или фазовое пространство [Рабинович, Мюезинолу, 2010].

Исходя из этих соображений, математическую модель нейронной сети можно представить в виде нелинейной системы дифференциальных

уравнений, полученных на основе экспериментальных исследований биофизических процессов в нейронах и астроцитах (глиальных клетках).

Сети взаимодействующих клеток описываются многомерными системами дифференциальных уравнений [Прокин, Симонов, Казанцев, 2012]. Динамику межнейронных взаимодействий и астроцитов предлагается моделировать уравнениями Ли-Ринцеля. В качестве модели отдельного нейрона предлагается использовать модель Ходжкина-Хаксли [Hodgkin & Huxley, 1952].

Для моделирования предлагается использовать тщательно проверенные, задокументированные симуляторы нейронных сетей типа NEURON – симулятор [Прокин, Симонов, Казанцев, 2012].

Заключение

Интеллектуальная информационная система организации учебного процесса, на основе активной нейронной сети с возвратным торможением, позволяет учитывать влияние эмоций студента на формирование индивидуальной траектории изучения студентом, курсантом некоторой предметной области знаний. Сеть сохраняет, анализирует и объединяет результаты контроля эмоций, сформированные на основе учета частоты срабатывания гиперссылок. По результатам контроля формируется мультимедийный контент для содержания дидактических единиц модуля знаний, для нейтрализации эмоций.

В данной статье предложен метод контроля индивидуальной траектории изучения студентом некоторой предметной области знаний, с помощью активной нейронной сети с возвратным торможением, описывающих состояние и поведение студента во время изучения дидактической единицы модуля знаний.

Для моделирования такой сети предлагается математический аппарат на основе систем дифференциальных уравнений, модель нейрона Ходжкина-Хаксли и среда моделирования на основе симулятора нейронных сетей.

Библиографический список

- [Козлов, 1999] Козлов О.А. Развитие методической системы обучения информатике курсантов военно-учебных заведений Министерства обороны Российской Федерации. Дисс.... докт. пед. наук. – Серпухов. 1999.
- [Козлов, 2001] Козлов О.А. Теоретико-методологические основы информационной подготовки курсантов военно-учебных заведений. Монография. - М.: МО, 2001. – 328 с.
- [Круглов, 2001] Круглов В.В., Дли М.И., Голунов Р.Ю. Нечеткая логика и искусственные нейронные сети. – М.: Горячая линия – Телеком, 2001.
- [Михайлов, 2001] Михайлов Ю.Ф. Технология информационной подготовки курсантов в условиях моделирования экстремальных ситуаций профессиональной деятельности. Дисс.... канд. пед. наук. – Москва. 2001.
- [Прокин, Симонов, Казанцев, 2012] Прокин И.С., Симонов А.Ю., Казанцев В.Б. Математическое моделирование нейродинамических систем: Электронное учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2012. – 41 с.

[Роберт, 2009] Роберт И.В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты), 2-е издание, дополненное.– М.: ИИО РАО, 2009.

[Рабинович, Мюезинолу, 2010] Нелинейная динамика мозга: эмоции и интеллектуальная деятельность – М.: Успехи физических наук. Обзоры актуальных проблем. Т. 180, №4, апрель 2010.

[Рутковская, 2006] Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы //Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с польск. И.Д. Рудинского - М.: Горячая линия - Телеком, 2006. - 452с.: ил.

[Шамис, 2006] Пути моделирования мышления. Активные синергетические сети, мышление и творчество, формальные модели поведения и «распознавания с пониманием».– М.: КомКнига, 2006. – 336 с.

[Bezzi & Volterra, 2001] Bezzi P., Volterra A. A neuron-glia signalling network in the active brain // Curr Opin Neurobiol. 2001. V.11. № 3. P. 387-394.

[Hodgkin & Huxley, 1952] Hodgkin A. L., Huxley A. F. A quantitative description of membrane current and its application to conduction and excitation in a nerve // Journal of Physiol-ogy (London). 1952. V.117, P. 500-544.

[Vinogradova, 2001] Vinogradova O.S. Hippocampus as comparator: role of the two input and two output systems of the hippocampus in selection and registration of information // Hippocampus. 2001. V.11. №5. P. 578-598.

[http://www.intuit.ru/studies/professional_skill_improvement_s/1598/courses/423/lecture/9637] Активные нейронные модели (А-сети).

[<http://aideus.ru>] AIDEUS — Сильный Искусственный Интеллект.

DEVELOPMENT OF A MODEL CONSIDERING THE INFLUENCE OF EMOTIONS ON THE FORMATION OF INDIVIDUAL TRAJECTORY OF STUDENT LEARNING BASED ON NEURAL NETWORK TECHNOLOGY

Kozlov O.A., Mikhailov J.F.

*Federal state budgetary scientific institution
"Institute of education management, Russian
Academy of education",
Moscow, Russia*

ole-kozlov@yandex.ru

*VA SRF them. Peter The Great Va strategic
missile forces Peter The Great(branch in
Serpukhov), Russia*

mikhayurij@yandex.ru

The article describes the approach to the formation of the assessment of individual trajectory of studying a certain subject area knowledge through a hybrid artificial neural network. Key words: the vector of the parameters of assessment of quality of learning, fuzzy neural networks, intelligent training system.

Introduction

Relevant from the point of view of learner-centered learning is the concept of an individual trajectory of teaching and learning. The formation of individual trajectory of learning is a task with many uncertain parameters, it is therefore proposed to assess the results of its decision to apply an artificial neural network. An important task in this case is the right approach to each student, and assess the degree of understanding of the material, the interest in the subject student. This

requires the use of systems that simulate the behavior of subjects of educational process depending on the current state of the educational process and emotional state of the student. The module recognition status of a student on the basis of neural networks should include information about various emotional and psychological States of the person, their characteristics, and analysis systems and neural networks for pattern recognition. To solve the above problems is extremely useful may be the continuation and development of computer modelling of the active neural networks.

Main Part

The complexity of modern technological processes, the transition to the GEF 3 generations imposes requirements on educational technology, and require adequate changes in education. In this regard, become more and more important technologies in innovative learning, based on the formation of individual trajectory of student learning.

An important task in this case is the right approach to each student, and assess the degree of understanding of the material, the interest in the subject student.

Recognition of the emotional state of the system of interaction between the teacher and the student should include information about the various techniques and languages that allow to describe and model the process of interaction of parallel systems: the teacher and the student.

We consider emotions as a mechanism for motivating learning and planning of learning activities, focusing on learning as the basis for the organization of behavior in class.

The influence of emotions on students is manifested in a situation where the student must make the transition from didactic learning one unit to another. The student must change scenarios presentation and assimilation of new knowledge.

However, the presence of emotions in the mind of the student, makes changes in the thought process. The student, being at the mercy of emotions, distracted from the learning process, interrupts the learning process. It is in the power of emotions, his consciousness does not perceive knowledge, information, and activates a hyperlink.

The time interval that forms the value of the input vector, which launches ince the formation of the emotional state of the student, and evaluation of this condition. The value assessment of the emotional state causes the control nature of the presentation of the content of didactic units.

Control involves the introduction of multimedia content in the content DE. It is necessary to neutralize emotions, the influence of emotions on the nature of the learning process [Rabinovich, Muezzinoglu, 2010].

It is proposed to use active tools A networks to solve problems [Shamis, 2006]. These tools allow you to form an emotional student model and to calculate the value of the emotional immersion of the student. This value is linked to the delay time switching scenario presentation.

Option model considering the influence of emotions on the formation of individual trajectory of student learning based on neural network technology is shown in figure 1.

The model of active neural network is constructed using the architecture in its ongoing internal challenges, consisting in the need to continuously maintain optimal functional States of neurons and require their constant activity. An internal challenge, first, occurs as a result of the deviation from the optimum functional status is not working (not excited) neuron due to the occurrence of student emotions. This task is solved, for group *vzaimootnosheniio*, due to the fact that emotions change. Internal problem arises when setting the external task of perception or behavior, reducing, ultimately, to external excitation of a group of neurons in the network. External task, in the form of emotions that disturbs the regime of the General uniform optimization leads to deviations from the optimum functional States of neurons.

The analysis of the literature [Shamis, 2006], [Kruglov, 2001], [Rutkowski, 2006] allowed us to propose for the implementation of the scheme of active neuronal network with recurrent inhibition scheme on the basis of the classical three-layer perceptron.

For modelling procedures pre-synaptic inhibition is proposed to use artificial neuroglial networks (INGS), in which glial cells are as much a part of the network, as the neurons [<http://aideus.ru>], that is illustrated in the following figure 2.

Certain types of glial cells, the astrocytes, are active participants in the diffuse transmission of information in the brain. Astrocytes are capable of generating calcium oscillations and form networks through gap contacts [Bezzi & Volterra, 2001]. Astrocytes affect synaptic communication of neurons.

Balanced work synaptic excitation/inhibition, and neuromodulatory systems is the basis for the transfer, processing and saving information in the brain, as well as the generation of brain rhythms, which are a kind of clock operating frequencies of structures [Vinogradova, 2001]. These same mechanisms underlying the formation of emotional brain patterns.

The flow of emotional information through glia leads to modifications of the weights of links between neurons, which leads to the expected lock training. In particular, by introducing a specially selected system of glial cells, in multilayer perceptron, it can be extended so that it is studied by the method of back propagation without any external algorithm.

Thus, the signal flow through some synapses does not affect the activity of neurons and conduction of other synapses. This means that the "weight of contacts" between neurons can vary not only slow in learning, but also dynamically, under the influence of current information emotion.

Of neurons in first layer neurons in second layer are formed only excitatory connections. Glial cells from the first layer to the glial cells of the second layer are formed only inhibitory connections. In physiology such scheme of interaction of structures called neurons with

recurrent inhibition [Shamis, 2006].

The lack of emotion hindering relationship, going to the neuron of the first layer from the second layer element, does not differ from bonds within the first layer, and formed it potential is included in the algebraic sum of the potentials of a neuron, which corresponds to a stable learning process. In the presence of emotion hindering relationship, coming to the neuron from the first layer element in the second layer, completely prohibits the firing of a neuron at the time it is created reduce the brake capacity to a certain value. All this time there is an independent summation of all other created on the neuron potential.

Emotions and thinking are active processes that appear as a result of changes in the organization of and power relations between the different centers of the brain continuously in time, i.e. the dynamic response of the brain to information about the environment and inner world. Dynamic variables describing the dynamics of emotions and cognitive functions and their interactions form a joint working or the phase space [Rabinovich, Muezzinoglu, 2010].

Based on these considerations, a mathematical model of a neural network can be represented as a nonlinear system of differential equations obtained on the basis of experimental studies of biophysical processes in neurons and astrocytes (glial cells).

Network of interacting cells are described by multidimensional systems of differential equations [Prokin, Simonov, Kazantsev, 2012]. Dynamics of neuronal interactions and astrocytes is proposed to model the equations of the Lee-Rences. As a model a single neuron, it is proposed to use the model of Hodgkin-Huxley [Hodgkin & Huxley, 1952].

For modeling it is proposed to use a carefully tested, documented simulation-type neural network NEURON simulator [Prokin, Simonov, Kazantsev, 2012]

Conclusion

Intelligent information system of educational process organization, on the basis of active neuronal network with recurrent inhibition that takes into account the impact of student emotions on the formation of individual trajectory of studying, a student of some subject area knowledge. Network maintains, analyzes and combines the results of control of emotions, formed on the basis of the response frequency of the hyperlinks. The evaluation of results is generated multimedia content to the contents of didactic units of knowledge module, to neutralize emotions.

This article proposes a new method of control of an individual trajectory of studying some subject area knowledge through active neural networks with recurrent inhibition, describing the state and behavior of the student during the study of the didactic units of the module of knowledge.

To simulate such a network we propose a mathematical model based on systems of differential equations, the model neuron of Hodgkin-Huxley and simulation environment based on the simulation of neural networks.



УДК 621.9

ОЦЕНКА ЭФФЕКТИВНОСТИ ПРОЦЕССА ОБУЧЕНИЯ САМ-ОБРАБОТКЕ СТУДЕНТОВ МАШИНОСТРОИТЕЛЬНОГО ПРОФИЛЯ

Барabanова Е.Ю.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

puzakina@yandex.ru

В данной работе предложено решение вопроса по оценке эффективности процесса обучения САМ-обработке на основе выведенного коэффициента эффективности. При построении процедуры оценки использовались достижения современного интеллектуального анализа временных рядов, в частности, метод меры ассоциации между временными рядами. Помимо этого, рассмотрены особенности самой разработки рабочей учебной программы для обучения студентов машиностроительного направления на основе онтологии предметной области.

Ключевые слова: эффективность обучения, САМ-обработка, станки с ЧПУ.

Введение

В настоящий время наблюдается рост интереса со стороны промышленности к инженерной сфере деятельности вообще и к сфере автоматизированного проектирования в частности. В области последнего на большинстве производств имеет место острый дефицит в квалифицированных специалистах в области САМ (Computer-aided Manufacturing). Под термином Computer-aided manufacturing (САМ) понимают как программы, используемые технологами-программистами для подготовки управляющих программ (УП) для станков с числовым программным управлением (ЧПУ), так и сам процесс автоматизированной подготовки производства.

Сама проблема недостаточного количества квалифицированных кадров обсуждается на разных уровнях и в разных странах. Так, к примеру, Джеймс Вэйкфорд в своей статье "Управление цепочкой поставки" анализирует первопричины сложившейся негативной ситуации и предлагает в качестве одного из решений введение нового термина - "Цепочка поставки САМ-специалистов". Возьмем за основу данную терминологию и пристально рассмотрим сторону эффективности самого процесса обучения САМ-обработке.

1. Освещенность вопроса оценки эффективности обучения

Вопрос оценки эффективности обучения персонала широко освещен в научной литературе.

Так, в частности, специалисты выделяют четыре основных способа качественной оценки результатов профессионального обучения.

При первом способе происходит оценка способностей и знаний сотрудников непосредственно в ходе или по завершению курса обучения. При использовании второго способа оцениваются профессиональные знания и навыки конкретно в условиях производства. Третий способ – оценивается влияние обучения на параметры производства. Четвертый способ – это способ экономической оценки.

Однако описанные выше способы оценки носят слишком общий характер и не учитывают специфику узконаправленной предметной области – машиностроения. Решение в этом направлении предложил В.А. Селезнев в своей статье «Комплексные показатели оценки эффективности освоения конструкторско-технологических компьютерных систем». В ней он предлагает интегральные коэффициенты оценки эффективности освоения САД/САМ систем, а также компьютерная программа для их расчетов и обработки общих результатов по учебной группе. Особый интерес представляет выведенный автором интегральный коэффициент эффективности освоения технологического модуля компьютерной программы, который определяется по формуле

$$KE = \frac{T_{\text{вып}} + N_{i\phi}}{(N_{\text{н}})^2}, \quad (1)$$

где $T_{\text{вып}}$ - время выполнения технологической

разработки в мин, $N_{\text{ош}}$ - количество ошибок в выполненной разработке, а $N_{\text{оп}}$ - количество технологических переходов (например - установить, точить, сверлить, фрезеровать, контролировать и т.п.) в разработке, возведенное в квадрат.

Использование вышеописанного подхода возможно лишь на начальных этапах обучения, т.к. оно излишне упрощает и сам процесс САМ-проектирование и сам процесс обучения САМ-проектированию. Также представленный подход не позволяет отследить общую динамику и прогресс обучающихся и выявить взаимосвязь между временем работы, количеством неточностей и ошибок и ростом квалификации. Однако автор и не ставил перед такую задачу, он работал с общим результатами по группе обучаемых.

2. Методика оценки эффективности САМ-обучения на базе САП-преобразований

Отдельно стоит отметить, что эффективность обучения носит динамический характер, а значит, его представляется возможным формально рассмотреть как наблюдаемую величину - сигнал, а в качестве метода исследования следует воспользоваться реконструкцией динамических систем. Этот раздел теории динамических систем называется анализом временных рядов. В качестве аналога для сравнения результатов, снимаемых с наблюдаемой величины, между собой и поиска взаимосвязей были приняты преобразования Скользящих Аппроксимаций, САП-трансформ (Moving Approximation Transform, MAP). Их особенность - инвариантность относительно линейных преобразований временных рядов, а также ненужность сглаживания данных, поскольку САП-трансформ основан на сглаживании временных рядов. Наиболее полную информацию об ассоциациях между временными рядами y , x дает последовательность ассоциаций локальных трендов, посчитанных для всех размеров скользящих окон $k=2, \dots, n$. Такая последовательность называется ассоциативной функцией, значения которой зависят от размера. График этой функции дает полезную информацию об ассоциациях между временными рядами. Можно рассматривать некоторое подмножество $K \{2, \dots, n\}$ всех возможных окон и говорить о функции $AFK(y, x)$, определенной на множестве окон K . Среднее или максимальное значение этой функции может быть использовано как мера ассоциации между временными рядами. Ассоциативная функция позволяет выявлять положительные и отрицательные ассоциации между временными рядами, имеющими сложную структуру.

Простой двухфакторный анализ эффективности процесса обучения САМ-обработке можно осуществить при продолжительном ведении преподавателем отчета по обучаемым студентам, в который он заносит следующую информацию:

- Класс сложности задания (Ксл)
- Время выполнения задания (Твып)
- Количество операций в САМ-обработке ($N_{\text{оп}}$)
- Количество неточностей и ошибок при выполнении задания (оно же качество выполнения типового задания) ($N_{\text{ош}}$)

Мера же ассоциации по методу САП-трансформ будет определяться между Отношением $(Твып + N_{\text{ош}})/(N_{\text{оп}})^2$ и классом сложности задания Ксл.

В своем исследовании мы, во-первых, выстраиваем научный подход к самому процессу обучения САМ-обработке и, во-вторых, изучаем зависимость между временем, затраченного студентом на создание САМ-обработки, и его конечной квалификацией.

3. Построение программы обучения САМ-обработке на основе экспертных знаний

В сфере профессионального обучения студентов машиностроительного профиля отсутствует как таковой целый пласт обучающих методик по прикладной специализированной тематике - программирование станков с ЧПУ - которые бы одновременно с этим отвечали современным требованиям крупных производств. Т.к. данное направление требует обширных фундаментальных знаний в области теории металлообработки, то его изучение целесообразно включать на старших курсах средне-профессиональных и высших учебных заведений. Однако оно не должно ограничиваться освещением элементарных основ: понятий G-кодов, M-функций, классификации и устройств станков с ЧПУ, тем более что в качестве примеров последних даются, зачастую, примеры 70-х и 80-х годов.

При составлении учебных программ и методик стоит применять последние достижения области педагогики и психологии высшей школы. Но если в наиболее востребованных специальностях (IT-сфера, PR, экономика, менеджмент) это ярко выражено, то в области высшего образования по технологии машиностроения таких наработок нет вовсе.

Однако создание современной учебной программы не способно в полной мере адекватно решить проблему подготовки квалифицированных кадров. Для этого необходимо сформировать модель обучения с ее качественным анализом, выражающемся в измерении конкретных параметров и построенных на их основе временных рядах.

В нашем случае под моделью обучения мы понимаем динамический процесс по получению необходимых квалификационных навыков, необходимых для автоматизированного создания корректных управляющих программ для станков с ЧПУ. Квалифицированность навыков заключается в

самостоятельном выполнении обучающих задач, формируемых преподавателем перед своими студентами. Под корректностью управляющих программ стоит понимать отсутствие столкновений режущего инструмента с заготовкой и элементами специального технологического оснащения; минимизация смен позиций режущего инструмента; безопасность всех перемещений режущего инструмента на «быстром ходе» (при максимально возможных подачах).

Задача создания такого обучающего курса состоит также в подборе метрики учебных заданий с учетом сложности САМ-обработки, которые в полной мере охватывают все типовые детали по механообработке. Сам фрагмент метрики САМ-обработки представлен в таблице 1.

Таблица 1 – Название таблицы

Наименование конструктивного элемента	Характеристика элемента, которая может повлиять	Условие влияния, усложняющего механообработку
Паз (имеет постоянную ширину)	ширина	Если соотношение глубина/ширина ≥ 2 ,
	глубина	
	скругление	Если соотношение ширина/величина скругления ≥ 3 , то "подчистка углов"
		Паз = "поднутрение", то или отдельный установ или специнструмент (грибковая фреза)
Карман колодец (внешний ограничивающий контур всегда замкнут и дно расположено ниже плоскости привязки)	Толщина стенок пола	Если толщина стенок пола $< 2,5$, то рисуем отдельные ребра жесткости, которые потом срезаем
	Радиус углов стенок	Если соотношение ширина/величина радиуса ≥ 3
Скос (наклонная поверхность)	Угол	Угол между скосом и верхней поверхностью нестандартный, то или использовать обработку 3+2, или раскатывать поверхность сферической фрезой

Эти задачи с ростом сложности мотивируют обучающихся на более широкий и неординарный подход к решению их. Финальным этапом в обучении становится следующая ролевая игра. Обучающиеся делятся на группы по 2 человека, один из которых принимает на себя роль инженера-конструктора, спроектировавшего деталь для механообработки на станке с ЧПУ; второй принимает роль инженера-технолога, которому необходимо составить УП на данную деталь. В ходе работы они должны взаимодействовать между собой, решая задачи по упрощению конструкции детали и совместному проектированию УП. Со стороны преподавателя-модератора после выполнения задания выставляется оценка работе каждой группы, которая в общей картине профессиональной квалификации имеет больший вес.

Укрупнено процесс написания управляющих программ можно представить следующим образом. Технологи-программисты при получении задания на разработку в первую очередь смотрят на конструкторский документ – чертеж (электронную модель детали/сборочной единицы), и выявляют следующие ключевые моменты:

- габарит детали/сборочной единицы;
- материал детали/сборочной единицы;
- общая геометрия детали/сборочной единицы;
- размерная цепочка с предельными отклонениями от номинальных данных;

Далее назначается нулевая точка и составляется общая стратегия САМ-обработки. В последнюю очередь, после проверки корректности всей обработки, управляющую программу проводят через постпроцессор и отдают в цех оператору.

Более наглядно описываемый процесс представлен на рисунке 1.

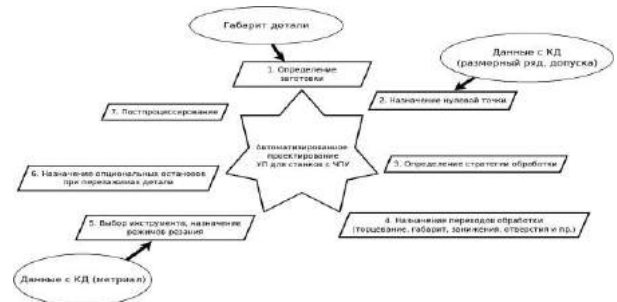


Рисунок 1 – Схема процесса автоматизированного проектирования управляющих программ для станков с ЧПУ

Описанный выше подход нашел свою реализацию в разработанных нами методических указаниях к практическим и лабораторным работам для студентов машиностроительного профиля.

4. Практическая реализация при исследовании эффективности обучения САМ-обработке

В рамках проведения практической работы среди студентов авиационного колледжа указанные данные были получены первоначально из отчета преподавателя, отмечавшего количество разработанных УП студентами и дававшего свою экспертную характеристику сложности их работ по 10-бальной шкале. Отдельно стоит отметить, что при обучении студентов была применена онтология предметной области по станкам с ЧПУ, а также использовались учебный справочник технического переводчика «Станки с ЧПУ» и база знаний технического переводчика «Юнитех».

Статистические данные были предоставлены за период декабрь 2014 г - февраль 2015 г в формате excel-отчета. После просчета локальных трендов по каждому из шести результатов анализа были построены ассоциативные функции, показанные ниже.



Рисунок 2 – Ассоциативные функции для данных по количеству УП и степени их сложности

Заключение

Таким образом, предложенный в данной работе подход при оценке эффективности процесса обучения САМ-обработке студентов машиностроительного профиля базируется на собственных разработанных методических указаниях и учитывает динамических характер самого процесса обучения. В ходе разработки методики обучения САМ-обработке была разработана метрика сложности самой САМ-обработки, а также систематизирован подход к ней.

Также в данной работе представлены практические результаты оценки эффективности обучения написанию управляющих программ для станков с ЧПУ с применением метода скользящих аппроксимаций и последующим построением ассоциативной функции.

Библиографический список

[Селезнев, 2012] Селезнев В.А. Комплексные показатели оценки эффективности конструкторско-технологических компьютерных систем. // Вестник Брянского государственного университета. Брянск. 2012. – С.67-73.

[James Wakeford, 2014] Managing the CAM Workforce Supply Chain // CAD/CAM/CAE OBSERVER, 2014. – P. 62-65.

[Барабанова и др, 2015] Барабанова Е.Ю., Башаев В.А., Клейн В.В., Мокин В.С. Построение информационной поддержки автоматизированного проектирования управляющих программ для станков с ЧПУ // Радиотехника. Москва. 2015. – С.63-67.

EVALUATING THE EFFECTIVENESS OF THE LEARNING PROCESS OF STUDENTS CAM-PROCESSING MACHINE BUILDING

Barabanova E. Yu.

Ulyanovsk State Technical University, Russian Federation

puzakina@yandex.ru

This paper proposes a solution to the problem of assessing the effectiveness of the learning process CAM-processing based on the derived coefficient of efficiency. In constructing the assessment procedures used to achieve the modern intelligent time-series analysis, in particular, the method measures the association between time series. In addition, the peculiarities of the development of the working of the curriculum for training students of engineering directions based on ontology.

Introduction

Currently, there is a growing interest from the industry in the field of engineering activities in general and to the field of computer-aided design in particular. In the latter in most industries there is an acute shortage of qualified specialists in the field of CAM (Computer-aided Manufacturing). The term Computer-aided manufacturing (CAM) is understood as a program, the technology used by programmers, to prepare control programs (NC) machine tools with numerical control (CNC).

Main Part

A review of existing approaches to learning efficiency. Disclosed is a question of methodology for assessing the effectiveness of CAM-teaching with an explanation of all the features of the development of training programs CAM-treatment of students of engineering profile. We describe the practical results in the evaluation of the effectiveness of student learning aviation college.

Conclusion

The proposed approach in this study to assess the effectiveness of the learning process of students CAM-processing engineering profile based on its own developed guidelines During the development of methods of teaching CAM-processing was developed metric complexity of the CAM-processing, as well as a systematic approach to it.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКАЯ МОДЕЛЬ ПОЛЬЗОВАТЕЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СПРАВОЧНОЙ СИСТЕМЫ ПО ПРОЕКТИРОВАНИЮ ПРОГРАММ

Пивоварчик О.В.

*Барановичский государственный университет,
г. Барановичи, Республика Беларусь*

pivovarchyk@gmail.com

В работе проведен анализ существующих моделей пользователя и перечислены способы адаптации компьютерных систем к пользователю. Предложена семантическая модель пользователя справочной системы по проектированию программ, ориентированных на обработку баз знаний. Справочная система строится на базе технологии OSTIS.

Ключевые слова: компьютерная система, управляемая знаниями; справочная система; интеллектуальная система консультационного обслуживания и обучения; модель пользователя; база знаний.

Введение

Одним из критериев интеллектуальности компьютерной системы является возможность адаптироваться к индивидуальным особенностям пользователя. Для реализации адаптационных функций необходимо наличие в базе знаний информации, которая представляет собой совокупность параметров, характеризующих пользователя и оказывающих наибольшее влияние на процесс его взаимодействия с системой, а также соответствующих механизмов адаптации. Представление знаний о пользователе, эффективное извлечение информации, связанной с пользователем, и использование этой информации для согласованной и выразительной адаптации является критическим фактором для успеха адаптивных систем [Sosnovsky et al, 2010].

Применение адаптации в справочных и обучающих системах позволяет повысить их эффективность и улучшить процесс взаимодействия с пользователем. Основная идея адаптации в системах данного класса основана на предположении, что различие в некоторых характеристиках пользователя влияет на порядок представления учебного материала и полезность информационной обеспеченности.

Целью работы является формирование комплексной модели пользователя для справочных и обучающих систем, используемых разработчиками программ, ориентированных на обработку баз знаний. В статье перечислены адаптивные

технологии, применяемые в интеллектуальных обучающих системах, представлена классификация и основные подходы к разработке модели пользователя, представлена семантическая модель пользователя. Кроме этого, обсуждаются проблемы формирования структуры модели пользователя для различных областей, а также вопросы получения значений параметров, входящих в ее состав.

Интеллектуальная система проектируется на базе *Технологии OSTIS* (Open Semantic Technology for Intelligent Systems) [Голенков и др., 2015]. Формальные модели интеллектуальных систем, проектируемых на базе OSTIS, строятся на основе семантических сетей, что создает необходимые условия для обеспечения семантической совместимости интеллектуальных систем и их компонентов [Голенков и др., 2013].

1. Способы адаптации в интеллектуальных обучающих системах

Среди множества адаптивных систем, адаптивные образовательные системы имеют самую большую историю исследований [Sosnovsky et al, 2010] и являются наиболее разнообразными и многочисленными. Существуют различные виды способов адаптации, применяемые в системах данного класса. В работах [Weber, 1999], [Брусиловский, 2002] выделены следующие способы:

- построение последовательности курса обучения;

- интеллектуальный анализ ответов пользователя;
- интерактивная поддержка в решении задач;
- помощь в решении задач, основанная на примерах;
- адаптивное представление учебного материала;
- адаптивная поддержка в навигации по учебному пространству;
- адаптивная поддержка сотрудничества.

2. Модель пользователя как комплексный критерий адаптации

Первые модели пользователя в компьютерных системах появились в 1970-х годах в работах J. Allen, P. Cohen, C. Perrault и E. Rich [Kobsa, 2007]. В настоящее время большинство систем в той или иной мере использует информацию о пользователе. Современные исследования, проводимые в данной области, можно разделить на три класса: определение структуры и основных характеристик пользователя, которые требуется включать в систему для адаптации; способы построения модели пользователя; методы и средства представления характеристик пользователя.

2.1. Структура и основные характеристики модели пользователя

Существует множество схем, которые содержат большое количество характеристик для использования в составе модели пользователя [Голенков и др., 2001]. В работах [Gavrilova et al, 1997], [Голенков и др., 2001], [Brusilovsky et al, 2007] выделены основные группы характеристик пользователя, используемые для применения в адаптивных интеллектуальных системах. Приведем обобщенные группы:

- демографические показатели;
- индивидуальные психологические характеристики;
- психомоторные характеристики;
- когнитивные показатели;
- мотивация;
- подготовленность к работе с системой;
- характеристики взаимодействия с системой.

При разработке каждой конкретной системы состав характеристик пользователя изменяется. В интеллектуальных справочных и обучающих системах модель пользователя традиционно включает:

- когнитивные и психологические характеристики;
- группы значений, характеризующих цели и задачи, подготовленность;
- значения параметров, описывающих процесс взаимодействия с системой.

2.2. Способы построения модели пользователя

На основании анализа работ [Brusilovsky, 2007], [Sosnovsky et al, 2010], [Валькман и др., 2011] можно выделить следующую классификацию существующих моделей пользователя:

- По способу представления знаний:
 - *декларативные* – содержат информацию о знаниях пользователя. Разбиваются на следующие подклассы:
 - стереотипные;
 - оверлейные;
 - генетический граф;
 - на основании байесовских сетей.
 - *процедурные* – содержат информацию о знаниях предметной области и механизмы решения задач. Разбиваются на следующие подклассы:
 - пертурбационные;
 - модели ограничений.
- По способу извлечения знаний:
 - *явные* – система явным образом запрашивает информацию, необходимую для создания и обновления модели;
 - *неявные* – система неявно формирует модель в процессе наблюдений за взаимодействием пользователя с системой.
- По способу построения:
 - построение модели в процессе *тестирования пользователя*;
 - формирование в процессе *наблюдений за взаимодействием пользователя с системой*.
- По степени специализации:
 - *общие* – система предполагает одну общую модель для однородной группы пользователей системы;
 - *индивидуальные* – используются индивидуальные особенности каждого пользователя.
- По времени существования:
 - *долгосрочные* – сохраняют информацию на протяжении жизненного цикла системы;
 - *краткосрочные* – сохраняют информацию в пределах одного сеанса работы системы.

Классическими подходами к построению модели пользователя являются *оверлейная* и *стереотипная* модели.

Стереотипная модель применяется в системах, где возможно идентифицировать типичные категории пользователей, которые используют систему по одному пути, с одинаковыми результатами и которые можно описать одним множеством характеристик. Каждая такая категория называется стереотипом и описывается с помощью профиля стереотипа. Пользователь характеризуется

определенным стереотипом или комбинацией нескольких ортогональных стереотипов и обновляется в соответствии с профилем стереотипа. Часто стереотипы формируют иерархию, где стереотип более высокого уровня наследует свойства стереотипа более низкого уровня. Стереотипные модели используются в системах для быстрой, но менее точной оценки пользователя. Популярным путем моделирования пользователей на основе стереотипов является линейное множество категорий для представления типичного уровня умений пользователя.

Более точной моделью пользователя является оверлейная модель. Главная идея данного подхода – представление предметной области в виде элементарных компонентов и использование их для представления знаний или интересов и предпочтений пользователя. В работах встречаются различные названия компонентов: элементы знаний, результаты знаний, концепты, понятия и др. Понятия являются атомарными элементами декларативного представления предметной области, семантически полными и взаимосвязанными.

Простейшим вариантом оверлейной модели пользователя является *векторная модель* [Валькман и др., 2011]. Векторная модель пользователя использует модель предметной области или модель эксперта, сформированную на основании модели предметной области, как эталон знаний и состоит из множества пар понятие-значение, где значение представляет оценку знаний пользователя связанного понятия. Каждому изучаемому понятию ставится в соответствие элемент, принимающий значение «знает/не знает» или значения по определенной дискретной шкале. Значение может быть проинтерпретировано как оценка или как вероятность. Более гибким вариантом является *сетевая модель*. В отличие от предыдущей модели, в ней предметная область представлена в виде понятий и отношений между ними.

К преимуществам оверлейной модели можно отнести точность и гибкость. Выделение элементарных понятий позволяет системам настраивать свои действия на очень детальный уровень представления пользователя. Также она позволяет отражать точную динамику эволюции характеристик пользователя (очень важно в обучающих системах). Недостатком модели является сложность реализации для некоторых предметных областей.

Генетический граф является расширением сетевой модели. В дополнение к ней он содержит различные обобщения, конкретизации, аналогии, уточнения, отклонения от правил, применяемых экспертом [Валькман и др., 2011]. На основании этих данных модель позволяет описать возможные пути развития пользователя.

Байесовские сети очень популярный формализм для представления различных аспектов моделирования пользователя [Sosnovsky et al,

2010]. Большинство систем используют их, чтобы моделировать отношения между разными компонентами модели, например, эмоциями, целями, знаниями.

В *пертурбационных моделях* фиксируется множество ошибок и ошибочных элементарных операций пользователя. Основой пертурбационных моделей является идентификация причин расхождения знаний эксперта и пользователя. Среди таких причин могут быть: недостаток знаний у пользователя, неправильное применение знаний, ошибки невнимательности или умышленные ошибки [Валькман и др., 2011].

Модели пользователя, *основанные на ограничениях*, успешно используются для моделирования знаний пользователей в интеллектуальных обучающих системах. В данном подходе каждое ограничение представляет подходящее множество состояний эквивалентных возможным проблемам, ошибкам пользователя. Нарушение ограничения говорит об ошибке.

В современных исследованиях популярным способом является построение модели пользователя с точки зрения их взаимоотношений с другими пользователями. Модель сотрудничества пользователя реализуется на основе вектора оценок знаний других пользователей, что повышает мотивацию пользователя к обучению и заинтересованность системой.

Система адаптируется к пользователю по определенному набору характеристик, представленных одним из вышеперечисленных способов. Например, адаптивное представление и реализация простой последовательности курса могут быть основаны на стереотипной модели пользователя. Поддержка адаптивной навигации и динамические виды последовательности курса используют как минимум оверлейную модель пользователя. Самые сложные адаптационные техники: интеллектуальный анализ ответов, интерактивная поддержка в решении задач, помощь в решении задач, основанная на примерах – требуют более развитых средств и методов представления модели пользователя.

3. SC-модель пользователя интеллектуальной системы

Консультационная система обслуживания и обучения разработчиков программ является *компьютерной системой, управляемой знаниями*, построенной по *Технологии OSTIS* [Голенков, 2015] – *ostis-системой*. Она обладает функциональностью справочных и обучающих систем. SC-модель системы представлена в работе [Пивоварчик, 2015]. В статье рассмотрим один из компонентов системы – модель пользователя.

Пользователем интеллектуальной системы консультационного обслуживания и обучения может быть *физическое лицо* или другая *компьютерная*

система. По отношению к разрабатываемой системе они являются *внешними субъектами ostis-системы, с которыми осуществляется взаимодействие*. Типология субъектов деятельности представлена в [Шункевич и др., 2016]. В данной работе в качестве конечного пользователя *ostis-систем* данного класса рассматривается только физическое лицо. Далее планируется развитие модели пользователя для хранения информации о других видах субъектов и для взаимодействия с любыми техническими системами.

Модель пользователя представляет собой компонент *OSTIS*, следовательно, она является компьютерной системой, управляемой знаниями. В соответствии с общей *sc-моделью компьютерной системы* [Давыденко и др., 2016] представим *sc-модель пользователя*:

sc-модель конечного пользователя компьютерной системы консультационного обслуживания и обучения

= *sc-модель конечного пользователя ostis-системы консультационного обслуживания и обучения*

= *sc-модель физического лица*

<= базовая декомпозиция*:

- {
- *sc-модель базы знаний пользователя*
- *sc-модель машины обработки знаний пользователя*
- *sc-модель интерфейса модели пользователя*
- }

Структура *sc-модели базы знаний пользователя* строится в соответствии с универсальной структурой, представленной в работе [Давыденко, 2016]. Рассмотрим *предметную часть базы знаний*, которая содержит всю информацию о *Предметной области конечного пользователя компьютерной системы консультационного обслуживания и обучения*.

Предметная область конечного пользователя компьютерной системы консультационного обслуживания и обучения

= *Предметная область характеристик конечного пользователя ostis-системы консультационного обслуживания и обучения*

= *Предметная область характеристик физического лица*

=> частная предметная область*:

- *Предметная область характеристик, не зависящих от учебного материала*
=> частная предметная область*:
- *Предметная область персональных данных пользователя*
- *Предметная область демографических показателей пользователя*
- *Предметная область когнитивных показателей пользователя*
- *Предметная область психомоторных характеристик пользователя*
- *Предметная область индивидуальных психологических характеристик*

- *Предметная область лингвистических характеристик пользователя*
- *Предметная область характеристик, зависящих от учебного материала*
=> частная предметная область*:
- *Предметная область знаний пользователя*
- *Предметная область умений пользователя*
- *Предметная область интересов и предпочтений пользователя*
- *Предметная область целей и задач пользователя*
- *Предметная область характеристик взаимодействия пользователя с системой*
- *Предметная область деятельности пользователя*
- *Предметная область ситуаций деятельности и способы их решения, в том числе типовые задачи и функции, затруднения и типичные ошибки*
- *Предметная область квалификации пользователя*

Раздел *характеристики, не зависящие от учебного материала*, содержит информацию, характеризующую пользователя как личность. Эта информация либо не изменяется вообще, либо изменяется в большом промежутке времени. Заполнение данного раздела знаниями о конкретном пользователе производится частично посредством диалога с пользователем, а частично посредством специализированных психологических тестов. Данный компонент может использоваться различными компьютерными системами, в том числе, не являющимися консультационными и/или обучающими.

Набор используемых характеристик определяется разработчиками каждой конкретной системы. Для системы консультационного обслуживания и обучения разработчиков программ учитываются *когнитивные показатели, психомоторные характеристики, индивидуальные психологические характеристики*. Совокупность этих качеств обеспечивает более успешное выполнение задач, возникающих в обучающей среде, а также самообучение и саморазвитие.

Раздел *характеристики, зависящие от учебного материала*, содержит информацию, отражающую степень владения знаниями из соответствующих предметных областей. Заполнение данного раздела знаниями о конкретном пользователе производится либо на основании тестирования, либо неявно в процессе взаимодействия с системой. Для представления знаний данного раздела используется *оверлейная модель*, которая описывает знания как подмножество *эталонных моделей предметных областей* в рамках обучающего курса.

Раздел *квалификация пользователя* содержит информацию о степени подготовленности пользователя. Раздел может использоваться

системами разных классов. Данный раздел содержит информацию:

- О квалификации пользователя до начала работы с системой. Заполняется пользователем. Является статическим.
- О квалификации пользователя, приобретенной в процессе работы с системой. Формируется на основании раздела характеристики, зависящие от учебного материала, и отражает результат обучения. Является динамическим.

Описание объектов в рамках предметных областей отражается различными типами онтологий [Шункевич и др., 2015]. Рассмотрим на примере *Предметная область знаний пользователя*.

Предметная область знаний пользователя

∈ предметная область и ее онтологии

⇐ декомпозиция раздела*:

- {
- *Предметная область знаний пользователя*
- *Структурная спецификация предметной области знаний пользователя*
- *Терминологическая онтология предметной области знаний пользователя*
- *Теоретико-множественная онтология предметной области знаний пользователя*
- *Теоретико-множественная онтология предметной области знаний пользователя*
- *Логическая онтология предметной области знаний пользователя*
- *Логическая иерархия высказываний о предметной области знаний пользователя*
- *Онтология задач и решений задач предметной области знаний пользователя*
- *Онтология классов задач и способов решения задач в предметной области знаний пользователя*
- }

Рассмотрим фрагмент базы знаний *Структурная спецификация предметной области знаний пользователя*.

Предметная область знаний пользователя

⇒ sc-онтология*:

Структурная спецификация предметной области знаний пользователя

= [

Предметная область знаний пользователя

⇒ не максимальный класс объектов исследования':

• sc-элемент знаний предметной области
⇒ исследуемое отношение':

- *изученный учебный материал**
- *изучаемый учебный материал**
- *известный учебный материал**
- *выведенный учебный материал**
- *решенная задача**
- *решаемая задача**

]

Рассмотрим пример реализации адаптивной

технологии построение последовательности курса обучения на основании фрагмента базы знаний *Предметная область знаний пользователя*. Sc-элемент знаний предметной области может связываться с пользователем одним из следующих отношений:

- *изученный учебный материал**, если пользователь успешно выполнил достаточное количество тестов или заданий или изучил достаточно примеров;
- *изучаемый учебный материал**, если пользователь начал изучение, но не выполнял тесты или задания;
- *изучен весь учебный материал в составе**, если в первое множество входят все его подразделы (в случае, если элементом знаний является раздел) или ключевые элементы раздела или подраздела;
- *известный учебный материал**, если пользователь отметил, что знает элемент знаний, но элемент не входит в первое или третье множество.

Ассиметричное отношение нефиксированной арности *сценарий обучения** связывает пользователя с последовательностью обучения. Кортеж отношения связывает пользователя, обучающий курс, статический сценарий и/или динамический сценарий обучения под атрибутами *статический'*, *динамический'* [Пивоварчик, 2015]. Последовательность состоит из разделов и подразделов обучающего курса, формируется согласно целям обучения и доступным режимам взаимодействия, содержит рекомендации на основании уровня знаний пользователя. Примеры рекомендаций для разделов/подразделов в зависимости от состояния их или входящих в них других sc-элементов знаний предметной области. Они включаются в последовательность с установкой ролевых отношений:

- *предлагается для изучения'*, если элемент не входит ни в одно из предыдущих множеств, но все предварительные условия для изучения выполнены;
- *готов для изучения'*, если элемент не входит в первое, второе и третье множество, но все предварительные условия для изучения выполнены;
- *не готов для изучения'*, если элемент не входит в первое, второе и третье множество и не выполнены все предварительные условия для изучения.

SC-модель машины обработки знаний пользователя состоит из множества агентов, обеспечивающих формирование, модификацию, мониторинг и анализ деятельности пользователя на предмет соответствия действия или множества действий предыдущим ошибкам, типичным ошибкам пользователей и др., а также на предмет возникновения некоторых критических ситуаций.

SC-модель интерфейса модели пользователя включает компоненты, которые обеспечивают взаимодействие с пользователями, а также обмен знаниями о пользователе с другими компонентами.

Заклучение

Представленная в статье модель пользователя разрабатывается как компонент компьютерной системы, управляемой знаниями, построенной на основе Технологии OSTIS, что позволяет ее легко интегрировать в другие смежные системы. Включение компонента в системы обучения позволяет автоматически настраивать характер адаптации к учебному материалу и не требует значительных затрат времени на реализацию адаптивных параметров интерфейса.

Предложенная модель содержит большое количество характеристик, позволяет вносить изменения в них и предоставляет возможность задействовать только необходимые.

Библиографический список

[Sosnovsky et al., 2010] Sosnovsky, S. Ontological technologies for user modeling / S. Sosnovsky, D. Dicheva // International J. of Metadata, Semantics and Ontologies. – 2010. – Vol. 5, № 1. – P. 32-71.

[Голенков и др., 2015] Голенков, В.В. Семантическая технология компонентного проектирования систем, управляемых знаниями / В.В. Голенков, Н.А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS 2015) : материалы V Междунар. научно-техн. конф., Минск, 19-21 февр. 2015 г. / БГУИР ; редкол.: В.В. Голенков [и др.]. – Минск, 2015. – С. 57-78.

[Голенков и др., 2013] Голенков, В. В. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н. А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS 2013) : материалы III Междунар. научно-техн. конф., Минск, 21-23 февр. 2015 г. / БГУИР ; редкол.: В.В. Голенков [и др.]. – Минск, 2013. – С. 55-78.

[Weber, 1999] Weber, G. Adaptive Learning Systems in the World Wide Web / G. Weber // UM99 – User Modeling ; Proceedings of the Seventh International Conference, Banff, Canada, June 20 - 24, 1999 / CISM International Centre for Mechanical Sciences ; ed. J. Kay. – Banff, 1999. – P. 371-378.

[Брусиловский, 2002] Брусиловский, П.Л. Адаптивные и интеллектуальные технологии в сетевом обучении / П.Л. Брусиловский // Новости искусственного интеллекта. – 2002. – № 5. – С. 25-31.

[Kobsa, 2007] Kobsa, A. Generic User Modeling Systems / A. Kobsa // The Adaptive Web : methods and strategies of web personalization / Eds P. Brusilovsky, A. Kobsa, W. Nejdl. — Berlin : Springer-Verlag Berlin Heidelberg. — P. 136-154.

[Голенков и др., 2001] Голенков, В.В. Интеллектуальные обучающие системы и виртуальные учебные организации. Монография / В.В. Голенков [и др.] ; под ред. В.В. Голенкова. – Минск : БГУИР, 2001. – 488 с.

[Gavrilova et al., 1997] Gavrilova, T. An Approach to Mapping of User Model to Corresponding Interface Parameters / T. Gavrilova, A. Voinov // UM97 – User Modeling ; Proceedings of the Workshop «Embedding User Models in Intelligent Applications» on 6th International conference on User Modeling. – Chia Laguna, Sardinia, 1997. – P. 24-30.

[Brusilovsky et al., 2007] Brusilovsky, P. User Models for Adaptive Hypermedia and Adaptive Educational Systems / P. Brusilovsky, E. Millan // The Adaptive Web : methods and strategies of web personalization / Eds P. Brusilovsky, A. Kobsa, W. Nejdl. – Berlin : Springer-Verlag Berlin Heidelberg. – P. 3-53.

[Валькман, 2011] Валькман, Ю. Р. Сравнительный анализ моделей знаний обучаемого в интеллектуальных обучающих системах / Ю.Р. Валькман, А.В. Скляр // Збірник наукових праць Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України. – 2011. – Вип. 59. – С. 74-86.

[Шункевич и др., 2016] Шункевич, Д.В. Формальное семантическое описание целенаправленной деятельности различного вида субъектов. / Д.В. Шункевич // Открытые

семантические технологии проектирования интеллектуальных систем (OSTIS 2016) : материалы VI Междунар. научно-техн. конф., Минск, 18-21 февр. 2016 г. / БГУИР ; редкол.: В.В. Голенков [и др.]. – Минск, 2016.

[Давыденко, 2016] Давыденко, И.Т. Структура баз знаний систем, управляемых знаниями, и средства их разработки. / И.Т. Давыденко [и др.] // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS 2016) : материалы VI Междунар. научно-техн. конф., Минск, 18-21 февр. 2016 г. / БГУИР ; редкол.: В.В. Голенков [и др.]. – Минск, 2016.

[Пивоварчик, 2015] Пивоварчик, О. В. Компонентная архитектура интеллектуальной системы консультационного обслуживания и обучения разработчиков программ / О. В. Пивоварчик // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS 2015) : материалы V Междунар. научно-техн. конф., Минск, 19-21 февр. 2015 г. / БГУИР ; редкол.: В.В. Голенков [и др.]. – Минск, 2015.

[Шункевич и др., 2015] Шункевич, Д. В. Методика компонентного проектирования систем, управляемых знаниями / Д. В. Шункевич, [и др.] // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS 2015) : материалы V Междунар. научно-техн. конф., Минск, 19-21 февр. 2015 г. / БГУИР ; редкол.: В.В. Голенков [и др.]. – Минск, 2015. – с. 93-110.

SEMANTIC USER MODEL OF INTELLECTUAL REFERENCE SYSTEM OF THE PROGRAM DESIGN

Pivovarchyk O.

*Baranovich State University,
Baranovich, Belarus
pivovarchyk@gmail.com*

The paper considers the review of the adaptive techniques to provide user adaptation for computer systems and the existing user models. It presents a semantic user model of intellectual reference system of software developers which is developed on the basis of Technology OSTIS.

Introduction

One of the criteria of intellectuality of the computer system is able to adapt to the individual needs of the user. The goal of this work is to develop the user model for intelligent systems of consultation providing and training of software developers. The system has a functionality of reference and training systems.

Main Part

The user model is a component of intelligent systems of consultation providing and training of software developers which is developed on the basis of Technology OSTIS. In accordance with the general sc-model of computer systems developed on the basis of Technology OSTIS the user model includes a knowledge base, a knowledge processing machine and interface.

Conclusion

The inclusion the user model as a component in a training system allows to adjust adaptation to the educational material and does not require significant time for implementation of adaptive parameters of the interface.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891.2

ЭКСПЕРТНЫЕ МЕТОДЫ ОЦЕНКИ, КАЧЕСТВА И НАДЕЖНОСТИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНЫМ ПРОЦЕССОМ

Шарипбай А.А., Ниязова Р.С., Кузенбаев Б.А.

*Евразийский национальный университет им. Л.Н. Гумилева,
г. Астана, Республика Казахстан*

Sharalt@mail.ru

Rozamgul@list.ru

Bekz@bk.ru

Данная статья рассматривает управление вузом, его функции и особенности, методы оценки качества и надежности автоматизированной системы управления образовательным процессом (АСУ ОП), уровни интеллектуальной системы управления образовательным процессом, задачи оценки качества программных средств, методы экспертных оценок, а также современное состояние методов оценки, качества и надежности АСУ ОП.

Ключевые слова: образовательный процесс, оценка качества, экспертные методы.

Введение

Управление современной социальной системой, какой является высшее учебное заведение (ВУЗ) в настоящее время невозможно без создания надежной и качественной автоматизированной системы управления (АСУ), реализующей математические и физические модели, описывающие образовательный процесс.

Сущность управления образованием заключается в поддержании целенаправленности и организованности учебно-воспитательных, инновационных и обеспечивающих их процессов в системе образования.

Подчиняясь общим закономерностям социального управления, управление образованием имеет специфические особенности, обусловленные способами постановки и достижения социально значимых целей в конкретных условиях организованного учебно-воспитательного процесса.

Применительно к образовательным системам сегодня принято определять управление как целенаправленную деятельность субъектов управления различного уровня, обеспечивающую оптимальное функционирование и развитие управляемой системы, перевод ее на новый, качественно более высокий уровень по фактическому достижению цели с помощью необходимых оптимальных педагогических

условий, способов, средств и воздействий. Данное определение управления созвучно пониманию современного менеджмента.

1. Система управления образовательным процессом

Процесс управления всегда имеет место там, где осуществляется общая деятельность людей для достижения определенных результатов.

Под управлением понимается систематическое воздействие субъекта управленческой деятельности (одного человека, группы лиц или специального созданного органа) на социальный объект, в качестве которого может выступать общество в целом, его отдельная сфера (например, экономическая или социальная), отдельное предприятие, фирма и т.п., с тем, чтобы обеспечить их целостность, нормальное функционирование, динамическое равновесие с окружающей средой и достижение намеченной цели.

Поскольку образовательное учреждение - социальная организация и она представляет собой систему совместной деятельности людей (педагогов, учащихся, родителей), то целесообразно говорить об управлении ею.

Социальное управление осуществляется путем воздействия на условия жизни людей, мотивацию их интересов, их ценностные ориентации.

Основными направлениями развития информационных технологий в работе многих высших учебных заведений являются развитие программных средств управления учебным процессом, внедрение системы дистанционного обучения, промежуточного и итогового контроля знаний, поступления в институт, организация доступа к учебно-методическим материалам.

Причём как показывает практика, наибольший эффект достигается при объединении всех решаемых прикладных задач в единую информационную среду.

Управление вузом, как социальная система, помимо выполнения основных пяти функций, таких как планирование, организация, учет, контроль и прогнозирование деятельности, имеет ряд особенностей, важнейшими из которых являются [Шубинский, 2001]:

- иерархичность управления (вуз — факультет — кафедра);
- разнообразие классов решаемых задач от подбора и расстановки кадров, финансово-экономической деятельности, управления учебно-методической работой, до задач интеллектуальной поддержки в принятии решения руководством любого уровня (кафедра, факультет, вуз);
- многообразие разработчиков программных продуктов, используемых при управлении и др.

Сущность управления образованием заключается в поддержании целенаправленности и организованности учебно-воспитательных, инновационных и обеспечивающих их процессов в системе образования.

Подчиняясь общим закономерностям социального управления, управление образованием имеет специфические особенности, обусловленные способами постановки и достижения социально значимых целей в конкретных условиях организованного учебно-воспитательного процесса.

2. Разработка интеллектуальной системы управления образовательным процессом (ИСУОП)

Разработка интеллектуальной системы управления образовательным процессом (ИСУОП) часто происходит в самом учебном заведении. Создание программных средств — дорогостоящий и трудоемкий процесс. Каждый программный продукт — это сложное изделие, обладающее множеством различных свойств. Поэтому, как разработчик, так и заказчик, должны иметь представление о комплексном показателе качества созданного программного изделия, значениях основных качественных характеристик, таких как безопасность и надежность, а также владеть методами их прогноза. Однако, вуз не имеет возможности проводить долгосрочное и дорогостоящее тестирование разработанных ИСУОП.

Традиционные методы оценки качества и надежности АСУ, приспособленные, в основном, к техническим системам, не рассматривают:

- особенности социальных систем и, в частности, образовательных систем;
- случай возможного внесения ошибки при тестировании;
- исправление групп ошибок в процессе отладки;
- влияние человека-программиста на отладочный процесс.

Кроме того, существующие методы требуют большого числа испытаний для оценки качества и надежности АСУ, что существенно удорожает и увеличивает по времени процесс отладки, делая его громоздким и недоступным для пользователя (вуз, факультет, кафедра).

Высшие учебные заведения можно отнести к большим сложным системам социального типа, важнейшим аспектом функционирования которых является качественное управление ими.

Эффективное и современное управление деятельностью вуза является важнейшей задачей, стоящей перед его руководством. Поскольку основная цель вуза - обучение, то в дальнейшем будем говорить о интеллектуальных системах управления образовательным процессом (ИСУОП).

ИСУОП в вузе осуществляется на трех различных уровнях:

- вузовский уровень - это уровень, на котором решаются задачи, относящиеся ко всем структурным подразделениям;
- факультетский уровень - это уровень, на который передается решение задач, связанных с деятельностью именно этого факультета;
- кафедральный уровень - это уровень, на который передаются задачи управления, относящиеся только к этому структурному подразделению.

Необходимо отметить, что в зависимости от специфики организации вуза, те или иные задачи могут передаваться на другие уровни.

Поэтому, для совершенствования управлением социальной системой типа вуз назрела необходимость разработки методов, позволяющих быстро, но в тоже время без потери в объективности, получить обобщенную оценку качества и дать прогноз надежности ИСУОП, расширив границы распространения программных продуктов в вузах различных ведомств [Дегтяренко и др., 2008].

3. Экспертный метод оценки уровня качества продукции

Так как, информационные технологии являются мощным инструментом для решения задач управления ВУЗом и их внедрение является

необратимым процессом, для принятия решений в системах управления вузом применяются экспертные методы. Методы экспертных оценок - это методы организации работы со специалистами-экспертами и обработки мнений экспертов. Эти мнения обычно выражены частично в количественной и частично в качественной форме. Экспертные исследования проводят с целью подготовки информации для принятия решений лицами принимающие решение (ЛПР). Для проведения работы по методу экспертных оценок создают Рабочую группу, которая и организует по поручению ЛПР деятельность экспертов, объединенных (формально или по существу) в экспертную комиссию.

Экспертные методы оценки качества продукции могут использоваться при формировании сразу общей оценки (без детализации) уровня качества продукции, а также при решении многих частных вопросов, связанных с определением показателей качества чего-либо.

Экспертный метод оценки уровня качества продукции не может быть использован, если качество можно оценить другими аналитическими или экспериментальными методами с большей точностью или с меньшими затратами.

Результаты общей экспертной оценки такого сложного комплекса свойств, каким является качество продукции, имеют элементы неопределенности и необоснованности. Поэтому экспертная оценка качества продукции в целом является предварительной, ненасыщенной информационно и только в первом приближении, ориентировочно характеризует качество оцениваемого изделия. На основе такой экспертной оценки качества, очевидно, нет возможности принимать какие-либо инженерно-технические решения. Этот метод может, например, использоваться при коммерческих сделках, когда нет конкретных (численно выраженных) сведений об уровне качества приобретаемой продукции и т.п. Экспертные методы оценки качества продукции могут использоваться при формировании сразу общей оценки (без детализации) уровня качества продукции, а также при решении многих частных вопросов, связанных с определением показателей качества чего-либо.

Экспертный метод оценки уровня качества продукции не может быть использован, если качество можно оценить другими аналитическими или экспериментальными методами с большей точностью или с меньшими затратами [Ясельская, 2004].

Результаты общей экспертной оценки такого сложного комплекса свойств, каким является качество продукции, имеют элементы неопределенности и необоснованности. Поэтому экспертная оценка качества продукции в целом является предварительной, ненасыщенной

информационно и только в первом приближении, ориентировочно характеризует качество оцениваемого изделия. На основе такой экспертной оценки качества, очевидно, нет возможности принимать какие-либо инженерно-технические решения. Этот метод может, например, использоваться при коммерческих сделках, когда нет конкретных (численно выраженных) сведений об уровне качества приобретаемой продукции и т.п.

Экспертное оценивание осуществляют экспертные комиссии, включающие в себя экспертные и рабочие группы.

Экспертная группа по оцениванию качества продукции - совокупность специалистов, организованных для экспертного оценивания качества продукции.

Рабочая группа по оцениванию качества продукции (рабочая группа) - часть экспертной комиссии по оцениванию качества продукции, которая состоит из специалистов, организующих деятельность экспертной группы и обрабатывающих суждения экспертов о качестве продукта.

Экспертной оценивание качества продукции осуществляется в четыре этапа:

1-й этап. Осуществляется формулирование цели и оценки и формирование рабочих и экспертных групп.

Цель оценивания качества продукции формулируется лицом, принимающим решение (ЛПР), до начала формирования экспертной группы.

Лицо, принимающее решение (ЛПР), - это руководитель работы, эффективность которой зависит от результата оценивания качества.

Рабочая группа осуществляет подготовку и проведение экспертного оценивания качества продукции и анализ получаемых результатов. В состав рабочей группы входит организатор, консультант по оцениваемой продукции, технические работники.

Эксперты, входящие в состав экспертной группы, должны одинаково понимать цели и задачи оценивания качества продукции, удовлетворять требования по компетенции заинтересованности в участии в работе экспертной комиссии, деловитости и объективности.

2-й этап. Осуществляется выбор методов, способов и процедур оценивания. Он выполняется рабочей группой. Рабочая группа учитывает особенности оцениваемой продукции, установленной сроки выполнения работы, ее трудоемкость, область возможного использования полученных результатов.

Определение перечня операций, которые должны быть выполнены экспертами, осуществляется рабочей группой, исходя из целей оценивания, имеющейся информации, выбранных

методов, способов и процедур оценивания, сроков и условий выполнения оценивания.

Для получения суждений экспертов используется метод опроса.

Опрос эксперта по качеству продукции (опрос) - процесс выявления экспертных суждений по качеству продукции.

Методы опроса экспертов делятся на групповой и индивидуальный. При групповом совместно опрашивается вся группа или ее часть. При индивидуальном осуществляется опрос эксперта отдельно.

Процедура опроса экспертов включает интервьюирование, анкетирование и смешанное анкетирование.

3-й этап. Осуществляется экспертной группой, члены которой выражают свои суждения в соответствии с установленными на втором этапе методами, способами и процедурами.

4-й этап. Работы экспертной комиссии. Осуществляется обработка экспертных суждений и оформляется экспертное заключение.

Задача органолептического оценивания показателей качества возникает тогда, когда нужно оценить сенсорно воспринимаемое свойство продукции, такие, как вкус продукта питания цвет ткани, различимость шрифта, соответствие моде одежды. В этом случае задача заключается в создании шкалы порядка в которой может быть осуществлена оценка. Обычно роль такой шкалы выполняет бальная шкала оценок.

Бальная шкала сенсорно воспринимаемых свойств продукции при органолептическом оценивании служит для назначения оцениваемому свойству количественных характеристик, являющихся мерой этого свойства.

Главная проблема при внедрении процедур управления качеством, как правило - персонал. Внедрение процедур повышения качества требует, зачастую, изменения корпоративной культуры предприятия. В первую очередь культуры и квалификации управляющего персонала среднего звена. Именно от этой группы управляющих зависит, удастся или нет вовлечь рядовых сотрудников в процесс тотального совершенствования технологий и контроля качества, внедрить в сознание понимание того, что повышение качества ведет к снижению затрат.

Заключение

Существует достаточно большое количество методов обработки экспертных оценок. Практически все методы, которые применяются для решения задач управления в ВУЗах, требуют автоматизации. Разработке автоматизированных систем управления учебным процессом и качеством образования посвящено немало работ. Анализируя

существующие решения, можно прийти к выводу, что системы управления не достаточно автоматизированы. Можно отметить, что задачи сбора данных решаются практически всеми видами ИС управления вузов, которые имеют учетный характер.

Таким образом, современное состояние методов оценки, качества и надежности АСУ таково, что для преодоления существующих недостатков необходимы новые подходы, оценки и критерии.

Библиографический список

[Шубинский, 2001] Шубинский М.И., Анализ программного обеспечения систем управления образовательным процессом //н.-метп.сб.Щ,ВВМУ - С-Пб., с. 17-25

[Дегтяренко и др., 2008] Дегтяренко Г.А., Совершенствование системы управления образовательным процессом с использованием автоматизированной системы управления. Отчет по НИР "Система", ВВМУ - С-Пб. 1997.

[Ясельская, 2004] Ясельская А.И. Управление качеством продукции: Учебное пособие Томск: Изд-во ТПУ, 2005.

EXPERT EVALUATION METHODS, QUALITY AND RELIABILITY OF AUTOMATIC SYSTEM MANAGEMENT OF EDUCATIONAL PROCESS

Sharipbay A.A., Niyazova R.S., Kuzembaev B.A.

*Eurasian National University L.N. Gumilyov,
Astana, Kazakhstan*

Sharalt@mail.ru

Rozamgul@list.ru

Bekz@bk.ru

Abstract: This paper examines the management of the university, its functions and features, methods of assessing the quality and reliability of the automated control system of the educational process, the levels of intellectual control systems of educational process, the task of evaluating the quality of software, methods of expert assessments, as well as the current state of assessment methods, quality and the reliability of intelligent control system of educational process.

Key words: educational process, quality assurance, expert methods.



УДК 004.822:514

ПРОБЛЕМЫ СЕМАНТИЧЕСКОЙ ОРГАНИЗАЦИИ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ И ПОДХОДЫ К ИХ РЕШЕНИЮ

Горбачёв Н.Н.

*Минский филиал Российского экономического университета им. Г.В. Плеханова
г. Минск, Республика Беларусь*

Gorbachev.NN@rea.ru

В работе рассматриваются проблемы семантической организации электронной образовательной среды, связанные с актуализацией образовательного контента и предлагаются их решения на основе модели предметной области вуза в виде онтологии. Предложены методы управления перспективными информационными запасами вуза, возникающими в результате индивидуальной и коллективной работы преподавателей и студентов с внешними источниками знаний с использованием современных информационных технологий совместной работы.

Ключевые слова: управление контентом; модель предметной области; учебно-методический контент; онтология предметной области вуза.

Введение

Размещение качественного образовательного контента ведущих мировых вузов на открытых образовательных ресурсах является вызовом для отечественных университетов. Многообразие источников информации, которая стала доступной для образования требует постоянной актуализации учебно-методических материалов, которые доступны студентам во внутренней информационно-знаниевой среде.

Современные информационно-коммуникационные технологии предоставляют возможность развития личностных знаний путем создания сетевых сообществ и использования концепции социальных сетей для организации обмена опытом и выводят на новый уровень возможности получения знаний, умений и навыков. Примером этого являются «знаниевые хабы» (knowledge hubs) – концепция сетевых сообществ электронных курсов платформы Coursera, которые служат совершенствованию личностных знаний участников и развитию учебного контента.

Это требует соответствующего совершенствования систем управления знания ВУЗов с использованием перспективных технологий моделирования.

Статья подготовлена в рамках проекта «Курс и онлайн ресурсы по теме «Экологический менеджмент в ЕС» (Environmental Management in EU course and online resources (MOOC)),

реализуемого при финансовой поддержке программы ERASMUS Plus (Jean Monet Modules).

1. Рекордный рост инвестиций, обеспечение мобильности, коллаборативная работа – основные направления развития современных образовательных технологий.

Аналитический обзор, опубликованный агентством Ambient Insight [Adkins, 2014] в январе 2015 г [1] показал рекордный рост частных инвестиций в развитие частных разработчиков образовательных технологий с 1,6 млрд. долл. США в 2013 году до 2,3 млрд. долл. США в 2014 году (для сравнения объем инвестиций составлял около 150 млрд. долл. США в 2004 году). При этом основной объем разовых инвестиций составил от 1 до 10 млн. долл. США. Еще одним интересным фактом, который следует из этого аналитического отчета является рост числа компаний, получивших крупные инвестиции от 50 до 100 млн. долл. США с 4 в 2012 году до 12 в 2014 году. Из этих 12 компаний 5 находится в США, 4 в Китае, и по одной в Канаде, Индии и Бразилии. Из них услуги электронного обучения для самообразования (Self-paced elearning) предоставляют 5 компаний, услуги коллаборативного обучения (Collaboration based learning) предлагает 4 компании, мобильного обучения (mobile learning) 2 компании и 1 компания занимается разработкой программного обеспечения для научного цитирования.

В Китае в 2014 году самая значительная инвестиция в технологии электронного обучения (TutorGroup) сделана крупнейшим сетевым ритейлером alibaba.com в размере 135 млн. долл. США. При этом все инвестиции в онлайн обучение в Китае в 2014 году оценены в 554,4 млн. долл. США, основные из них – в создание языковых курсов (191,6 млн. долл. США), академические курсы университетов (143,2 млн. долл. США), профессиональное образование (50,8 млн. долл. США), техническое обучение (47,0 млн. долл. США) и развитие технологий и платформ (46,1 млн. долл. США). Такие инвестиции обусловлены высокой востребованностью электронного обучения и появлению новых областей применения технологий электронного и коллаборативного обучения. Например, платформа Chunyu, которая разрабатывалась для коллаборативного обучения врачей и медицинских работников получила в 2014 году инвестиции в объеме 50 млн. долл. США, а Xingshuai Teach, предоставляющая курсы в области графического дизайна получила в 2013 и 2014 году 32,4 млн. долл. США на развитие более 2000 курсов, которые преподаются более 100.000 студентам. Анализируя инвестиции в основные направления электронного обучения, можно выделить 3 основных группы технологий, в развитие которых в 2014 году вложены наибольшие суммы: - электронные курсы для самообразования (около 300 млн. долл. США); - технологии и контент для мобильного обучения (около 200 млн. долл. США); - коллаборативное обучение (около 140 млн. долл. США).

Значительный рост инвестиций в разработку качественного образовательного контента, значительный объем которого размещен в открытом доступе в виде MOOC (Massive open online courses, Массовый открытый онлайн-курс) – обучающих курсов с массовым интерактивным участием – является вызовом для отечественных вузов, которые не стали лидерами в области технологичного обучения. Вместе с тем, анализ мирового опыта разработки и использования таких курсов в учебном процессе позволит кратчайшим путем подойти к разработке современного контента и внедрению перспективных методов передачи знаний.

2. Обзор основных тенденций в разработке образовательного контента

Систематизация знаний и целенаправленное развитие компетенций.

В большом разнообразии открытых электронных курсов, которые становятся доступны, даже в рамках одной, внутренней информационно-знаний среды крупного университета (например, лидеров в развитии этого направления - Массачусетского технологического института или Университета Гарварда) зачастую сложно сориентироваться, чтобы целенаправленно развивать выбранное направление обучения. Для решения этой задачи, в рамках платформы вводятся

«электронные специализации» - группы коротких курсов, которые связаны общей тематикой и датами. Единая календарная сетка имеет еще один важный аспект – повышение мотивации в изучении тематически связанных курсов.

Смена моделей: Blended education против Small Private Online Course.

Происходит изменение подхода к обучению – от смешанного, где часть занятий переносилась в онлайн к системе практикоориентированных занятий. В этой модели лекции и задачи, которые можно оценить автоматически переносятся в режим онлайн, а практические занятия, выполнение коллективных учебных проектов проводятся с привлечением преподавателя-эксперта. Такие курсы – ограниченные по числу участников - могут перезапускаться в зависимости от их востребованности в середине семестра.

Привлечение максимального числа партнеров.

Ограниченность в количестве разработчиков даже у крупных платформ MOOC приводит к тому, что все перспективные идеи не могут быть быстро реализованы для всех электронных курсов. Увеличение числа партнеров позволяет быстро разрабатывать и интегрировать новые элементы курсов (интерактивные тренажеры, конструкторы и т.д.), например, интерактивный конструктор интегральных схем (Coursera) и т.д.

Перманентная актуализация контента.

Массовая доступность MOOC позволила значительно увеличить количество людей, которые могут оценить качество контента и предложить конструктивные усовершенствования. Коллаборативная работа с материалами курсов привела к тому, что учебные группы преобразуются в сетевые учебные сообщества, в которых организуется передача персональных знаний в рамках образовательного процесса. Целями функционирования сетевого учебного сообщества являются: удовлетворение потребностей всех его членов в развитии индивидуальных компетенций на основе работы с внутренними и внешними источниками знаний; коллективное совершенствование знаний и использование их использование при совершенствовании контента.

Моделирование предметных областей.

Сложная структура и узкая специализация некоторых курсов потребовали от разработчиков платформ MOOC внедрения и развития технологий моделирования предметных областей и обеспечения интеллектуального поиска с использованием контентных метаданных. Контентные метаданные используются для построения логической последовательности изучения курсов (от общих к специализированным), составлении матриц компетенций (что нужно изучить, чтобы получить требуемые навыки), расширения функций поиска по контенту (с учетом высокой доли мультимедийного контента).

3. Пример реализации семантического подхода к организации информационно-образовательной среды

Для управления учебно-методическим контентом, развитием индивидуальных компетенций и организации целенаправленной деятельности сетевых учебных сообществ нами было предложено выполнить моделирование и построить онтологию предметной области на основе квалификационных требований (матрицы компетенций), расширяемой для описания актуальных компетенций [Гринберг, 2009].

Онтология имеет структурное значение, являясь основой для управления контентом электронных ресурсов и актуализации существующих образовательных информационных ресурсов, управления перспективными информационными

запасами.

Управляющая функция онтологии реализуется с использованием метаданных, описывающих объекты репозитория и обеспечивающих многократную публикацию обновлений во всех тематически связанных элементах электронных курсов.

Применительно к системе управления знаниями (СУЗ) ВУЗа модель предметной области в виде онтологии позволяет составить целостное представление у обучающихся и обучаемых о взаимосвязях ключевых понятий от теоретических к специальным дисциплинам. Это обеспечивает управление учебно-методическим контентом с использованием онтологически связанных метаданных для целенаправленного развития необходимых компетенций

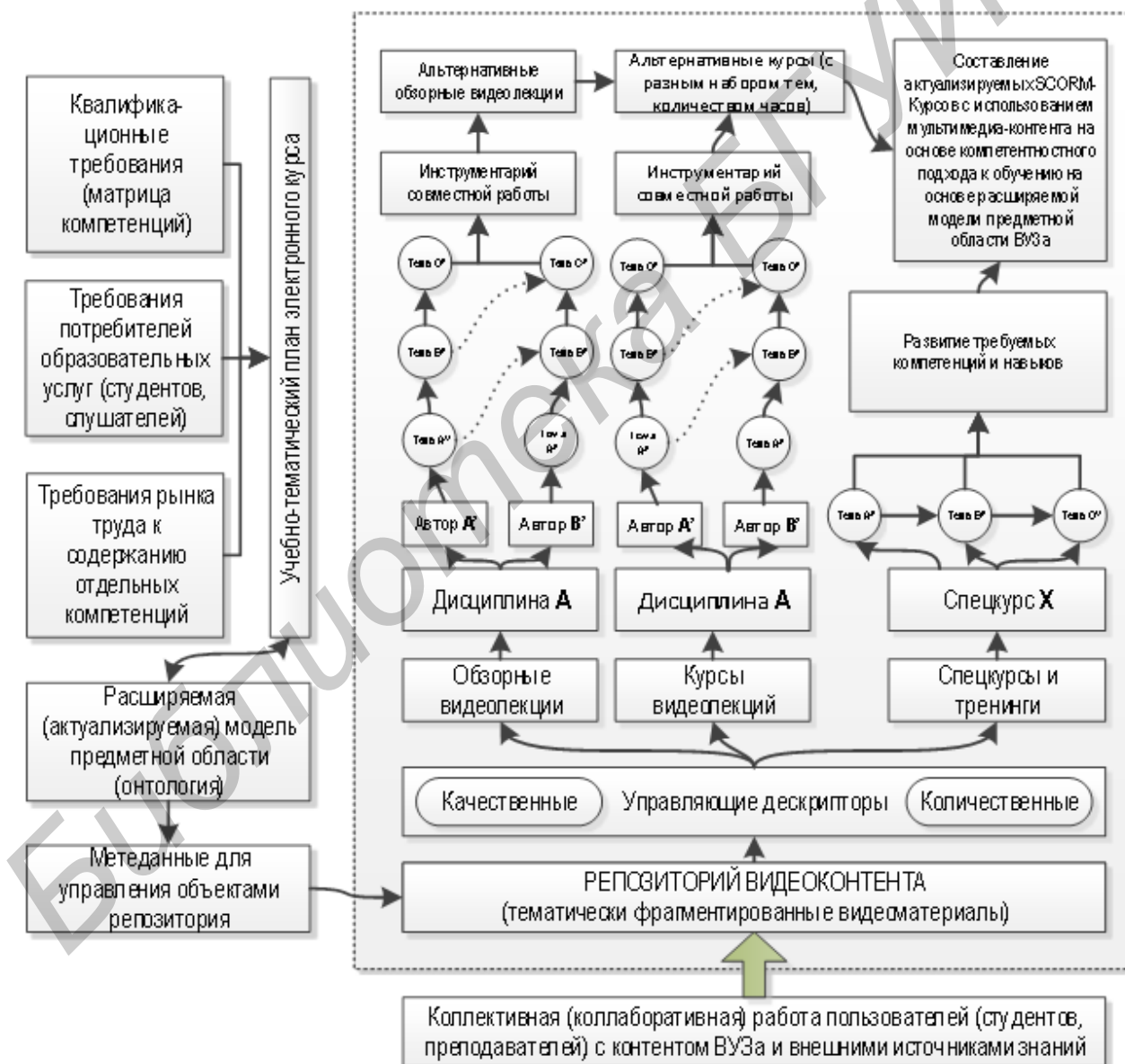


Рисунок 1 – Организация репозитория видеоконтента на основе онтологии предметной области ВУЗа

Заклучение

Развитие средств совместной работы, технологий моделирования предметных областей знаний позволяют по-новому организовать процесс обучения, превратить учебные группы в сетевые сообщества. В сложившихся условиях глобальной конкуренции на рынке образовательных услуг и при имеющихся технологических возможностях, ключевой задачей вузов является подготовка качественного образовательного контента, использование аннотированных открытых образовательных ресурсов, интегрированных в информационно-знаниевую среду.

Библиографический список

[Гринберг, 2009] Горбачёв Н.Н., Гринберг А.С. Инструментальный комплекс управления динамической публикацией образовательных информационных ресурсов // Открытое образование, №3, 2009. С. 34-43.

[Тельнов, 2015] Тельнов, Ю.Ф., Принципы и методы семантического структурирования информационно-образовательного пространства на основе реализации онтологического подхода / Ю.Ф. Тельнов // Экономика, статистика и информатика. Вестник УМО. 2014. № 1. С. 187-191.

[Adkins 2014] Adkins, Sam S. International Learning Technology Investment Patterns, Ambient Insight, 2015

PROBLEMS OF SEMANTIC ARRANGEMENT OF EDUCATIONAL CONTENT AND APPROACHES TO THEIR SOLUTION

Gorbachev N.N.

*Minsk Branch of the Russian Economic University.
GV Plekhanov (Republic of Belarus)*

Gorbachev.NN@rea.ru

The paper describes the problems of semantic organization of e-learning environment associated with the actualization of educational content and offer them solutions based on the domain model of the university as an ontology. Semantic methods of information management are applied to management of multimedia educational content and individual and collaborative work of tutors and students with external sources of knowledge.

Introduction

Availability of quality educational content from leading universities on open educational resources is a challenge for local universities. A variety of sources of information that has become available for education requires constant updating of teaching materials that are available to students in the internal information and knowledge-environment.

This article was prepared within the project "The course and the online resources on" Environmental management in the EU »(Environmental Management in EU course and online resources (MOOC)., Implemented with the financial support of ERASMUS Plus (Jean Monet Modules).

Main Part

Modern information and communication technologies provide the opportunity to develop personal knowledge through the creation of online communities and the use of the concept of social networks for the exchange of experience and brings a new level of learning opportunities and skills. An example of this is "Knowledge hubs» (knowledge hubs) - concept of network communities e-learning platform Coursera, which serve the improvement of personal knowledge of the participants and the development of educational content.

This requires a corresponding improvement of knowledge management systems of universities using advanced modeling techniques.

Overview of the main trends in the development of educational content:

- Systematization of knowledge and targeted development of competencies.
- Change of models: Blended education against Small Private Online Course.
- Raising the maximum number of partners.
- Permanently updated educational content.
- Modeling of knowledge domains.

To manage the educational and methodological content, the development of individual competences and organization of purposeful activity network learning communities we have been asked to perform simulations and build domain ontology based on qualifications (competence matrix), expandable to describe the actual skills. Ontology has a structural significance, being the basis for the development of electronic content management and updating of existing educational information resources, advanced information management inventory.

The control function is implemented using the ontology metadata describing the repository objects and provides multiple updates in all publication thematically related elements of e-learning courses.

Conclusion

The development of collaborative simulation technology domains of knowledge provide a new organizing the learning process, learning to turn the group into a network community. In the current conditions of global competition in the education market and the available technological possibilities, the key task of universities is to train high-quality educational content, use the annotated open educational resources, integrated information and knowledge-environment.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ ПОТРЕБНОСТИ В КАДРАХ НА ОСНОВЕ СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Ровба Е.А., Бойко В.К., Войтукевич Ю.А., Лявшук В.Е., Петров С.В.

*Гродненский государственный университет имени Янки Купалы,
г. Гродно, Республика Беларусь*

**rovba.ea@gmail.com; 50v.boiko@gmail.com; yury.vait@gmail.com; uliaushuk@gmail.com;
p2004r@gmail.com**

В работе рассмотрен опыт разработки системы поддержки принятия стратегических решений в области управления сферой профессионального образования Республики Беларусь, развитие которой зависит от знания долговременных тенденций спроса и предложения кадров на рынке труда. Разработка системы потребовала интеграции данных, накапливаемых в формализованных и неформальных системах знаний республиканских органов государственного управления, а также картографической информации из открытых сетевых источников. Задача решена на основе онтологического подхода путем построения семантических сетей.

Ключевые слова: онтология; профобразование; общегосударственные классификаторы.

Введение

В последние годы руководство Республики Беларусь (далее – РБ) все чаще отмечает нарастающую необходимость упорядочить и гармонизировать потоки информации об экономическом и социальном развитии регионов и отдельных территорий страны [Указ Президента, 2011]. Начало было положено в Директиве № 2 Президента Республики Беларусь от 27 декабря 2006 г. [Директивы Президента, 2011], которая определила пути улучшения работы с информацией органов госуправления и наделила особыми полномочиями по сбору статистической информации Национальный статистический комитет РБ (Белстат). Это позволило обеспечить сбор, обработку и хранение достоверной статинформации в одном месте и одновременно обеспечить доступ к ней заинтересованных структур госуправления и субъектов хозяйствования. Однако возможность доступа к информации является необходимым, но недостаточным условием для ее использования во благо экономического развития.

Министерство образования РБ систематически накапливает значения показателей и индикаторов, о состоянии процессов сферы образования. Но за большим количеством разноформатных данных организованных в таблицы по учреждениям образования и регионам тяжело увидеть полноту

картины развития системы образования на всех уровнях, тем более в динамике за несколько лет. Когда в условиях оптимизации госрасходов актуализировался вопрос повышения отдачи от сферы образования, то наиболее остро он встал перед системой профобразования, тесно связанной с реальным сектором экономики. Поэтому министерство инициировало разработку системы поддержки принятия стратегических решений на основе понимания своего потенциала подготовки кадров рабочих специальностей и информации о тенденциях процессов на региональных рынках труда, которая была бы привязана географически, статистически достоверна, визуально понятна и отражала сложность и взаимосвязь протекающих процессов. Авторы данной статьи разработали реализацию этой системы в рамках выполнения научно-исследовательского проекта «Разработка средств пространственно-временного моделирования для оценки и прогнозирования развития системы профессионального образования Республики Беларусь» (Кадастр профобразования). Перед проектом стояло две задачи. Во-первых, интегрировать информационные ресурсы, как накопленные, так и непрерывно поступающие на верхний уровень управления, для пространственно-временного моделирования, оценки и прогнозирования развития системы профобразования РБ. Во-вторых, построить многомерную модель данных, поступающих от учреждений образования с их локализацией на географической карте.

1. Этапы решения задачи

Первым этапом работы стала инвентаризация данных о деятельности учебных заведений профобразования; о потребности регионов в рабочих специальностях, их демографической и экономической ситуации. Второй этап состоял в формировании платформы интеграции данных системы профобразования из программных продуктов для формирования и сопровождения онтологий RDF (OWL), для интеграции в формат RDF разнородных данных, а также продуктов для хранения и обработки запросов к данным формата RDF(OWL). Третьим этапом стала разработка онтологий предметных областей, то есть семантическое описание сведений, распределенных в информационных базах системы профобразования, и понятий, описывающих деятельность системы профессионального образования на разных уровнях с учетом Международной стандартной классификации образования (МСКО 2011). На базе разработанных онтологий на четвертом этапе была реализована интеграция данных о географической привязке всех источников данных системы профобразования, с данными госстатистики о демографии и о реальном секторе экономики. Пятый этап заключался в организации хранения данных формата RDF(OWL) и реализации SPARQL-запросов о деятельности учебных заведений, со стороны OLAP-системы пользователя. На шестом этапе при помощи многомерных алгоритмов анализа, используемых в рамках методологии HASTD, были выявлены границы естественных группировок данных, связанных с потребностями рынка труда и возможностями их удовлетворения системой профессионального образования. На последнем этапе полученные результаты были объединены в виде информационной системы и электронных динамических карт-справочников (в части касающейся системы профессионального образования и ее связи с запросами рынка труда) на платформах MapServer и Google Earth для пространственно-временного моделирования, оценки и прогнозирования развития системы профессионального образования РБ. В качестве исходных данных для построения семантических сетей была отобрана информация из Общегосударственных классификаторов РБ, которая актуализируется Министерством экономики РБ, Национальным статистическим комитетом РБ, Министерством образования РБ; из единых квалификационных справочников (ЕКСД и ЕТКС) Министерства труда и соцзащиты РБ, международных классификаторов в области образования, а также сложившейся системы показателей оценки деятельности учреждений профобразования. Были проанализированы структура и содержание следующих документов:

- Единый квалификационный справочник должностей служащих (ЕКСД) (Постановление Минтруда и соцзащиты РБ № 32 от 30.03.2004);

- Единый тарифно-квалификационный справочник работ и профессий рабочих (ЕТКС) (постановление Минтруда и соцзащиты РБ от 30.03.2004 № 34);

- Международная стандартная классификация образования (МСКО-97, МСКО 2011 и МСКО 2013);

- Международная стандартная классификация образования и подготовки МСКО-Ф;

- Общегосударственные классификаторы ОКРБ 005-2006 и ОКРБ 005-2011 «Виды экономической деятельности» (ОКЭД);

- Общегосударственный классификатор ОКРБ 006-2009 «Профессии рабочих и должности служащих» (ОКПД);

- Общегосударственный классификатор ОКРБ 011-2009 «Специальности и квалификации» (ОКСК);

- Общегосударственный классификатор ОКРБ 014-2007 «Занятия» (ОКЗ).

Анализ представленной в сборниках статистической информации выявил проблемные места в структуре данных об экономическом развитии республики. Наиболее значимым, с точки зрения авторов настоящей статьи, является скачкообразное изменение структуры учета в разрезе видов экономической деятельности. Это связано с состоявшимся 01.01.2011 г. переходом органов статистики к формированию и предоставлению официальной статистической информации на основе ОКЭД.

ОКЭД был разработан на базе классификации видов экономической деятельности Европейского союза с введением дополнительного уровня классификации, учитывающего особенности белорусской экономики, и заменил действовавший до него Общесоюзный классификатор отраслей народного хозяйства (ОКОНХ). Переход с модели ОКОНХ на модель ОКЭД продолжается до сих пор, так как существует большое количество действующих документов республиканских органов госуправления, до сих пор использующих классификацию ОКОНХ. Для учета этого в разрабатываемой онтологии были использованы таблицы с переходными ключами не только из кодов одной модели в другую, но также из кодов ОКЭД версии 2011 г. в ОКЭД версии 2015. Обе таблицы переходных ключей переведены в цифровой формат для интеграции в разрабатываемую онтологию.

Перечисленные выше классификаторы представляют собой графовые базы данных, в некоторой степени связанные между собой. Однако формат их представления (.pdf, .doc) не позволяет напрямую сформировать из них базу данных проекта. Для приведения в требуемый формат использовался программный продукт Open Refine, с помощью которого содержание классификаторов денормализовано в плоские электронные таблицы, подготовленные к привязке в разрабатываемую онтологию. Выделенные при денормализации понятия включены в семантическую сеть,

описывающую первую версию целевой онтологии системы профобразования РБ.

В ряде случаев стандартных средств пакета Open Refine оказалось недостаточно и тогда была написана утилита для обработки сложных ситуаций «Парсер-комбинатор для преобразования текстового файла классификатора в xml-описание». Выделенные при денормализации понятия добавлены в семантическую сеть, описывающую первую версию целевой онтологии системы профобразования РБ.

При составлении семантической сети использовались возможности утилиты Grafviz [Электронный ресурс] – Режим доступа: <http://www.graphviz.org/>. Наглядность, достигаемая при использовании утилиты Grafviz, ограничена тем, что, во-первых, многомерный граф проецируется на плоскость, а, во-вторых, при большом количестве узлов сама плоскость вырастает до невообразимых размеров. Поэтому при построении онтологий понятий, содержащихся в классификаторах, представляющих собой документы размером более 250 страниц, использовался программный продукт Protege [Электронный ресурс] – Режим доступа : <http://protege.stanford.edu/>. Предоставленные в Protégé возможности визуализации упрощают работу по поддержанию классификаторов в актуальном (достоверном) состоянии, поиску совпадений и опечаток в кодировке, а также позволяют развивать семантическую сеть, дополняя ее новыми понятиями и их системами.

На базе информации и разработанных онтологий была построена семантическая сеть, описывающая потоки человеческих ресурсов между системой профессионального образования (ССУЗы и ПТО) и отраслями экономики РБ с проекцией на географию

регионов и демографическую ситуацию. Моделирование осуществлялось в среде программного продукта Protégé с помощью модуля Graphviz. Структура сети имеет сложное строение и для удобства она разбита на четыре модуля.

Первый онтологический модуль «Экономика» отражал связи и отношения между видами экономической деятельности, отраслями экономики, должностями, профессиями и занятиями в РБ, описывая иерархию связей между понятиями, определенными в справочниках ЕТКС и ЕКСД, а также классификаторах ОКОНХ, ОКЭД (версии ОКРБ 005-2006 и ОКРБ 005-2011), ОКПД и ОКЗ. Данный модуль необходим, ибо классификация отраслей экономики РБ неоднократно менялась, что приводило и к изменениям в составе показателей собранной и подвергаемой обработке информации.

Второй онтологический модуль «География» представлял собой проекции учебных заведений и видов экономической деятельности на территориально-административную структуру РБ. С его помощью произведена визуализация распределения на карте РБ учреждений системы профобразования, показателей экономического потенциала регионов и их демографической ситуации. В качестве источника картографической информации использовались данные открытых сервисов OSM (OpenStreetMap), Яндекс.Карты, Google Карты. Границы административных регионов получены из бесплатного (для некоммерческого использования) источника Global Administrative Areas (GADM). В этой базе доступны шейп-файлы трех уровней пространственного разрешения – на уровне страны, областей и районов. Хотя данные не имеют статуса официальных, но для целей проекта их точность и полнота признаны удовлетворительными.

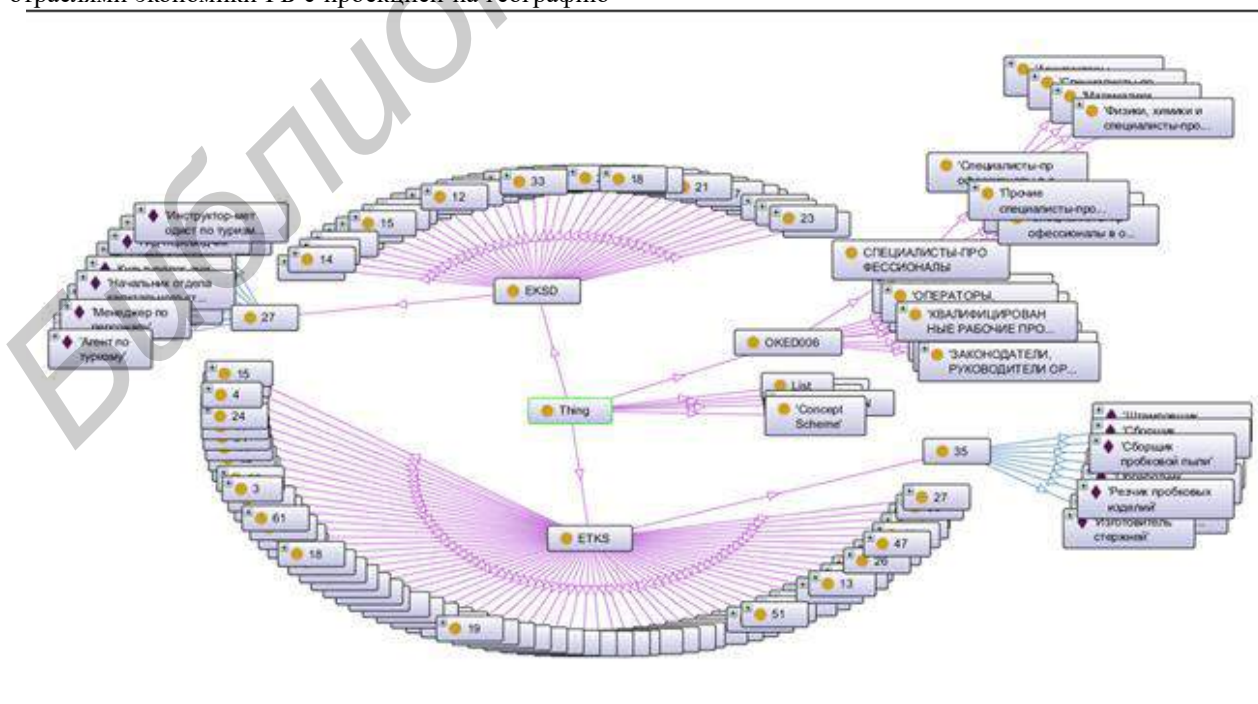


Рисунок 1 -- Граф онтологии, включающей ЕКСД, ЕТКС и ОКПД, выполненный в Protege

Третий онтологический модуль «Система профобразования» был проекцией перечня специальностей подготовки специалистов на виды экономической деятельности и региональную сеть учреждений профессионального образования. Он содержал распределение специальностей системы профессионального образования в соответствии с ОКСК по учреждениям образования, их юридические адреса для геопривязки, и историю изменения их изменений и ведомственной принадлежности, необходимую для корректной обработки информации.

Четвертый онтологический «Модуль данных» был проекцией динамических показателей экономической деятельности, демографической ситуации, результатов образовательной деятельности учебных заведений, возрастно-половой структуры и образовательного уровня трудоспособного населения на отрасли экономики и территориально-административную структуру РБ. Он содержал информацию о наборах и выпусках учебных заведений по специальностям, информацию о распределении видов экономической деятельности по регионам, о кадровом обеспечении отраслей, демографической ситуации в регионах.

Разработка перечисленных выше модулей семантической сети велась в среде программного продукта Protégé. Объединение модулей в единую онтологию выполнялось в среде программного продукта KARMA [Электронный ресурс] – Режим доступа :<http://www.isi.edu/integration/karma/>.

Заключение

В результате построения семантических сетей выявились информационные лакуны и нестыковки онтологий, лежащих в основе официальных государственных документов. Так, в комплексе экономических классификаторов проявилось отсутствие явной связи между описанием занятий работников на рабочих местах и видами экономической деятельности, реализуемой субъектами хозяйствования. Направления образования, задаваемые классификатором специальностей ОКРБ 011-2009, в ряде случаев опирались на термины из классификатора профессий ОКРБ 006-1996. Кроме этих проблемных мест построение онтологий позволило выявить недостаток практикуемого в настоящее время учета трудовых ресурсов. Он ведется по «условным поколениям» (возрастным группам), а не по реальным поколениям, что приводит к несопоставимости данных и проблемам оценки движения трудовых ресурсов. В результате в органах статистики отсутствует полнота информации о возрастном составе занятого населения и его квалификации. Перечисленные проблемные зоны, выявленные на этапе построения онтологий, с одной стороны усложнили разработку методик анализа и прогнозирования потребностей экономики в кадрах, но с другой стороны побудили к разработке ряда моделей, показывающих

относительное распределение трудовых ресурсов по регионам и отраслям экономики.

Тем не менее, среди рекомендаций, предложенных по итогам выполнения описываемого научно-исследовательского проекта, ключевыми были названы три предложения, сформулированные в процессе создания семантических сетей: разработать единую образовательную онтологию РБ, изменить систему учета трудовых ресурсов, перейдя на учет по дате рождения и квалификации (специальности) и обеспечить согласованную взаимосвязь национальных классификаторов.

Библиографический список

[Указ Президента, 2011] О некоторых вопросах сбора информации, не содержащейся в государственной статистической отчетности: Указ Президента Респ. Беларусь, 2 марта 2011 г., № 95 // Эталон – Беларусь [Электронный ресурс] / Нац. центр правовой информ. Респ. Беларусь. – Минск, 2015.

[Директивы Президента, 2011] Директивы Президента Республики Беларусь: сборник нормативных правовых актов. – Минск: Национальный центр правовой информации Республики Беларусь, 2011. – 38 с.

DESIGN OF INTELLECTUAL SYSTEM TO FORECAST STAFFING NEEDS BASED ON SEMANTIC KNOWLEDGE REPRESENTATION

Y.A. Rouba, V.K. Boika, Y.A. Vaitukevich,
U.J. Liaushuk, S.V. Piatrou,

*Yanka Kupala State University of Grodno,
Grodno, Belarus*

rovba.ea@gmail.com;

50v.boiko@gmail.com;

yury.vait@gmail.com;

uliaushuk@gmail.com;

p2004r@gmail.com

The paper describes the experience of designing of strategic decision support system in the field of vocational education of the Republic of Belarus. The system development depends on knowledge of long-term trends in staff supply and demand in the labor market. The design of the system required the integration of data accumulated in the formalized and informal knowledge systems of republican state bodies, as well as cartographic information from public online sources. The problem has been solved on the basis of ontological approach by building semantic networks.



УДК 504.06.33

АГЕНТНЫЙ ПОДХОД К УПРАВЛЕНИЮ БЕЗОПАСНОСТЬЮ ПРИРОДНО-ТЕХНОГЕННОЙ СИСТЕМЫ

Богатилов В.Н., Виноградов Г.П., Воронин Ю.А.

Тверской государственный технический университет, Россия, г. Тверь

is@tstu.tver.ru

Рассматривается система управления природно-техногенной системы на основе агентного подхода. Обеспечение безопасности населения и окружающей природной среды представляет собой весьма сложную научно-техническую задачу, решение которой невозможно без совершенствования и углубления знаний в области исследования надежности, прогнозирования и обеспечения безопасности технологических систем. Предлагается один из вариантов решения этой задачи на основе оценки идеала центра безопасности агентной системы и прогноза состояния среды на основе импульсных функций.

Введение

Безопасность функционирования природно-техногенной систем является одним из основных условий эффективного управления функционированием и развитием социально-экономических систем различного уровня. В современном обществе в настоящее время активно развиваются технологии управления на основе методологий искусственного интеллекта [Богатилов В.Н. и др., 2014]. В данной работе рассматривается подход к системе управления совокупностью территориально рассредоточенных объектов природно-техногенной системы. Разнородность факторов, различных по своей физической сути, которые оказывают влияние на разнообразные компоненты природно-техногенной системы, существенно осложняют построение формальной модели оценки безопасности региона.

Для разработки таких моделей в первую очередь необходимо провести концептуальный анализ особенностей региональной экосферы, влияющих на характеристики природно-техногенной системы.

Необходимо отметить, что теория оценки степени безопасности природно-техногенных систем находится в постоянном развитии. Опубликовано много работ посвященных оценке риска, например [Соложенцев Е.Д., 2004]. Риск рассматривается как кортеж $\langle P, U \rangle$ (P – вероятность наступления события, U – ущерб, наступления события).

Существуют и другие поисковые работы. Например в работе [Акимов Т.А. и др., 2001] и

ряде других работ в качестве параметров оценки состояний среды используется понятие экологической техноёмкости и предельно допустимая техногенная нагрузка на глобальном и региональном уровнях.

В работе [Акимов Т.А. и др., 2001] экологическая ёмкость территории определяется объемами основных природных резервуаров - воздушного бассейна, поверхностных вод, продуктивностью и запасами фитомассы. Устойчивость природно-техногенных систем определяется таким понятием как экологическое равновесие. К критерию экологического равновесия можно отнести экологическую ёмкость территории. Экологическая техноёмкость территории может служить обобщенной характеристикой территории, количественно соответствующая максимальной техногенной нагрузке, которую может в течение длительного времени совокупность экологических систем территории без нарушения их структурных и функциональных свойств.

1. Характеристика задачи управления

Главной целью управления является повышение уровня промышленной безопасности региона за счет [Яковлев С.Ю., 2000]:

- повышение обоснованности и оперативности принимаемых решений;
- обеспечение комплексного и непрерывного характера управления;
- сопряжение с другими уровнями иерархии и системами управления;

- распределение функций управления и координация работ;
- предоставление лицу, принимающему решение, всей доступной и необходимой информации в максимально удобной форме.
- К основным функциям управления относятся:
 - сбор и предварительная обработка информации (мониторинг);
 - регистрация информации, ведение баз данных и баз знаний;
 - наглядное отображение (визуализация) информации, при необходимости в картографическом виде;
 - анализ и оценка динамики обстановки;
 - оперативный и долгосрочный прогноз на моделях поведения;
 - выработка рекомендаций;
 - контроль исполнения;
 - документирование;
 - прием и передача внешней информации.

Эти функции реализуются в рамках единой организационно-технической системы и должны быть обеспечены организационно, аппаратно и программно.

Безопасность техногенной системы должны обеспечивать процесс ее эволюции так, чтобы заинтересованные в позитивных изменениях агенты искали пути разрешения проблемных ситуаций без разрушительных последствий. Это требует разработки специальной теории и методологии управления системами, основанными на обучении, адаптации и самоорганизации. В работе рассматривается один из возможных вариантов, использующий парадигму агентно-ориентированных систем, в которой для формализации поведения субъектов понятие интеллектуальный агент рассматривается, как высокоуровневая абстракция, обладающая свойствами креативности, активного поведения.

Модель внешнего окружения агента. Процесс функционирования любой системы можно рассматривать как последовательную схему смены ее состояний на некотором интервале времени (t_0, t_k). Состояние системы в каждый момент времени t из этого интервала характеризуется набором параметров этой системы, на которые накладываются ограничения $\bar{\phi}(\bar{T}, \bar{K}, \bar{U}) \leq 0$, зависящие от множеств параметров $\{T_i, K_j, U_{pr}\}$ (технологических – $\{T_i, i=1..I\}$; конструктивных – $\{K_j, j=1..J\}$; управления – $\{U_{pr}, l=1..L\}$). Выход за эти ограничения означает переход процесса во внештатную ситуацию. Эти ограничения, «вырезают» на множестве всех состояний процесса n -мерную область, в которой процесс не выходит во внештатные ситуации – это область всех работоспособных состояний процесса: $S_p \subseteq S$.

Оценка идеала. На основе оценки свойств системы агент формирует с помощью когнитивных механизмов субъективное представление о

системы. Это представление включает область возможных состояний, законы поведения системы, оценки риска.

Центр области субъективного идеала. В большинстве случаев управление складывается из целеустремлений к определенным состояниям, которые в конкретных ситуациях являются наиболее предпочтительными. Основой такого управления является представление о некоем центре идеала – точке в пространстве идеала, которая доминирует по своим свойствам над остальной областью. Задача управления в этом случае понимается как задача перевода и стабилизации свойств системы в точку как можно близкую области субъективного идеала.

Характеристика механизма принятия решений агентом в некоторой конкретной ситуации при заданном множестве управлений центра. Агент оценивает ситуацию, возникающую в системе, и ставит в соответствие каждой ситуации \mathcal{S}_i из определенного набора ситуаций S_s , характеризующего все возможные состояния объекта, некоторое управляющее решение R_i .

Перечень ситуаций, входящие в набор S_s , формируется агентом на основе своих знаний. Будем называть эти ситуации эталонами представлений агента. Входная ситуация ${}^T S_0$ сравнивается с эталонными ситуациями $\mathcal{S}_i \in S_s$, и определяется эталонная нечеткая ситуация, в некотором смысле наиболее близкая входной нечеткой ситуации. Модель операции сравнения можно построить, используя операцию нечеткой эквивалентности [Палюх Б.В., и др. 2009].

Формальное определение «нечеткой» ситуации.

Пусть $X = \{x_1, x_2, \dots, x_n\}$ – множество признаков, которые используются агентом. Каждый признак x_i описывается соответствующей лингвистической переменной $\langle \beta_i, E_i, F_i \rangle$. β_i – название лингвистической переменной; $E_i = \{E_i^1, E_i^2, \dots, E_i^{M_i}\}$ – терм-множество лингвистической переменной β_i ; F_i – базовое множество лингвистической переменной β_i .

Нечеткое равенство или эквивалентность. В качестве меры близости между ситуациями обычно рассматриваются два критерия: *степень нечеткого включения* и *степень нечеткого равенства*.

Степень включения ситуации в ситуацию обозначается и определяется выражением:

$$v(\tilde{S}_{X_i}, \tilde{S}_{X_j}) = \frac{\&}{X} v\left(\mu_{S_{X_i}}(X), \mu_{S_{X_j}}(X)\right), \quad (1)$$

где

$$\nu \left(\mu_{S_{Xi}}(X), \mu_{S_{Xj}}(X) \right) = \&_{E_k} \left(\mu_{S_{Xi}}(X)(E_k) \rightarrow \mu_{S_{Xj}}(X)(E_k) \right) \quad (2)$$

$$\mu_{S_{Xi}}(X)(E_k) \rightarrow \mu_{S_{Xj}}(X)(E_k) = \max \left\{ 1 - \mu_{S_{Xi}}(X)(E_k), \mu_{S_{Xj}}(X)(E_k) \right\} \quad (3)$$

Обычно считают, что ситуация \mathcal{S}_i нечетко включается в \mathcal{S}_j , $\mathcal{S}_i \subseteq \mathcal{S}_j$, если степень включения \mathcal{S}_i в \mathcal{S}_j не меньше некоторого порога включения $t_{inc} \in [0.6; 1]$, определяемого условиями управления, то есть $\nu(\mathcal{S}_i, \mathcal{S}_j) \geq t_{inc}$.

Степень нечеткого равенства. Если множество текущих ситуаций \tilde{S}_X содержит такие ситуации \tilde{S}_{Xi} и \tilde{S}_{Xj} , что \tilde{S}_{Xi} нечетко включается в \tilde{S}_{Xj} , а \tilde{S}_{Xj} нечетко включается в \tilde{S}_{Xi} , то ситуации \tilde{S}_{Xi} и \tilde{S}_{Xj} нужно воспринимать как одну ситуацию. Это означает, что при данном пороге включения t_{inc} ситуации \tilde{S}_{Xi} и \tilde{S}_{Xj} примерно одинаковы. Такое сходство ситуаций называется нечетким равенством, при этом степень нечеткого равенства равна:

$$\mu(\tilde{S}_{Xi}, \tilde{S}_{Xj}) = \nu(\tilde{S}_{Xi}, \tilde{S}_{Xj}) \& \nu(\tilde{S}_{Xj}, \tilde{S}_{Xi}) \quad (4)$$

В отличие от набора ${}^T S_s = \{ {}^T S_1, {}^T S_2, \dots, {}^T S_n \}$ текущих ситуаций, набор $S_s = \{ \mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_n \} (n \leq N)$ эталонных ситуаций агента не содержит нечетко равных при заданном пороге равенства ситуаций. Предполагается, что множество S_s полно. Таким образом, ситуация \mathcal{S}_i существует для любой входной ситуации S_0 . По решающей таблице для этой эталонной ситуации определяется управляющее решение. Данный подход построен на основании метода ситуационного управления.

Субъективная оценка идеала. Индекс. Для определения субъективной оценки идеала введем понятие индекса идеала. Для оценки агентом текущего состояния системы необходимо сравнить на нечеткое равенство входную нечеткую ситуацию с нечеткой ситуацией, которая характеризует центр безопасности. При этом степень нечеткого равенства: $In(\tilde{S}_x^*) = \nu(\tilde{S}_x^*, \tilde{S}_{x_0}) \& \nu(\tilde{S}_{x_0}, \tilde{S}_x^*)$ и покажет величину, которую можно определить как субъективный индекс идеала агента.

Формализация цели. Субъективная оценка идеала. Процесс принятия решений агентом может быть промоделирован на основе лингвистических

переменных, с помощью которых формализуется качественная информация, представленная в словесной форме.

По результатам контроля функционирования системы по имеющимся функциям принадлежности заполняется база данных, на основе которой проводятся вычисления индексов субъективных оценок идеала.

Если учитываются ущербы которые возникают в процессе работы системы, то агент может определять и риск, который возникает при различных режимах работы системы. Определение индекса субъективной оценки идеала риска реализуются на основе того же механизма вычислений, как и индекса субъективной оценки идеала безопасности. Риск, в данном случае, определяется как двойка *<индекс субъективной оценки идеала безопасности, индекс субъективной оценки идеала ущерба>*

$$In_{Risk}(\tilde{S}^*) = \{ In(\tilde{S}_p^*), In(\tilde{S}_a^*) \}. \quad (5)$$

2. Прогноз состояния системы

Для прогноза состояния природно-техногенной системы в данной работе выбран один из перспективных методов на основе упреждающего прогнозирования техногенной деятельности промышленных предприятий. Это повысит вероятность предотвращения предельного негативного масштаба ущерба, превышение которого оборачивается катастрофами, экологическим ущербом.

В качестве упреждающих прогнозных моделей предлагается использовать импульсные модели. Реакция $y(t)$ системы в фиксированный момент времени t на последовательность прямоугольных импульсов, образующих воздействие $x(t)$ может быть вычислена по формуле

$$\begin{aligned} y(t) &= \omega(0) x(t) \Delta \xi + \omega(\Delta \xi) x(t - \Delta \xi) \Delta \xi + \\ &+ \omega(2\Delta \xi) x(t - 2\Delta \xi) \Delta \xi + \dots = \\ &= \sum_{i=0}^{\infty} \omega(i\Delta \xi) x(t - i\Delta \xi) \Delta \xi \end{aligned} \quad (6)$$

Точное решение получим при $\Delta \xi \rightarrow 0$. При таком предельном переходе реакция $y(t)$ на последовательность прямоугольных импульсов [т.е. на сигнал $x(t)$] стремится к реакции $y(t)$ на воздействие $x(t)$, реакция на прямоугольный импульс единичной площади $\omega(\Delta \xi)$ - к реакции на дельта функцию, т.е. к импульсной переходной характеристике $\omega(t)$, а сумма переходит в интеграл

$$y(t) = \int_0^{\infty} \omega(\xi)x(t-\xi)d\xi. \quad (7)$$

Реализовать данную прогнозную модель можно на основе на основе фильтров с конечной импульсной характеристикой (КИХ фильтров).

Замечание 1. Для КИХ фильтров N – количество коэффициентов импульсной характеристики $\omega(n)$, $n = 0 \dots, N - 1$ на единицу больше чем порядок фильтра $P = N-1$. Порядок фильтра всегда равен количеству линий задержки структурной схемы КИХ фильтра.

Замечание 2. Коэффициенты КИХ фильтра равны значениям отсчетов импульсной характеристики $\omega(n)$, $n = 0 \dots, N - 1$. Таким образом, фильтры четного порядка содержат нечетное количество коэффициентов, а фильтры нечетного порядка — четное.

Замечание 3. ФЧХ цифрового фильтра связана с групповой задержкой $\tau(\omega)$ как

$$\tau(\omega) = - \frac{d}{d\omega} \Phi(\omega), \quad (8)$$

откуда

$$\Phi(\omega) = - \int \tau(\omega)d\omega. \quad (9)$$

Процессы, представленные в КИХ, описывают выходную переменную:

$$y_k = b_k + \sum_{i=1}^n \omega_i x_{k-i}$$

где $\{\omega_i\}_{i=1}^n$ – коэффициенты импульсной характеристики, составляющие смещения оценочного модуля; b_k - объясняет расхождения между прогнозируемым и фактическим выходами.

После определения состояния, наиболее общем случае, задача состоит в том, чтобы построить стратегию поведения системы, то есть стратегию, при которой ее будет максимизироваться некоторый показатель качества. Поведение системы определяется центром на основе экспертных знаний системы принятия решений. Конечность числа этапов в данной задаче отражается в том, что важна смена состояний в течении N периодов. В целом задача может быть представлена как задача ситуационного управления.

3. Рискоустойчивое целеустремление

Процесс функционирования системы – это непрерывная смена состояний. Смена состояний происходит под воздействием внешних и внутренних факторов. Могут возникать различные внештатные ситуации. Процесс смены состояний системы сопровождается также непрерывной сменой значений индексов и происходит это

непрерывно во времени. В наиболее общем случае задача теперь состоит в том, чтобы минимизировать отклонение прогнозного значения переменных состояния от точки идеала. Т.е. определить оптимальную стратегию поведения системы—например, стратегию, при которой ее доход будет максимальным. Поведение системы определяется центром на основе знаний агента. Конечность числа этапов в данной задаче отражается в том, что важна смена состояний системы в течении N периодов. В работе задача поиска стратегии решается как задача динамического программирования.

Принятие решений основано на моделировании способов действия агентом.

Моделирование поведения агента. Моделирование поведения будем основывать на предположении, что агент располагает некоторым набором элементарных операций, из которых он строит сложные действия применимые к конкретной предметной области, то он на основе располагаемых понятий может построить последовательность операций. Если каждая элементарная операция не вызывает сомнения у агента в возможности ее реализации, то осмысление алгоритма рождает у агента состояние убежденности в реализуемости алгоритма. Таким образом, алгоритмическое описание способа действия следует рассматривать как поиск в пространстве состояний последовательности допустимых операций. Такой подход допускает его формализацию с помощью теории графов, что позволяет исследовать эффективность полученных алгоритмов.

Поскольку с элементарной операцией связано какое-либо понятие, то параллельно с выявлением алгоритма действий строится дерево онтологии понятий, которыми манипулирует агент при выработке алгоритма.

Модель поведения представляет собой описание последовательности операций, которые агент использует для достижения наблюдаемого результата. Для ее построения анализируются необходимые речевые паттерны агента и невербальные реакции. Результатом этого анализа являются пошаговые стратегии и приемы, на основе которых строится алгоритм. Одновременно выполняется анализ дерева онтологий для выявления возможности раскрытия «верхних онтологий». Формирование на этой основе новых наборов пошаговых операций является основой для построения более эффективных алгоритмов. В процессе моделирования выявляются эвристики, которые агент использует для сужения пространства возможных вариантов. Их набор является индивидуальной характеристикой агента.

Центральным понятием при таком подходе к моделированию деятельности агента является понятие дерева в пространстве состояний. Корень дерева представляет собой исходное описание

ситуации, из которого выходят ветви возможного изменения начального состояния с помощью доступных операций. Альтернативным представлением алгоритма способа действия следует считать граф состояний. Формирование алгоритма способа действия предполагает задание исходного состояния и правил формирования дерева (допустимых операций по изменению текущего состояния). Процедура порождения дерева на основе такой информации называется порождающей процедурой.

Эвристики в процедуре поиска дополнительно к порождающей процедуре (правила порождения вершин) вводят оценивающую функцию и процедуру формирования рабочих оценок. Это позволяет существенно уменьшить варианты стратегии поиска алгоритмического описания способа действия.

Статистическая оценивающая функция практически полностью зависит от задачи и в нее вкладывается знание и опыт агента. При этом изменение правил (операций) сказывается на том, какую оценивающую функцию следует использовать. Таким образом, выявление оценивающей функции, которую использует агент, является одной из центральных задач при моделировании процесса построения способа действия при алгоритмическом подходе.

Один из возможных подходов основывается на использовании статистики. Пусть некоторое исходное состояние оценивается набором признаков. Формируется база реакций и способов действия, которые приводили к определенным результатам. Это позволяет восстановить примененных в каждом случае последовательности операций и вид примененной оценивающей функции. Для более быстрого определения вида такой функции следует использовать заключение агента о правилах поиска.

Назначение процедур формирования рабочих оценок состоит в уточнении представлений о качестве получающейся новой ситуации целеустремленного состояния. И с их помощью можно определить, какие именно дочерние узлы следует раскрывать и как именно комбинируются значения оценивающей функции для них.

Алгоритмы способов действия – это приемы, которые представляют собой последовательность операций, которые ведут к выполнению того или иного задания. Их эффективность зависит от способностей и навыков, которые связывают убеждения и ценности с определенным поведением. Способности и навыки обеспечивают связи и «рычаги», позволяющие проявляться видению и идентификации ситуации, ценностям и убеждениям в виде действий в том или ином окружении. Они позволяют агенту в любое время моментально воспроизвести тот или иной навык в условиях любой задачи, ситуации или контекста. Это означает, что все психические и поведенческие

программы связаны и определяются целью, ценностями, нормами и располагаемыми средствами достижения цели.

Это предположение означает, что на каждом шаге построения способа действия агент в процессе мышления (осознанно и неосознанно) определяет цели и разрабатывает процедуру проверки того, достигнута цель или нет. Если цель еще не достигнута, агент с помощью изменений или каких-либо других средств определяет способ приблизиться к желаемой цели. Когда критерии проверки удовлетворены, агент переходит к следующей стадии построения способа действия. Таким образом, функция любой отдельно взятой части поведенческой программы заключается в том, чтобы **проверить** поступающую информацию, оценить свои успехи или приступить к **действиям**, позволяющим изменить какую-либо часть актуального переживания так, чтобы она смогла удовлетворить критериям **проверки** и можно было бы **перейти** к реализации следующей части алгоритма.

Как показано в [Виноградов и др., 2011] в качестве критерия агент использует два показателя: удельную ценность по результату и удельную ценность по эффективности новой ситуации целеустремленного состояния. Существует множество способов построения таких показателей, так в их основе лежит вся структура представлений агента, что предполагает наличие множества вариантов представлений, построенных на различных допущениях и обоснованиях. Варьируя эти обоснования, можно установить принципиально различные результаты осуществления деятельности агента. Ошибки в обоснованиях и способах проверки, не улавливаемые агентом, приводят к действиям, составляющих разницу между эффективным и неэффективным исполнением. Следовательно, для того чтобы успешно моделировать полученный результат агентом, необходимо идентифицировать следующие ключевые элементы по отношению к наблюдаемому результату:

1. Цели, которые ставит перед собой агента.
2. В какой форме он представляет эти цели.
3. Обоснование и процедуры обоснования, используемые агентом для определения эффективности и успешность действий.
4. Когнитивные шаги использует агент для построения способа действия.
3. Действия, которые предпринимает агент для достижения цели, а также специфические виды поведения как средства осуществления этих действий.
4. Реакция агента, если цель не достигнута с первого раза.

Краткое изложение методики пошаговой процедуры моделирования

Основные стадии моделирования можно представить в виде следующей последовательности

шагов:

1. Определяется специалист, способности которого требуется смоделировать, а также контексты, в которых он применяет данную способность.

2. Выполняется процедура сбора информации в соответствующих контекстах и с различных позиций восприятия, начиная с интуитивного понимания данной способности из «второй позиции», затем делается попытка воспроизвести результаты с собственной «первой позиции». Используя «третью позицию», определяется разница между способами действия на основе эксплицитной информации, и действиями агента.

3. Выполняется фильтрация собранной информации для выделения значимые когнитивные и поведенческие паттерны.

4. Сведение этих паттернов в логически связную структуру или «модель».

5. Проверка эффективности и полезности построенной модели путем опробования ее в различных контекстах и ситуациях для формирования состояния убежденности, что на ее основе можно достичь желаемых результатов.

6. Редукция модели для выявления самые простых и элегантных форм, позволяющих достичь желаемых результатов.

7. Построение плана информационных воздействий для передачи или «установки» способов действия, определенных в процессе моделирования.

8. Определение параметров и средств измерения полезности модели, а также области ее применения.

Заключение

Рассмотрено развитие идей системного подхода применительно к задачам управления симбиозом природной и техногенной систем, для реализации которых необходимо построение таких механизмов их функционирования, которые используют способность систем к самоорганизации и адаптации. Эффективность такого управления во все большей степени зависит от поведения агентно-ориентированных систем, их заинтересованности в поиске и использовании резервов и ресурсов развития организации, их способностью видеть перспективы и последствия принимаемых решений.

Библиографический список

[Богатиков и др., 2014]. Богатиков, В.Н. Рискоустойчивое управление поведением многоагентных систем / Богатиков В.Н., Виноградов Г.П., Палюх Б.В. // Труды международной научной школы МА БР – 2014 (Санкт – Петербург, 18-20 ноября, 2014 г.) ГОУ ВПО «СПбГУАП». СПб. 2014. С. 181-185. [Соложенцев, 2004]. Соложенцев Е.Д. Сценарное логико-вероятностное управление в бизнесе и в технике. Издательский дом «Бизнес пресса». Санкт – Петербург. 2004. С. 431.

[Акимова и др., 2001] Акимова Т.А. Экология. Природа - Человек – Техника / Акимова Т.А, Кузьмин А.П, Хаскин В. В. // М.: ЮНИТИ-ДАНА, 2001. - 343 с

[Яковлев, 2000] Яковлев С.Ю. Основные положения

концепции информационного обеспечения управления промышленной безопасностью региона (на примере Мурманской области) // Теоретические и прикладные модели информатизации региона: Сб. науч. тр. – Апатиты: изд. КНЦ РАН, 2000. – С.12-19.

[Алексеев и др., 2009] Палюх Б.В., Приложения метода разделения состояний к управлению технологической безопасностью на основе индекса безопасности / Палюх Б.В., Богатиков В.Н., Пророков А.Е., Алексеев В.В. // – Тверь: ТГТУ, 2009. – 368 с.

[Виноградов и др., 2011] Виноградов, Г.П. Моделирование поведения агента с учетом субъективных представлений о ситуации выбора / Виноградов, Г.П., Кузнецов В.Н. // Искусственный интеллект и принятие решений. № 3. с. 58-72.

AGENT-BASED APPROACH TO SAFETY MANAGEMENT OF NATURAL AND TECHNOGENIC SYSTEM

Bogatikov VN, Vinogradov GP, Voronin Yu A
Tver state technical university».Russia, Tver

Introduction

Safety of functioning natural and technogenic systems is one of the main conditions of effective management of functioning and development of social and economic systems of various level. In modern society now technologies of management on the basis of methodologies of artificial intelligence actively develop. The purpose of this process consists in increase of intellectuality and quality of management in decision-making processes. Such changes in technology of management have to increase ability of organizational and technological systems in general to self-organization and adaptation. In this work approach to a control system of the territory of the Tver region as associations the agentnykh of educations at the heart of creation which set of territorially dispersed objects of natural and technogenic system of various type and appointment interacting with each other lies is considered. Heterogeneity of regional factors, various on the of a physical essence which have impact on various components of natural and technogenic system, significantly complicate creation of formal model of an assessment of safety of the region.

Main Part

Report theses in volume of 3 paragraphs.

Conclusions

The report contains the results of the proposed methodological theoretical positions. We consider the development of the ideas of system approach to management, which is necessary for the realization of the construction of such mechanisms of their functioning, the ability to use the systems for self-organization and adaptation. The effectiveness of this control is increasingly dependent on the behavior of the agent-oriented systems, their interest in the search for and use of reserves and resources of the organization, their ability to see the prospects and implications of decisions



УДК 004.9

КОНЦЕПЦИЯ ИНСТРУМЕНТАЛЬНОЙ ПЛАТФОРМЫ МАРКЕТИНГОВОЙ ДЕЯТЕЛЬНОСТИ В СРЕДЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ

Вишняков В.А., Казак Е.А.

*Минский инновационный университет,
г. Минск, Республика Беларусь
vish2002@list.ru,*

Показаны основные направления маркетинговой деятельности при использовании интеллектуальных технологий. Представлены направления использования интеллектуальных технологий в управленческой деятельности с использованием ОБ. Дан анализ особенностей маркетинговой деятельности в ОБ. В качестве тенденций развития рассмотрено совершенствование методов и моделей маркетинговой деятельности в ОБ. В качестве концепции предложено создание инструментальной платформы на базе многоагентной технологии, объединяющей интеллектуальные системы маркетологов для облачной среды.

Ключевые слова: интеллектуальные технологии, маркетинговая деятельность, принятие решений, социальные сети, инструментальная платформа маркетолога

Введение

Для современного этапа развития теории и практики маркетинговой деятельности (МД) характерна такая ситуация: с одной стороны, усиленное внимание к использованию информационных объектов, повышение требований по МД, принятие международных стандартов, растущие расходы на обеспечение механизмов и средств маркетинга, с другой – недостаточная эффективность МД, о чем свидетельствуют публикуемые данные об мировой экономике [Железны М., 2002].

Выходом является внедрение на всех этапах МД интеллектуальных технологий, приобретающих все большее распространение в системах управления. С одной стороны, сбор и обработка информации из Интернета о состоянии, направлении развития и уровне товаров (услуг) или иных процессов в мировом сообществе и синтез знаний, отраженных в тех или иных источниках, осуществленный на основе их интеллектуальной обработки, дает новое интегральное качество, позволяющее прогнозировать, смоделировать и предупредить развитие тех или иных процессов на мировых рынках. С другой стороны, применение интеллектуальных технологий обработки данных дает возможность повысить уровень эффективности МД различных маркетинговых информационных систем (МИС) [Вишняков В.А., 2014].

1. Направления интеллектуализации в маркетинговой деятельности

Одной из главных проблем построения инновационных экономик является интеллектуализация, суть которой заключается в разработке эффективных механизмов формирования, публикации, актуализации и массового использования инновационных знаний в управленческих и маркетинговых технологиях. Среди таких знаний в работе [Бородаенко Ю.В., 2013] выделены: разработки в области интеллектуальных агентов на основе семантик-Вэб; Вэб-сервисов и семантических Вэб-сервисов; облачных вычислений. Полученные результаты предложено взять за основу при интеллектуализации принятия маркетинговых решений [Вишняков В.А. и др., 2014]:

– с использованием технологий Semantic Web в маркетинговой деятельности: интеллектуализация принятия маркетинговых решений; интеллектуализация электронной коммерции;

– интеграционное решение на основе семантических Web-сервисов является одним из ключевых направлений разработок в области повышения автоматизации маркетинговых процессов и интеллектуализации распределенной обработки маркетинговой информации.

Использование технологий облачных вычислений позволяет более эффективно решать

отдельные задачи в маркетинговой деятельности, в частности в области поиска информации, электронной торговли, принятия решений.

Все больше предпринимателей и владельцев Интернет-магазинов (ИМ) приходят к пониманию того, что социальные сети (СС) не только обеспечивают трафик и коммуникацию с клиентами, но и помогают продавать. Рассмотрим некоторые популярные программные пакеты для интеллектуализации маркетинговой деятельности в Интернете и СС [Вишняков В.А. и др., 2014]. Автоматизированный сервис «B2B.FileCloud» решает проблему выгрузки товаров из ИМ в СС в автоматическом режиме. Одним из примеров сервисов получения подробной картины по ценам товаров является пакет «Z-price». С помощью пакета «Ahrefs», можно получить информацию об общем количестве обратных ссылок на сайт ИМ конкурента, количестве ссылающихся страниц, количестве ссылающихся IP-адресов, реферральных доменах. Одним из них является «Rooke», с помощью которого можно в автоматическом режиме проводить SEO-анализ сайта. Инструмент для анализа языка общения в сетях *Twitter* и *Instagram* «Zipline» отслеживает ключевые слова через данные социальные сети в режиме реального времени.

2. Интеллектуализация управления в ОВ

Web 3.0 (интеллектуальная сеть) станет очередным новым этапом использования Интернета и поможет снизить избыток информации. Онтология формирует семантику, создав новые возможности для интеллектуальных агентов выполнять наши запросы. Открытое извлечение информации (Information Extraction – IE) обеспечит работу новых форм поиска, освободив пользователей от задачи по исследованию документов, выданных поисковой машиной. Сейчас широко применяются серверы исполнения деловых регламентов (BRE – Business Rules Engine). Но, чтобы справиться со сложностью бизнес-процессов, связывающих несколько предприятий или цепочку создания ценности в Web 3.0 компании потребуют создания «умных» процессов, превосходящих современные серверы исполнения регламентов [Вишняков В.А. 2014 и др., 2011].

Распределенный искусственный интеллект – DAI (Distributed Artificial Intelligence) основывается на агентных технологиях. Стандартный программный агент имеет три свойства: автономность, способность реагировать и способность выйти на связь. Простые программные агенты могут общаться с другими «сущностями»: людьми, другими программными агентами или объектами [Фингар П., 2011]. Добавив к этому способность планировать и ставить цели, поддерживать модели представлений, рассуждать о действиях и

повышать уровень знаний и качество работы через обучение, и получим главные компоненты «интеллектуального агента».

Интеллектуальные агенты могут быть интегрированы в структуры ОВ, содержащие конкретные функции по решению задач, обработки данных и управления. Они поддерживают естественное соединение информации и технологий, основанных на знаниях и могут поддерживать процесс логических рассуждений (например, включение деловых регламентов в них). Они позволяют включить функцию обучения и самосовершенствования как на уровне инфраструктуры (адаптивная маршрутизация), так и на уровне приложения (адаптивные пользовательские интерфейсы).

Интеллектуальные агенты будут использоваться для сбора бизнес-аналитики BI (Business Intelligence) и процессов обработки сложных событий – CEP (Complex Event Processing). В Web 1.0 важным показателем было количество посещений определенной страницы. Сегодня за внимание аудитории показатель посещений страниц устарел. Важно количество связей в социальных сетях, количество отправленных сообщений и время, проведенное на конкретном сайте [Фингар П. 2011].

Получение информации и непрерывный анализ в реальном времени в ОВ – это следующая сложная задача для корпоративного интеллекта, особенно когда для того, чтобы найти ценную информацию и «управлять репутацией», надо переходить от «поиска в данных» к «поиску в блогах». Надо выйти за пределы поисковика Google, обработать Интернет-шум, чтобы понять, что же происходит в отрасли, ситуацию о товарах и услугах компании, т. е. нужна аналитика Web 3.0.

Используя обработку сложных событий для корпоративного интеллекта, можно создать обратную связь между корпоративным интеллектом и системой управления бизнес-процессами, которая, в свою очередь, воздействует на корпоративный интеллект. Когда компании выводят управление бизнес-процессами в сложную деловую экосистему, формирующуюся вокруг цепочки создания ценности, ценность обработки сложных событий различных обнаружителей становится опорой для корпоративного интеллекта и анализа процессов в реальном времени, необходимых для того, чтобы создать и отточить постоянно меняющиеся бизнес-процессы.

Service Science, Management and Engineering (SSME) – термин, используемый IBM Research в своих разработках в области сервисных систем. HP создала «Научный центр систем и сервисов». Oracle Corp. присоединилась к IBM для создания индустриального консорциума под названием Service Research and Innovation Initiative. Группа NESSI (Networked European Software and Services Initiative) в Европейском союзе создана NESSI

(Рабочая группа по вопросам науки о сервисах). В Калифорнийском университете Беркли есть программа SSME. Все это происходит благодаря тому, что в сфере услуг ныне заняты более 50% рабочей силы в Бразилии, России, Японии и Германии, а также 75% рабочей силы в США и Великобритании [Ридз, Дж., 2011].

3. Маркетинговая деятельность в ОВ

Время монолитных, вертикально интегрированных компаний ушло. Главные и вспомогательные БП компании происходят в четырех взаимосвязанных областях: поставщики (прямые закупки), поставщики производственных ресурсов (непрямые закупки), торговые партнеры, клиенты. Эти многочисленные цепочки формирования ценности должны быть включены в новые бизнес-экосистемы ОВ, объединяющие «всех-со-всеми». Они будут доступны для соединения, разрыва и нового соединения в соответствии с изменениями в рыночных реалиях, предоставляя компаниям возможность работать на множественных рынках или создавать новые предложения для «рынка из одного человека».

Все эти многочисленные цепочки формирования ценности должны быть вплетены в новые бизнес-экосистемы ОВ, объединяющие «всех-со-всеми». Они должны быть доступны для соединения, разрыва и нового соединения в соответствии с изменениями в рыночных реалиях, предоставляя компаниям возможность работать на множественных рынках или создавать уникальные предложения для «рынка из одного человека».

Успешные компании стали представителями интересов своих клиентов. Они работают с поставщиками со всего мира для того, чтобы предложить клиентам наилучшую ценность. Ответы на вопросы кто же владеет бизнес-процессами всей цепочки создания ценности лежат за пределами CRM-систем – в новых системах ОВ: управления отношениями в цепочке создания ценности (Value-Chain Relationship Management) и управления отношениями с сообществом клиентов (Customer Community Relationship Management). Это порталы, бизнес-экосистемы и системы формирования информации. Это находится за границей продаж, маркетинга и услуг, предоставляемых одним и тем же предприятием. Теперь это процесс, лежащий вне системы одного предприятия, как, например, «индивидуальный запрос на продукт», поступающий по многочисленным каналам и от множества сообществ клиентов [Вишняков 2014, Фингер 2011].

Управление CRM 2 будет располагаться в ОВ. То же самое можно сказать и об управлении циклом жизни товара в новом мире инноваций, формируемых потребителями: системы автоматизированного проектирования, управления системами поставок и контрактными отношениями

должны будут выйти за пределы границ предприятия и включить клиентов и партнеров по дизайну и производству. Поскольку ни одна компания не «владеет» всей последовательностью создания ценности, бизнес-процессы компаний и системы их управления будут унифицированы и перенесены в ОВ.

Данная архитектура включает базу знаний в виде правил продукции, механизма логического вывода, рецепторов и эффекторов агента, модуль коммуникации с другими агентами. Применительно к задаче анализа рынка, рецепторы передают факты о внешних воздействиях в базу знаний. В результате логического вывода вырабатывается решение, которое передается эффектору об изменениях внешней среды.

Для распределенного решения задач могут быть использованы разные типы агентов: агент-субординатор, множество агентов исполнителей, агент-интегратор. Агенты могут быть связаны между собой в виде многоуровневой архитектуры, которая может быть горизонтальной или вертикальной. Для решения задачи обнаружения атак подходит вертикальная многоуровневая архитектура агентов. С учетом специфики решаемой задачи проектируемая многоагентная система должна включать несколько агентов, которые выполняют в системе различные функции.

В результате анализа информационного процесса распределенной МД в сетях можно рассматривать агентов, разграничивающих права доступа пользователей сети, агентов обнаружения потребностей, то есть изменения состояния рыночной среды в сети, агентов обнаружения инноваций, агентов, строящих сценарии поведения для распространения инноваций, агентов, являющийся посредником-координатором всей многоагентной системы.

В работе [Вишняков В.А., 2014] проанализированы основные разработки в области ИМД и рассмотрены основные тенденции их развития. В результате этого был определен перечень критериев и их значений, которым должна удовлетворять ИС МД:

- многоуровневость наблюдения за средой. СОА должна собирать сведения о состоянии рынка из различных источников на различных уровнях наблюдения – уровень сети, серверов и социальных систем;
- адаптивность, т. е. способность обнаруживать модифицированные реализации известных и новых инноваций рынка;
- проактивность, ИС МД должна обладать встроенными механизмами реакции на инновацию;
- открытость, ИС МД должна обладать возможностью добавления новых анализируемых ресурсов информационной системы.
- тип управления. ИСМД должна совмещать как централизованное, так и распределенное управление.

– защищенность. ИСМД должна обладать средствами защиты своих компонентов.

В результате представлены следующие решения по многоагентной системе обнаружения инноваций рынка ИС МД:

- структура и состав многоагентной системы обнаружения инноваций, включающая в себя агентов рабочих станций, серверов, маршрутизаторов и сетей и позволяющая делать вывод о состоянии и перспективах развития рынка;
- метод принятия агентами совместного решения, позволяющий сформировать круглый стол агентов и на основании их результатов анализа сведений, полученных из различных источников, оценить состояние рынка в целом;
- методика обнаружения атак с использованием многоагентных технологий, позволяющая обучить многоагентную систему обнаружению инноваций и использовать ее для дальнейшего обнаружения новых товаров (услуг);
- оценка эффективности всех предложенных методов, используя разработанные программные решения МД.

4. Концепция развития платформы МД в среде ОВ

В качестве тенденций и концепции развития по использованию ИТ в МД можно представить следующее [Вишняков В. А., 2014]:

- совершенствование архитектур систем МД в ОВ, обеспечивающих эффективное управление в условиях неопределенности состояния информационной среды;
- разработка новых моделей МД в ОВ и СС на основе выбора оптимального варианта реагирования на события рынка;
- совершенствование инструментальных программных комплексов для МД с интеллектуальной поддержкой принятия решений и исследованием эффективности методов, моделей и алгоритмов;
- развитие технологий многоагентных систем МД для обнаружения атак, противодействия угрозам нарушения ИБ, оценки уровня защищенности информации в КИС;
- разработка теоретических основ, моделей и средств облачной инструментальной платформы проектирования интеллектуальных систем МД на основе семантических технологий;
- разработка прикладных АРМов или сайтов для маркетологов с использованием рьяочных технологий.

Заключение

Маркетинг стал цифровым процессом, размывая грань между ИТ и маркетингом и превращая ИТ в необходимый ингредиент МД [Махаян Р., 2015].

Первым направлением в интеллектуальной распределенной МД является дальнейшая

разработка моделей, методов, архитектур и программных средств управления для решения проблемы адаптации на рынках и облачной инструментальной платформы проектирования интеллектуальных систем на основе семантических технологий.

Другим направлением ИМД является разработка моделей, методов, архитектур и программных средств сбора, структуризации информации из Интернете, формирования специализированных баз знаний и поддержки принятия решений (на базе ИТ) по всему накопленному аспекту задач МД.

Библиографический список

- [Железны М., 2002] Информационные технологии в бизнесе. / Под редакцией Железны М. – СПб.: Питер, 2002. – 1114 с.
- [Вишняков В.А., 2014] Вишняков, В.А. Информационное управление и безопасность: методы, модели, программно-аппаратные решения. Монография. / В.А. Вишняков / – Мн. МИУ, 2014. – 287с.
- [Бородаенко Ю.В., 2013] Бородаенко Ю.В., Направления интеллектуализации в современном маркетинге. / Ю.В. Бородаенко, В.А. Вишняков /Актуальные проблемы науки XXI века / Мн.: МИУ выпуск 2, 2013. – с. 91-96.
- [Вишняков В.А. и др., 2014] Вишняков, В.А. Методы, средства поддержки принятия маркетинговых решений, направления их интеллектуализации. / В.А. Вишняков, М.С. Жерносек / Актуальные проблемы науки XXI века / Мн.: МИУ выпуск 2, 2014. – с. 91-96.
- [Вишняков В.А.и др., 2014] Вишняков, В.А. Интеллектуальные средства в Интернет-маркетинге / В.А. Вишняков, Е.А. Казак // Мат. 20 межд. НТК «Современные средства связи». – Минск, ВГКС, 2014. – С.256.
- [Ридз, Дж., 2011] Ридз, Дж. Облачные вычисления. / Дж. Ридз / – С.Петербург: БХВ, 2011. – 288 с.
7. Облачные сервисы. Взгляд из России. Под ред. Е. Гребнева. – М.: Cnews, 2011. – 282 с.
- [Фингар П., 2011] Фингар П. Облачные вычисления – бизнес-платформа XXI века. Пер. с англ. Захаров А.В. / П. Фингар / – М.: Акваринная Книга, 2011. – 256 с.
- [Махаян Р., 2015] Махаян Р. Цифровой маркетинг: ИТ и будущее. [Электронный ресурс]. – режим доступа : <https://technet.microsoft.com/ru-ru/magazine/gg981693.aspx>. Дата доступа : 27.03.2015.

INSTUMENTAL PLATFORM CONCEPTION OF MARKETING ACTIVITY IN CLODE COMPUTING AREA WITH INTELLIGENCE TECHNOLOGIES USE

Vishniakou U.A., Kazak E.A.

*Minsk Innovation University,
Minsk, Republic of Belarus
vish2002@list.ru*

Main directions of marketing activity with use of intelligence technologies are shown. Main directions of intelligence technologies use in management activity with cloud computing (CC) are represented. The analysis of marketing activity particulars in area of CC directions is done. As trend the perfection of methods, models development of marketing activity in CC area is given. As conception development of intelligence marketing activity the crtiating instrumental platform on the multy agent technology conjuncting intelligence marketing systems in cloud computing is proposed.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.853

СЕМАНТИЧЕСКИЙ АНАЛИЗ СОЦИАЛЬНЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ WOLFRAM MATHEMATICA

Тякунов А.С., Славский В.В.

*Югорский государственный университет,
г. Ханты-Мансийск, Российская Федерация*

atyakunov@mail.ru

slavsky2004@mail.ru

В работе проводится обзор возможностей среды Wolfram Mathematica, используемых для семантического анализа открытых данных в социальных сетях. Рассмотрены варианты использования запросов Application Programming Interface (API) и их интерпретации. Подготовлена основа для создания учебно-методических пособий по использованию среды Wolfram Mathematica.

Ключевые слова: семантический анализ; социальные сети; API, теория графов.

Работа выполнена при поддержке Совета по грантам Президента РФ (грант НШ 2263.2014.1), Правительства РФ (госконтракт 14.B25.31.0029), РФФИ 15-41-00092 р-урал-а, 15-41-00063 р-урал-а, 15-01-06582 А.

Введение

Онлайновые социальные сети, помимо выполнения функций поддержки общения, обмена мнениями и получения информации их членами, в последнее время все чаще становятся объектами и средствами информационного управления и ареной информационного противоборства. В недалеком будущем они неизбежно станут существенным инструментом информационного влияния, в том числе, в целях манипулирования личностью, социальными группами и обществом в целом.

Таким образом, потребность в полном и качественном анализе социальных взаимодействий посредством социальных сетей является важной задачей современности, в том числе и с математической точки зрения. Для ее решения требуется наличие инструментов, позволяющих обрабатывать запросы к социальным сетям с целью получения понятных результатов анализа. Одним из таких инструментов является среда Wolfram Mathematica.

Математическое программное обеспечение Mathematica, известно как самое мощное в мире вычислительное приложение. Более того, Mathematica служит единой платформой для проведения исследований и вычислений, которые находят свое отражение в живых интерактивных документах издательского качества. Используя

мощный вычислительный аппарат и удобный интерфейс, пользователи Mathematica могут легко решить трудоемкие задачи и разобраться в сложных концепциях, используя весь имеющийся арсенал аппаратных средств.

Цель данной работы – продемонстрировать возможности среды Wolfram Mathematica 10 для масштабного анализа связей пользователей социальных сетей, графического представления и анализа данных, размещенных в открытом доступе.

1. Общие определения

Термин "сеть" имеет разные значения в разных дисциплинах. В социальных науках, сеть, как правило, определяется как набор субъектов сети (или агентов, или узлов, или точек, или вершин), которые могут иметь отношения (или ссылки, или ребра, или связи) друг относительно друга [Hanneman et al, 2005]. Сети могут иметь несколько или много узлов, и один или более видов отношений между парами узлов (Рисунок 1).

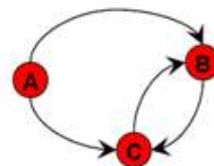


Рисунок 1 – Пример направленной сети

Социальная сеть – платформа, онлайн-сервис или веб-сайт, предназначенные для построения, отражения и организации социальных взаимоотношений в Интернете.

Получать данные из социальной сети удобнее всего посредством готовых API-функций. API (интерфейс программирования приложений) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах.

Стандартный формат ответа от API — это JSON (англ. JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript и обычно используемый именно с этим языком. Как и многие другие текстовые форматы, JSON легко читается людьми.

Это универсальные структуры данных: как правило, любой современный язык программирования поддерживает их в той или иной форме. Они легли в основу JSON, так как он используется для обмена данными между различными языками программирования. В частности, JSON легко конвертируется в списки и правила внутренними средствами Mathematica.

2. Запросы в социальных сетях

2.1. Использование API-функций социальной сети "ВКонтакте"

Социальная сеть "ВКонтакте" (<http://vk.com>) выбрана полем для исследования по следующим причинам:

- это одна из крупнейших мировых и крупнейшая русскоязычная социальная сеть;
- формат запроса достаточно прост и масштабируется, позволяя получать разнообразные данные о пользователе;
- для создания запросов на получение данных не обязательно (по крайней мере, на сегодняшний день) создавать standalone-приложение, передающее ключи доступа;
- учетная запись данной сети имеется у большинства пользователей в российском сегменте Интернет.

В общем случае, для того чтобы вызвать метод API сети "ВКонтакте", необходимо осуществить POST- или GET-запрос по протоколу HTTPS на URL <https://api.vk.com>. Для получения данных внешнего сайта или файла в Mathematica используется команда Import. Передавая в нее URL запроса, мы получим данные, которые размещаются в сети Интернет по соответствующему адресу. Используя необходимый формат (в данном случае - описанный выше JSON), можно легко получать необходимые для исследования данные.

2.2. Запрос данных о пользователе социальной сети

Для получения данных из сети "ВКонтакте" необходимо проанализировать структуру запроса. К примеру, мы желаем получить самую общую информацию о пользователе. Для этого будет использоваться метод `users.get`, имеющий параметр `user_id`, в котором размещается ID пользователя.

Перечисляя необходимые поля, которые необходимо вернуть, можно, к примеру, получить имя, фамилию, фотографию пользователя, его место жительства и пол. Mathematica позволяет совмещать типы данных внутри одного списка, что чрезвычайно удобно при обработке данных, полученных из социальной сети (Рисунок 2).


```
{Александр, Тякунов, Ханты-Мансийск, , Мужской}
```

Рисунок 2 – Вывод общей информации о пользователе

Получение информации, размещенной на стене и в ленте пользователя производится аналогичными средствами и позволяет исследователю, в том числе, анализировать и интерпретировать полученные результаты согласно выдвигаемым гипотезам. Так, например, аналитическим агентством "Скориста" (г.Москва) в 2014 г. был представлен алгоритм предсказания поведения кредитного заемщика на основе открытых данных. Проверая гипотезы о связи открытости профиля предполагаемого заемщика, а также частоты и направленности публикаций с предполагаемой добросовестностью, был сделан вывод, что чем больше люди говорят в социальной сети, тем они хуже возвращают займы.

2.3. Анализ социальных связей пользователя

Особый интерес для исследователя представляет, как правило, изучение и анализ информации о тех пользователях, с которыми связан данный узел. Для получения информации о "друзьях" пользователя в социальной сети "ВКонтакте" используется метод `friends.get`, который формируется аналогично `users.get`. К примеру, достаточно простым, но весьма наглядным могут быть графические представления ответов на запросы о половом разделении списка "друзей" (Рисунок 3) или градации по месяцам рождения (Рисунок 4). В контексте изучения Wolfram Mathematica подобные примеры позволяют осваивать функционал, не теряя интереса обучающегося, поскольку основаны на данных его собственного профиля социальной сети.

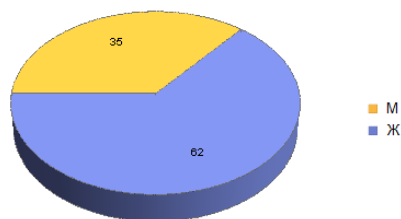


Рисунок 3 – Распределение "друзей" пользователя по полу

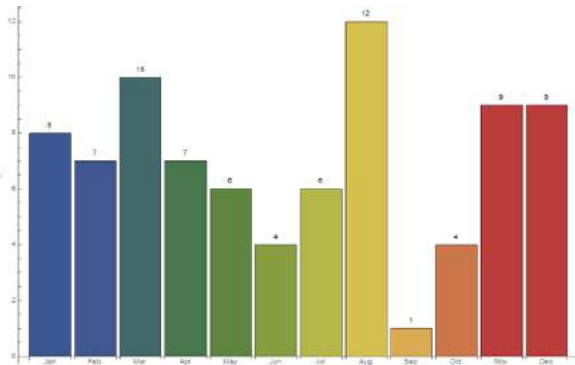


Рисунок 4 – Распределение дат рождения "друзей" пользователя по месяцам

С другой стороны, анализ данных о "друзьях" пользователя может быть использован и при проверке определенных гипотез [Watts, 1999]. В частности, практически подтвержденный факт того, что значительную (от 40% всех "друзей") часть связей в социальной сети составляют пользователи, проживающие в том же населенном пункте, что и сам пользователь, позволяет предположить, что подобное распределение характерно для всех подлинных пользователей сети. Это аналогично и для возраста, социальных убеждений, интересов и т.д. В то же время, как известно, многие социальные сети позволяют скрыть дату рождения, город проживания и другие поля. Таким образом, проверяя распределение "друзей" пользователя по параметру "город" с определенной долей уверенности мы можем сказать, что сам пользователь проживает в том городе, что и большинство его друзей (Рисунок 5). Подобное предположение позволяет, например, выявить недостоверные ("фейковые") учетные записи, созданные для рекламных или иных целей. Пользователи, у которых нет четко выделенной группы "друзей" (Рисунок 6), проживающих в одном населенном пункте (или регионе), безусловно, могут существовать, однако, это не является характерным для большинства. Аналогичная ситуация обстоит и с указываемым годом рождения. Такая проверка достоверности может быть использована, в том числе, при проверке добросовестности в кредитовании, наборе персонала, при пересылке информационных сообщений в социальных сетях и тому подобных операциях.

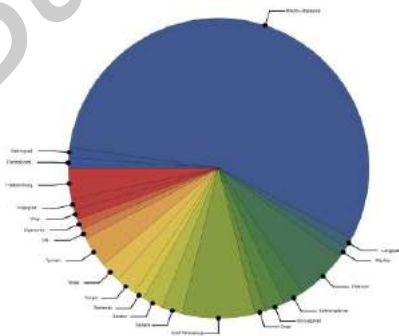


Рисунок 5 – Распределение "друзей" по городам

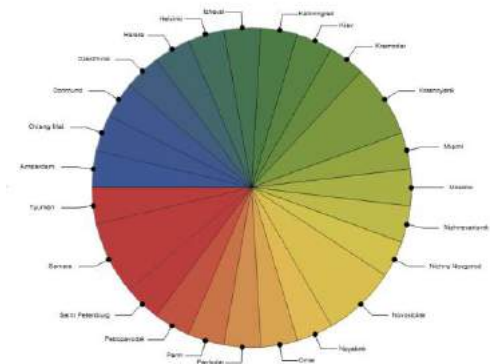


Рисунок 6 – Распределение "друзей" по городам в случае недостоверной учетной записи

3. Использование элементов теории графов в анализе социальных сетей

Одним из наиболее удобных и информативных методов представления информации о связях в социальной сети является изображение их в виде графа. В Wolfram Mathematica реализован значительный функционал по работе с графами, в том числе, масса встроенных функций, позволяющих анализировать структуру графов.

Так, например, чрезвычайно полезным как в образовательном, так и в исследовательском плане является получение графа связей "друзей" пользователя между собой. При этом средства Mathematica позволяют не только указать связи, но и продемонстрировать дополнительные свойства пользователей. Так на полученном графе (Рисунок 7) разным цветом обозначен пол пользователя, а диаметр круга зависит от количества друзей каждого из пользователей. Активность в сети можно использовать для понимания, какие пользователи из нашего окружения могут обладать наибольшим влиянием [Губанов и др., 2010].

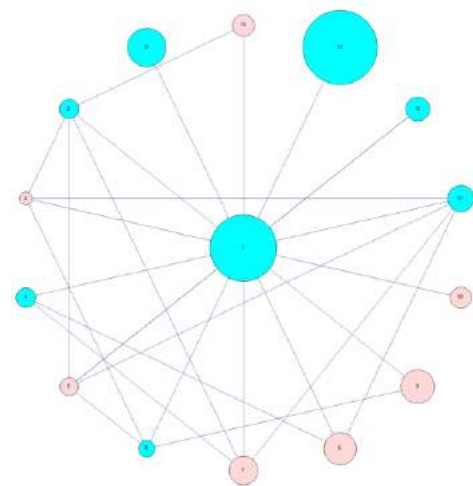


Рисунок 7 – Граф "друзей" пользователя

Используя встроенные функции для обработки графов можно представить полученный выше рисунок в виде "графа сообществ" (community graph), т.е. разделения всех "друзей" пользователя на

группы, имеющие связи между собой и, как следствие, максимальное влияние друг на друга (Рисунок 8).

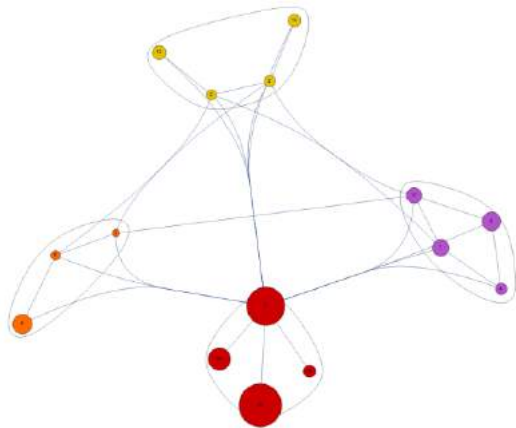


Рисунок 8 – Граф "друзей" пользователя

Заключение

Социальные сети на сегодняшний день прочно вошли в повседневную жизнь людей во всем мире. Изучение сетевых взаимодействий пользователей, а также той информации, которая передается между пользователями представляет собой важную задачу современных информационных технологий.

Wolfram Mathematica как наиболее перспективная и мощная среда для семантического анализа социальных данных позволяет производить разнообразные эксперименты по проверке социологических гипотез. Как следствие, можно утверждать, что изучение и развитие подобной среды является необходимостью для современного исследователя.

Библиографический список

[Губанов и др., 2010] Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Модели информационного влияния и информационного управления в социальных сетях / Губанов Д.А. [и др.]; - ФИЗМАТЛИТ, 2010

[Hanneman et al, 2005] Hanneman, R. A., M. Riddle Introduction to Social Network Methods/ R. A. Hanneman [et al]// University of California, Riverside. - 2005

[Watts, 1999] Watts, D.J. Small Worlds: The Dynamics of Networks between Order and Randomness. Princeton University Press, Princeton, NJ. - 1999

SEMANTIC ANALYSIS OF SOCIAL NETWORKS USING WOLFRAM MATHEMATICA

Tyakunov A.S., Slavsky V.V.

Ugra State University, Khanty-Mansiysk, Russia

atyakunov@mail.ru

slavsky2004@mail.ru

This article examines the use of Wolfram Mathematica for data analysis posted on social networks. First of all, interest is public data, on the basis of which the researcher can check various social hypothesis. The purpose of this work - to show the possibilities of Wolfram Mathematica for large-scale analysis of the

relations of social network users, graphical reporting and analysis of data placed in the public domain.

Introduction

Online social networks, in addition to support functions of communication, exchange of views and information of their members in recent years have increasingly become targets and means of information management and information warfare arena. In the near future, they will inevitably become an essential tool of information influence, including, for the purpose of manipulating the individual, social groups and society as a whole.

Thus, the need for full and qualitative analysis of social interactions through social networks is a major challenge of our time, including from a mathematical point of view. To solve it requires tools to handle requests to social networks in order to obtain clear results of the analysis. One such tool is an Wolfram Mathematica.

Main Part

The mathematical software Mathematica, is known as the most powerful in the world of computing application. Moreover, Mathematica provides a single platform for research and calculations, which are reflected in the live interactive publishing quality documents. Using a powerful computing unit and user-friendly interface, users can easily Mathematica to solve the time-consuming tasks and to understand the complex concepts using all available arsenal of hardware. Of particular interest to the researchers is, usually, the study and analysis of information about the users, which is associated with the node. Analysis of the data on the "Friends" of the user can be used for verification of various hypotheses. One of the most convenient and informative methods of presenting information about connections in the social network is an image of a graph. The Wolfram Mathematica implemented significant functionality for working with graphs, including plenty of built-in features to help you analyze the structure of graphs.

Conclusion

Social networks today have become part of the daily lives of people around the world. The study of network user interactions, as well as the information that is transferred between users is an important task of modern information technologies. Mathematica as the most promising and powerful environment for semantic analysis of social data allows for a variety of experiments to test sociological hypotheses. As a consequence, it can be argued that the study and the development of such an environment is a necessity for the researcher.



УДК 004.912

ИСПОЛЬЗОВАНИЕ СОЦИАЛЬНЫХ СЕТЕЙ ДЛЯ ПОИСКА КОМПЕТЕНТНЫХ ГРУПП ЛЮДЕЙ

Стратнев П.Ю.

Югорский государственный университет, г. Ханты-Мансийск, Россия

Pavel.Stratnev@gmail.com

В статье рассмотрен метод поиска определённых групп пользователей, имеющих компетентность в некоторой области деятельности.

Ключевые слова: социальные сети, классификация, компетентность, алгоритм.

Введение

За последние несколько лет уровень информатизации общества резко возрос. Одним из аспектов этого роста является развитие такого информационного ресурса как социальные сети. Подобные ресурсы представляют собой автоматизированную социальную среду для обеспечения коммуникации как отдельных, так и групп пользователей, объединёнными общими интересами. Таким образом, данные Web-сайты формируются за счёт пользовательского контента, и могут по праву считаться удобным средством обмена информацией. Имея в своём составе богатую информационную базу, а также, учитывая тот факт, что количество пользователей в этих сетях увеличивается с огромной скоростью [Boyd et al., 2007], такие сервисы могут использоваться в качестве источника актуальных и уникальных данных.

На сегодняшний день, в условиях сложившейся демографической ситуации обусловленной кризисом 90-х годов, имеет место проблема обеспечения организаций достаточным количеством наиболее подготовленного контингента. ПрофорIENTATIONная работа в школах направлена на решение данной проблемы, но в масштабах населения России требуются дополнительные инструменты и возможности разрешения этой задачи.

Данная статья посвящена поиску компетентных групп людей на основе данных, полученных из социальной сети.

1. Система анализа данных

1.1. Сбор данных из социальной сети

Интерфейс социальной сети является источником данных реального времени и предназначен для просмотра и взаимодействия со страницами социальной сети в веб-браузере. Для использования данных пользователей в специализированных приложениях ресурс предоставляет определённый набор функции (API) для получения данных из своей информационной базы. Поскольку сценарии использования интерфейсов социальных сетей не предполагают автоматического сбора данных множества пользователей, то возникает ряд проблем:

- *приватность данных* – чаще всего доступ к данным пользователей разрешён только для зарегистрированных и авторизованных участников сети, что требует поддержки эмуляции пользовательской сессии с помощью специальных учетных записей (*аккаунтов*);
- *слабая структурированность данных* – во многих случаях API социальных сетей имеют ограниченный функционал, что требует поддержки получения с помощью пользовательского веб-интерфейса статистических копий HTML-страниц, корректной обработки их динамической части (включая исполнение асинхронных запросов к серверу социальной сети), извлечения нужных данных с помощью алгоритма и/или шаблона и построения их структурированного представления, удобного для дальнейшей автоматической обработки;
- *ограничение доступа и блокировка* – с целью предотвращения несанкционированного автоматического сбора данных и ограничения нагрузки на инфраструктуру сервиса социальной

сети администраторы сервисов зачастую вводят явные или скрытые ограничения на допустимое количество запросов от одного пользовательского аккаунта и/или IP-адреса в единицу времени, что требует учёта количества посылаемых запросов, а также поддержки динамической ротации используемых для сбора данных пользовательских аккаунтов и IP-адресов;

- *размерность данных* обуславливает необходимость в параллельном методе сбора данных, а также в методах получения репрезентативной выборки пользователей социальной сети.

В связи с необходимостью постоянного получения больших наборов данных из социальной сети, была разработана программа реализации этой функции.

Созданный инструмент поддерживает скачивание данных из социальной сети «ВКонтакте», пользующуюся огромной популярностью в русском сегменте сети Интернет. Загрузка данных происходит постранично, с последующей обработкой элементов внутренней разметки страницы. Та часть элементов, получение значений которых, реализовано через API сервиса, автоматически структурируется и вносится в локальную базу данных приложения. Кроме функций скачивания, приложение реализует механизм автоматического выбора учетной записи социальной сети для каждого запроса, а также поддержка прокси-соединений. Это обеспечивает устойчивость к блокировкам по IP-адресам и учетным записям.

Для оценки производительности инструмента были проведены эксперименты, в которых скачивались и записывались в базу данных профили пользователей социальной сети «ВКонтакте». В среднем производительность достигала 250 пользователей в час.

1.2. Предварительная обработка данных

Так как вся основная информация размещённая на веб-ресурсе представлена в виде HTML-страницы, то логично предположить что большая часть данных является обычным текстом. Для дальнейшего использования информационной базы социальной сети в рамках данного проекта, требуется предварительная обработка собранных данных, которая в первую очередь включает в себя непосредственное выделение информации, признаков и параметров со страницы учетной записи. Это позволит внести некоторую структурированность в систему данных.

Модель «Множество слов» (bag-of-words) – упрощающее предположение, использующееся в обработке естественного языка и поиске информации (information retrieval). В этой модели, текст (предложение или документ) представляется как неупорядоченный набор слов, без учета грамматики и порядка слов в частности. Для этого

необходимо убрать HTML-теги, знаки препинания, стоп-слова, все слова привести к нижнему регистру [Lee et al., 2010].

Для определения базиса пространства требуется получить все основы слов в используемом наборе текста. Процесс нахождения основы слова для заданного исходного слова называется *стемминг*.

Существует много алгоритмов решающих поставленную задачу. В результате анализа был выбран стеммер Портера. Главным плюсом этого метода заключается в том, что он не использует никаких словарей и выделение основы осуществляется путём преобразования слова согласно определённым правилам [Porter, 1980]. К тому же, алгоритм поддерживает работу с русским языком.

1.3. Классификация профилей пользователей

Задача классификации представляет собой задачу отнесения образца к одному из, как минимум, двух попарно не пересекающихся множеств.

Пусть X – множество описаний объектов, Y – множество наименований (номеров) классов.

$$y: X \rightarrow Y. \quad (1)$$

Существует неизвестная целевая зависимость – отображение, значение которой известны только на объектах конечной обучающей выборки X^m , по которой производится настройка модели.

$$X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}. \quad (2)$$

Требуется построить алгоритм a , способный классифицировать произвольный объект $x \in X$.

Пусть дан набор пользователей n пользователей P .

$$P = \{p_1, p_2, \dots, p_n\}. \quad (3)$$

Каждый пользователь p_i имеет профиль u_i и связанный с ним контент. Профиль представляет собой страницу, которая создаётся и контролируется пользователем.

Задача заключается в определении компетентности пользователя p_i в математике, русском языке, иностранном языке, литературе и т.п. (область знаний зависит от обучающей выборки) с помощью классификатора c , используя u_i .

$$c: u_i \rightarrow \{1,0\}. \quad (4)$$

Классификатор c даёт прогноз, является ли пользователь p_i компетентным кадром. Для

построения классификатора необходимо определить, какие параметры влияют на принятие решения о том, принадлежит ли объект классу. Для этого необходимо иметь достаточно размерность пространства признаков (количество параметров входного вектора).

Точность классификатора можно оценить с помощью тестовой выборки с заранее известными данными о принадлежности элементов к классу. Оценку качества классификации, сделанную по тестовой выборке, можно применить для выбора наилучшей модели или дальнейшей модификации.

1.4. Структура приложения

Эксперимент требовал создания дополнительного программного обеспечения реализующего алгоритм классификации, и поэтому была разработана архитектура системы, представленная на рис. 1, состоящая из 3 основных узлов, каждый из которых выполняет соответствующую функцию.

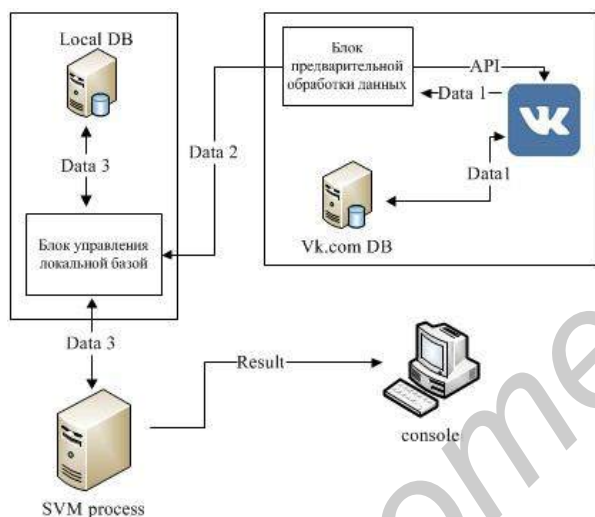


Рисунок 1 – Общая структура приложения

Любая социальная сети (в данном случае - «ВКонтакте») имеет в своём составе закрытую информационную базу, хранящую все профили своих пользователей. Так как обращение к базе данных напрямую не доступно, то параметры необходимые для непосредственной классификации, или информация необходимая для вычисления определённых атрибутов выделяет из многопользовательской системы с помощью API, которое предоставляет сервис vk.com.

Блок предварительной обработки данных включает в себя модуль сбора информации и реализует предварительную обработку данных. В качестве входных параметров используется поток данных data1, который содержит данные в формате целевого сервиса. Обработка параметров включает вычисление некоторых атрибутов, преобразование и передачи данных в блок управления локальной базой (data2).

Блок управления локальной базой берёт на себя функции работы с главной базой данных приложения, процедуру преобразования данных к векторной форме, необходимой для работы алгоритма классификации и непосредственно передачи преобразованных данных в узел SVM process (data3). Самостоятельность модуля необходима для обеспечения расширяемости системы в целом.

Важной частью системы является блок SVM process, в котором представлен алгоритм машинного обучения, который может работать с любыми данными из локальной базы данных. Модуль не зависит от набора данных data1, что делает его универсальным, предоставляя возможности по обработке данных из других многопользовательских сетей. Необходимым и достаточным условием станет лишь корректировка или добавление блока предварительной обработки данных для соответствующей системы.

Результаты работы приложения выводят в консоль.

1.5. Метод исследования

Анализ показал, что среди методов машинного обучения высокие перспективы имеет метод опорных векторов [Воронцов, 2007]. Этот метод позволяет добиться высокого качества в области классификации.

Метод опорных векторов относится к бинарным классификаторам, которые определяют только принадлежность (или не принадлежность) объекта к классу, но не сам класс. Иными словами полностью соответствуют выражению (4). Суть самого метода заключается в поиске гиперплоскости разделяющей два класса точек в n-мерном пространстве. Обучающая выборка задаёт целевую зависимость, значения которой заранее известны. Требуется построить алгоритм, аппроксимирующий целевую зависимость на всём пространстве.

Для построения классификатора требуется определение пространства признаков, по которым будет проходить классификация пользователей. Профиль пользователя будет представлен в качестве вектора большой размерности. Элементы вектора формируются путём предварительной обработки данных, полученных со страницы пользователя социальной сети.

Исследования проводились с использованием компьютера на базе Intel Core i5 с 4 Гб оперативной памяти. В качестве основного инструментария для проведения классификации использовался язык программирования python и библиотека libsvm реализующая работу метода. Благодаря этим программным средствам поставленная задача свелась к формированию обучающей выборки (настройка классификатора) и непосредственно классификация контрольной выборки.

1.5.1. Настройка классификатора

Обучение классификатора будет строиться на выборке, в которую войдёт часть профилей пользователей, которые обучаются (или обучались) в Институте (НОЦ) систем управления и информационных технологий Югорского государственного университета. В данном случае принадлежность к классу будет определять статус «студент института». Таким образом, мы получаем выборку вида:

$$V^i = \{(v_1,1), \dots, (v_i,1)\} \quad (5)$$

где v_i - вектор, элементы которого являются предварительно обработанные данные, полученные из профиля i -го пользователя студента. Предполагается, что 100 профилей будет достаточно для построения модели.

1.5.2. Тестирование точности

После получения настроенного классификатора, оценим его точность на новой выборке людей, по характеру отбора схожей с обучающей выборкой. Отличительной особенностью будет лишь то, что элементы этих множеств не будут пересекаться. Таким образом, мы имеем массив профилей, с заранее известной характеристикой принадлежности к классу.

Из 100 опытов, классификатор спрогнозировал 71 верный случай принадлежности объекта к классу. Точность 71% является допустимой в рамках эксперимента, однако стоит обратить внимание на один немало важный факт. Так как профили в социальной сети контролируются людьми и вся информация полученная из таких профилей является собственноручно заполненной, есть вероятность получить погрешности при построения модели, ввиду возможным нежеланием пользователей указывать достоверную информацию. Это, пожалуй, самый важный недостаток в использовании информационной базы многопользовательской сети.

2. Перспективы развития системы

Первоначальной идеей создания системы являлась поддержка принятия решения в приёмных кампаниях высших учебных заведений. Ввиду того что эта область применения довольно требовательна к точности прогноза, в настоящее время использовать систему не представляется возможным. Достигнув точности классификации в 90%, возможно, этот метод станет востребованным.

В целом, системы классификации решают огромное число задач статистического характера. В частности, адаптация предложенного метода классификации может решать не только задачу определения компетентности пользователей социальной сети, но так же может помочь получить результаты в анализе тональности сообщений пользователей и/или сообществ и выделение

необходимых сегментов пользователей. Кроме того, оперируя уникальными данными, подобные проекты найдут применение и в сфере разработок искусственного интеллекта.

3. Заключение

После своего появления социальные сети развивались в основном количественным путём, охватывая всё больше пользователей, сегодня же социальные сети переходят в стадию качественного развития, изобретая всё новые инструменты взаимодействия с пользователями. Поэтому технологическое развитие станет необходимым условием для выживания в конкурентной среде. Развитие технологий будет многогранным, но даже сейчас можно с уверенностью заявить развитие инструментов работы с контентом станет одной из ключевых тенденций [Семенов, 2011].

Представленный подход поддерживает расширяемость в плане количества источников информации. Добавление дополнительных блоков обработки данных положительно скажется на точности классификации пользователей. Кроме того, точность системы может быть увеличена путём преобразования данных к единому виду для корректного использования с другими методами машинного обучения.

В данной статье предложен метод обработки данных из социальной сети с целью получения компетентных групп пользователей. Рассмотрена архитектура системы исследования и классификации пользователей. Изложенная идея представляет собой законченный эксперимент, результаты которого были описаны в п.1.4.2.

Библиографический список

- [Boyd et al., 2007] Boyd. D. Social Network Sites: Definition, History, and Scholarship / D. Boyd [et al.] // Journal of Computer-Mediated Communication. – 2007 – Vol.13 №1 – P.210-230.
- [Lee et al., 2010] Uncovering Social Spammers: Social Honeypots+Machine Learning / K. Lee, J. Caverlee, S. Webb // Proceedings of the 33rd Annual ACM SIGIR Conference (SIGIR 2010). – July 2010, Geneva, Switzerland. – Geneva, 2010
- [Porter, 1980] Porter M.F. An Algorithm for Suffix Stripping / Porter M.F. // Program: electronic library and information systems 14 (3). – P.130-137
- [Воронцов, 2007] Воронцов К.В. Лекции по методу опорных векторов [Электронный ресурс] / К. В. Воронцов // Режим доступа: <http://www.ccas.ru/voron/download/SVM.pdf>
- [Семенов, 2011] Семенов Н. Все о социальных сетях. Перспективы развития [Электронный ресурс] / Н. Семенов // Режим доступа: <http://secl.com.ua/article-vse-o-socialnyh-setjah-perspektivy-razvitija.html>

USING SOCIAL NETWORKS FOR FINDING QUALIFIED GROUPS OF PEOPLE

Stratnev P.U.

Ugra State University, Khanty-Mansiysk, Russia

Pavel.stratnev@gmail.com

The article describes the method of searching for specific user groups with expertise in a certain field of activity.



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.432.4

ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ ПОДДЕРЖКИ ЗАПРОСОВ НАЛОГОВЫХ ИНФОРМАЦИЙ

Турмаганбетова Ш.К.

*Евразийский национальный университет имени Л.Н.Гумилева,
г. Астана, Республика Казахстан*

sh_kurmangali@e-mail.address

В рамках диссертационной работы определены основные понятия создаваемой онтологической модели, которая позволит организовать процесс реализации экспертной системы налоговой системы в Республике Казахстан. Таким образом, в рамках данной работы будет предложены три типа моделей, основанных на онтологии, которые структурируют и организуют информацию для поддержки информационного доступа. Извлеченная информация разработана для навигации информационного поиска пользователей, но не ограничения возможностей пользователей.

Ключевые слова: онтологическая модель, SemanticWeb, поиск налоговой информации.

Введение

Информационный портал - это потенциальная важная техника извлечения текстовой информации из базы знаний. Однако, задача информационного поиска сложна, и легко потеряться в сложной сети узлов, представляющих разные понятия. В этом проекте мы рассмотрим, как возможность исследования, может быть поддержана знанием, основанным на онтологии. Упомянутые выше неудобства избегаются, предоставлением услуг исследовательским задачам пользователей, в частности, если они не являются экспертами в этой области. Мы предлагаем информационным ищущим и исследователям, способ преследования информационных объектов, связанных с их задачами и применения действий к задаче для достижения первоначальных целей и намерений. Были предложены три типа моделей основанных на онтологии, которые структурируют и организуют информацию для поддержки информационного доступа. Извлеченная информация разработана для навигации информационного поиска пользователей, но не ограничения возможностей пользователей. Кроме того, мы обсуждаем обеспечение действий к задачам пользователей в целях поддержания повторного использования.

1. Назначение онтологии

Онтология является системой понятий некоторой предметной области. Назначение

словаря онтологии - предоставить терминологию для описания знаний о конкретной области.

Онтология структурирует понятия словаря. В самом простом случае онтология описывает иерархию связанных представлений. В более сложных случаях добавляются подходящие аксиомы (или ограничения), выражающие связи между представлениями и ограничивающие их интерпретацию.

Как правило, при описании знаний представления используются для обозначения типовых множеств объектов, обладающих общими признаками, или множеств объектов, существующих в данной предметной области, но не являющихся типовыми. Типовые множества объектов организуются в классы объектов, каждый из которых характеризуется набором типовых свойств (или атрибутами класса). Допустимая область применения знаний задаётся посредством аксиом или ограничений. Перечисленные составляющие - классы, атрибуты, отношения, аксиомы/ограничения - рассматриваются как элементы онтологии.

Онтология является составной частью базы знаний, которая, помимо онтологии, включает в себя логику (или правила вывода), а также может содержать неструктурируемую (или неформализованную) информацию, выраженную средствами естественного языка.

Онтологии позволяют оригинально построить web-ресурсы на основе формальных описаний тех областей знаний, что составляют суть контента ресурса.

Факт создания информационного портала для поддержки исследовательской (браузерной) функции может быть оспорим. Однако, поиск может оказаться задачей отнимающей много времени, как только число потенциально соответствующих страниц становится больше. Исследующие пользователи могут испытывать затруднения в адресации своих потребностей, если они не эксперты в этой области, например, просматривая информацию о налогообложении. При таких обстоятельствах пользователь может захотеть организовать понятия в кластер и обусловить поперечные связанные ссылки, чтобы выполнить определенную задачу и исследовать нужную информацию. Поэтому, требуется, чтобы пользователь нашел нужную информацию в пределах информационной структуры.

Однако, пользователям трудно сгруппировать понятия, если они не эксперты в интересующей области. Один из самых часто возникающих проблем у пользователей является сомнительная информация, встречающаяся в многократных источниках, использующая похожую терминологию. Исследование показало, что вероятность совпадения терминов в домене и словарного запаса пользователя равна <0.2 (Furnasetal, 1987).

Чтобы рассмотреть эти проблемы по отдельности, была предпринята попытка моделирования поведения исследования, сопоставляя исследовательские задачи пользователей с необходимой информацией, требуемой для выполнения этих задач. Поэтому извлеченная информация может быть разработана так, чтобы она могла вести пользователей и избежать путаницы в сложной сети узлов, представляющих понятия. К примеру, если пользователь исследует информацию о «налоге», необходимый вход, требуемый для такой информации, может быть поддержан, показывая ставку налога, назначения налога, стоимость и требуемое время для оплаты налога. Если пользователь исследует информацию «о земельном налогу», то необходимые документы и условия подачи заявки могут предоставить пользователям необходимую информацию о назначении и периодичности налога, таким образом, осуществляя поисковую задачу.

Поскольку Сеть экспоненциально расширяется в размере, недостатки в организации затрудняют эффективный подбор знания из Сети. Для того, чтобы обеспечить, средства навигации, в информации исследовательских задач пользователей в Сети, важно обеспечить семантическую структуру информационной окружающей среды. Онтология была предложена для использования в различных областях: системы, основанные на знаниях, информационный поиск, стандартизация, и т.д. Онтология была рассмотрена как общее организационное устройство для спецификации концептуального знания. Онтология

предоставляет базу для ясного раскрытия понятия во всех отношениях. Высшие категории могут быть полезным гидом в разложении сложной информационной проблемы и в формировании доступного и эффективного метода доступа к информации. Поскольку потребность в использовании больше чем одного типа онтологии была вынужденной из-за возрастающего размещения информации, мы предложили три типа моделей, основанных на онтологии для классификации и поисковых целей.

2. Модели, основанные на онтологии

Наши усилия были сосредоточены на использовании онтологии для поддержания информационных потребностей в поиске информации о правительстве. Сценарии, возникающие в приложениях, мотивируют развитие онтологии. Большинство информационных порталов интересуется тремя типами информации, исследуя правительственное содержание: организационная структура, бизнес функции, и деятельность человека. В этом проекте мы сконцентрируемся на определении организационной структуры и предпринимательской деятельности большинства правительственных организаций.

Два типа пользователей вовлечены в развитие компьютерной системы, Интранета: Инженеры Онтологии/Менеджеры Отчетов и конечные пользователи. Главная ответственность Инженеров Онтологии/Менеджеров Отчетов состоит в том, чтобы обеспечить контроль и последовательность в словарном запасе, используемого для классификации и индексирования отчетов. Классификация требует понимания контекста и содержания отчета, знания существующих отчетов в системе и индексация, которая включает прикрепление одного или более поисковых слов, к каждому записанному отчету, такому как файлу печатного или электронного документа. Это облегчает непринужденность поиска конечных пользователей и извлечения отчетов. Основанные на информационных потребностях, три типа моделей онтологии предложены: организационная онтология, онтология действий домена и пользовательская онтология действий. Организационные методы и методы деятельности домена предназначены для индексирования изаголовков, тогда как пользовательская онтология деятельности облегчает поиск и извлечение информации.

2.1. Организационная онтология

Организационная онтология предоставляет семантическую информацию правительственной структуры. Сложная структура часто решается разложением структур в иерархические единицы в компьютерной системе. Рисунок 1 показывает иерархическое дерево организационной онтологии правительства. Дерево представляет понятия

предприятий. Например, правительство состоит из государственных и местных департаментов. Государственные департаменты включают в себя ЗАКОН, Республики Казахстан и нормативные акты и т.д. Местные департаменты включают в себя Генерального прокурора Департамента, Министерство обороны, Министерство здравоохранения и помощи семьям, и т.д. Каждое предприятие связывается с соответствующими отношениями и признаками. Например, определение налог – ставка налога, и его функция определяется термином «назначается».

Организационная онтология в состоянии ответить – вопросы, базируемые на сложное знание, такие как нахождение «нарушение налогоплательщика». Поиск в существующей сети идентифицирует тысячи страниц о налогоплательщике в налогах на слово «нарушение налогоплательщика». В этом примере онтология включала бы такой признак как «фамилия», классификацию как «человек», и связь со словом «налогоплательщик».

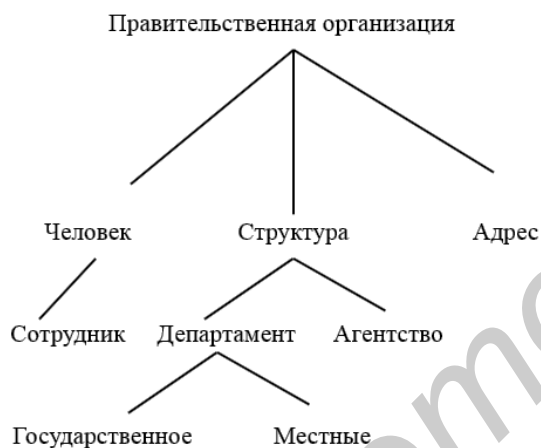


Рисунок 1 – Иерархическое дерево организационной онтологии правительства

2.2. Онтология действий домена

Онтология действий домена разработана для организации и структурирования информационной функции и действия в домене. В отличие от традиционной классификации библиотек, где книги классифицированы согласно предметам, онтология действий домена предлагает иерархическую структуру для классификации отчетов согласно информационным функциям и действиям, зарегистрированные в отчетах для классификации и индексации задач. Таким образом, последовательность информационной действий может сопровождаться группой запросов, которые были классифицированы вместе.

Развитие онтологии действий домена основано на CSIROKeywordAAA (CSIRO 1996), который является алфавитным списком терминов.

Роль онтологии действий домена - это определение запроса по их отношениями с

деятельностью, которую они документируют. Поэтому, отчеты, классифицированные и индексированные при помощи онтологии действий домена, будут иметь значение и будут размещены в пределах контекста, в котором они были созданы. Это также означает, что все файлы на одной функции или действиях группируются вместе в индексе.

2.3. Пользовательская онтология действий

Пользовательская онтология действий связана с информационными действиями поиска, выполненными конечными пользователями. Задача информационного поиска сложна. С точки зрения анализа основанного на знания, мы должны рассмотреть, каким пользователям нужно знать об информационных объектах и действиях, вовлеченных в информационную задачу поиска, и как это знание организовано. Действия пользователя обычно могут быть описаны, выполнением действий задачи информационными объектами, связанными с информационными потребностями. Например, пользователь ищет информацию в музыкальной библиотеке. Он или она выполняет действия, такие как вход в коллекцию, поиск индекса к полке и слушанию песни.

Развитие пользовательской онтологии действий начинается с построения таксономической классификации задачи и информационных объектов. Знание задачи включает словарный запас для процесса выполнения действий, как, вход, поиск и т.д. Рисунок 2 показывает концептуальную модель для пользовательской онтологии действий. Задача - это информационные действия поиска пользователей. Родовой глагол или фразовые глаголы используются для представления словарного запаса действия задачи, которое независимо от домена и может быть составлен вручную.

Сборник родовых глаголов используется, чтобы описать информационные задачи поиска пользователя, такие как:

- Доступ к: достижению, использованию или приближению к чему-то.
- Обратиться: формально попросить что-то.
- Иммигрировать: стать поселенцем, не туристом или посетителем.
- Дом: поместить (заявление, и т.д.) надлежащими властями.
- Поиск: искать, попытаться найти что-то.

Информационный объект представляет ряд объектов, связанных с информацией. С точки зрения конечного пользователя, они могут быть незнакомы с организационной структурой и терминологией, используемой для бизнес функций и действий домена. Поэтому, информационные объекты включают в себя высокоуровневое описание организационной онтологии и онтологии действий домена, например, они только включают

ключевые слова и описатели действия в онтологии действий домена. Список существительных или фразовых существительных используется, чтобы описать информационные объект, таких как:

- Налог (например, налог на печать, земельный налог, дебаты, налог, и т.д.)

Комбинация родового глагола и существительного, как «глагол + существительное» может быть предпосылкой на процесс решения общей проблемы. Такая комбинация облегчает процесс информационного поиска.

Комбинация задачи и информационных объектов относится к поисковым действиям пользователя, как:

- Налог Заявления (например, налог на печать, земельный налог, налог дебетов, и т.д.)

Вышеупомянутые подчеркивающие линии указывают на родовые существительные информационного объекта.



Рисунок 2 – Концептуальная модель онтологии действий пользователя

Заключение

Мы предложили два типа моделей онтологии, которые структурируют информацию путем поддержки средств навигации при доступе к информации. Мы показали, что исследовательский просмотр может быть объединен с основанным на онтологии знанием во избежание потери в киберпространстве, в особенности, если пользователи не являются экспертом в домене. Кроме того, мы рассмотрели проведение действий задачи пользователей, чтобы поддержать информационное повторное использование.

В рамках нашей будущей работы, важное рассмотрение информационной навигации явилось вызовом, поставленным Познавательным Движением. Познание основано на идее, что использование знания, полностью зависит от ситуации. Мы признаем, что вопросы навигации информации должны быть исследованы варьированием ситуаций, таким образом, мы

смогли бы отследить необходимые изменения в онтологических моделях. Это также говорит о том, что мы нуждаемся в системах, которые поддерживали бы, необходимые изменения в моделях онтологии, поскольку ситуации изменяются. Необходимо связать модели онтологии к расположенной парадигме познания.

Библиографический список

- [Berners-Lee, 2001] T., Hendler, J., Lassila, O.] The Semantic Web // Scientific American, May 17, 2001.
 [Stewart S.] Karlinsky and Daniel E. O'Leary. Tax-based expert systems: a first principles approach. Expert Systems in [Finance D.E. 1992] O'Leary and P.R. Watkins (Editors) 1992 Elsevier Science Publishers B.V.

ONTOLOGY MODELS FOR SUPPORTING EXPLORATORY INFORMATION NEEDS

Turmaganbetova Sh.K.

Евразийский национальный университет имени Л.Н.Гумилева,

г. Астана, Республика Казахстан

sh_kurmangali@ e-mail.address

Conclusion

We have provided three types of ontology models that structure information in a way to support navigation aids for information access. We have shown that exploratory browsing can be coupled with ontology-based knowledge to avoid getting lost in areas not experts in the domain. In addition, we have discussed the main tenancy of users' task actions to support information reuse purposes.



УДК 004.023

ГЕНЕРАЦИЯ ЗВУКОВ ПО ЦВЕТОВОЙ ГАММЕ ИЗОБРАЖЕНИЯ

Никитин Н.А., Розалиев В.Л. *, Орлова Ю.А. *, Заболеева-Зотова А.В.**

* *Волгоградский государственный технический университет, г. Волгоград, Россия*

nikitin.nikita@outlook.com

vladimir.rozaliev@gmail.com

yulia.orlova@gmail.com

** *Российский фонд фундаментальных исследований, г. Москва, Россия*

zabzot@gmail.com

В работе рассматривается теоретическое описание алгоритма соотнесения цветовых и музыкальных характеристик, базирующегося на светомузыкальной теории Афанасьева, а также схеме соотнесения цветов и тональностей Скрябина. Также описывается программная система генерации звуков по цветовой гамме изображений, которая состоит из модуля анализа изображений и синтеза звуков, а также приводится описание выбранного метода синтеза звуков.

Ключевые слова: светомузыкальная теория; синтез звуков; сэмплинг.

Введение

Не смотря, на все достижения в понимании творческих процессов, создание музыки не может проходить автоматически. Однако не прекращаются попытки установить зависимость между зрительными и музыкальными образами [Заболеева-Зотова и др., 2007]. Передаваемая музыкой и картинными эмоциональность сложно распознаваема [Розалиев и др., 2013]. Сам процесс создания музыки, на данный момент, не поддается четкой формализации, хотя и основывается на строго определённых музыкальных правилах. В данной работе рассматривается теоретическое описание генерации звуков по цветовой гамме изображений, как одно из приближений к автоматизации процесса создания музыки.

В основе работы лежит светомузыкальная теория В.В. Афанасьева [Афанасьев, 2002], а также разработанная русским композитором Скрябиным, схема соотнесения цветов и нот.

1. От цветовых характеристик к музыкальным

Согласно светомузыкальной теории Афанасьева В.В. невозможно навсегда привязать какую-либо ноту к определенному цвету в силу их различной природы, поэтому необходимо связать отношения звуков и цветов в зависимости от того, в какой

плоскости они представлены: мелодия, гармония или тональность. Иначе говоря, одна и та же нота в произведении может быть окрашена разными цветами.

Для начала определим таблицу соотнесения музыкальных и художественных характеристик.

Таблица 1 – Таблица соответствия музыкальных и художественных характеристик

Цветовые характеристики	Музыкальные характеристики
Оттенок (красный, синий, жёлтый...)	Нота (до, до-диез, ре, ре-диез, ми, ми, фа, фа-диез, соль, соль-диез, ля, ля-диез, си)
Цветовая группа (тёплый/холодный)	Музыкальный лад (мажор/минор)
Яркость	Длительность ноты
Насыщенность	Октава ноты

Пусть на вход подаётся некоторое изображение. Формальный алгоритм соотнесения цветовых и музыкальных характеристик можно описать следующими пунктами:

Шаг 1. В начале анализируется все изображение и определяется преимущественный цвет.

Шаг 2. Далее согласно схеме, построенной А.Н. Скрябиным (рисунок 1) и таблице соответствия

музыкальных и художественных характеристик (таблица 1), определяется тональность будущего произведения.

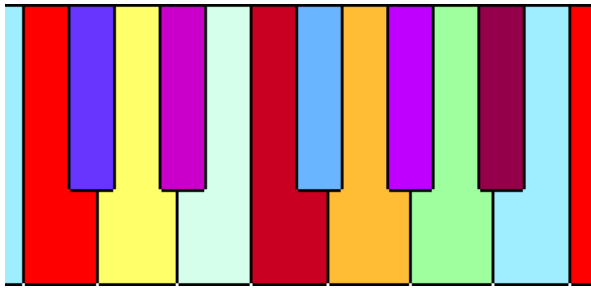


Рисунок 1 - Соответствие цветов и тональностей по Скрябину

Максимальное родство цветов определяется между двумя соседними хроматическими цветами, а степень тонального родства – между звуками, находящимися на расстоянии семи полутонов друг от друга и отражена в квинтовом круге тональностей.

Таким образом, совмещая хроматический цветовой круг и квинтовый круг тональностей, получим необходимую последовательность для модуляций.

Например, пусть в качестве преимущественного цвета был выбран красный. Согласно схеме Скрябина, это соответствует ноте До (рисунок 2).

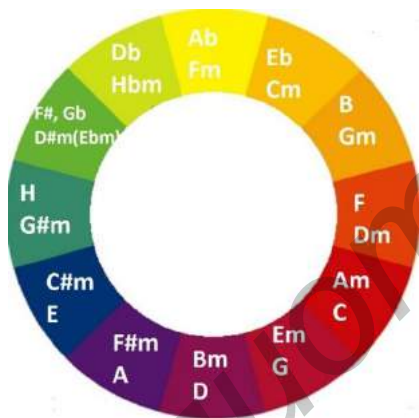


Рисунок 2 - Наложение цветового круга и квинтового круга тональностей

Шаг 3. Затем, сопоставляя хроматическую гамму нашей тоники, с цветовым кругом, получим соответствие нот и цветов для мелодической части будущего произведения (рисунок 3).

Шаг 4. После этого, необходимо определить соответствие цветов и ступеней для построения гармонии произведения. В музыке существуют функциональные отношения (тоника – субдоминанта – доминанта), а в цветоведении этому соответствует принцип дополнительности цветов (красный – синий – желтый). То есть, главными цветами при принятом основном, например, красном, являются дополнительные – желтый и синий, главными трезвучиями в музыке – тоническое, субдоминантовое и доминантовое.

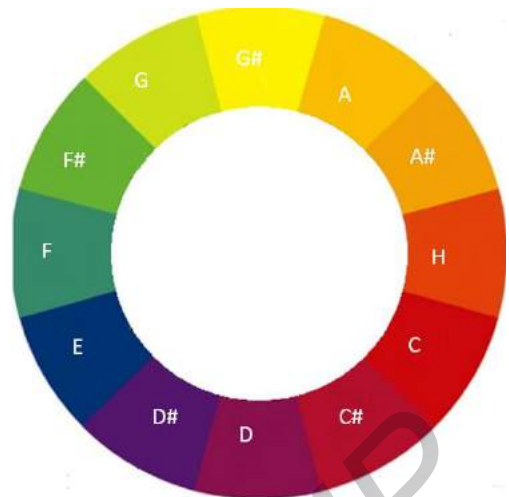


Рисунок 3 - Наложение хроматической гаммы и цветового круга

Таким образом, получим соответствие цветов и ступеней для построения гармонии произведения (рисунок 4).

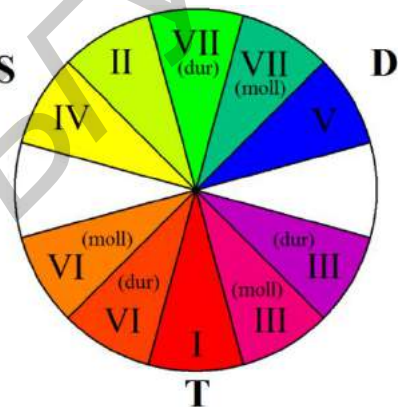


Рисунок 4 - Гармоническое соответствие цветов и ступеней для построения гармонии

Шаг 5. На предыдущих пунктах получили тональность будущего произведения, определили необходимую последовательность для модуляций, выявили соответствие цветов и нот применительно для мелодического сопровождения, а также получили гармонию.

Затем необходимо проанализировать всё изображение, оценивая каждый пиксель (или группу пикселей), находя для него характеристики и сопоставляя их музыкальным.

Таким образом получим набор нот, с соответствующими им характеристиками (см. таблица 1), идущих последовательно друг за другом.

Шаг 6. Затем данный набор необходимо перемешать таким образом, чтобы не было большого количества подряд идущих одинаковых нот, для повышения мелодичности композиции. Также в данном наборе нот необходимо, согласно правилам разрешения, преобразовать те ноты, которые не входят в тональность.

Шаг 7. И заключительным этапом алгоритма является разбиение полученной последовательности

нот на такты, согласно их длительностям, и расставление гармонии.

2. Описание метода синтеза звуков

В процессе исследования методов синтеза звука, были рассмотрены четыре наиболее популярных метода синтеза звуков: аддитивный синтез, FM – синтез, фазовая модуляция и сэмплинг.

Аддитивный синтез очень сложен для реализации, из-за необходимости отдельного контроля громкости и высоты каждой гармонии, которых даже несложный тембр насчитывает десятки.

FM – синтез хорошо применим для синтеза звука ударных инструментов, синтез же остальных музыкальных инструментов звучит слишком искусственно. Главный недостаток FM-синтеза — неспособность при его помощи полноценно имитировать акустические инструменты.

Фазовая модуляция даёт достаточно хороший звук, но сильно ограничена, поэтому редко используется на практике.

Сэмплинг применяется в большинстве современных синтезаторов, так как даёт наиболее реалистичный звук и достаточно прост в реализации.

Каждый из методов обладает своими достоинствами и недостатками, однако наиболее подходящим методом для генерации звуков по цветовой гамме изображений был выбран Сэмплинг. Данный метод даёт наиболее реалистичное звучание инструментов, что является важной характеристикой для программы, также данный метод относительно прост в реализации. Недостатком Сэмплинга является ограниченность метода, однако в рамках реализации проекта - это не существенно, так как для нужд программы не требуется больших возможностей изменения готовых пресетов.

3. Описание программы

Алгоритм описанный в пункте 1 является лишь частью, хоть и основной, программы для генерации звуков по цветовой гамме изображений. Создаваемая программа многокомпонентная и будет расширена в дальнейшем [Заболеева-Зотова и др., 2013].

На самом деле, второй важной частью программы является модуль синтеза звуков, опирающийся на метод, выбранный и описанный в пункте 2.

Таким образом, для генерации звуков по цветовой гамме изображений была спроектирована программа, состоящая, в упрощённом виде, из двух модулей: модуль анализа изображения, реализующий в том числе и алгоритм описанный в пункте 2, и модуль синтеза звуков, опирающийся на выбранный метод синтеза (рисунок 5).



Рисунок 5 - Упрощённая архитектура программы

На вход программы подаётся изображение, затем модуль проводит анализ изображения, используя библиотеку OpenCV, и согласно алгоритму, находит соответствие музыкальных и цветовых характеристик. На выходе первого модуля xml файл, или, точнее сказать, текст в xml формате, содержащий формализованное описание музыкальной композиции.

XML файл был введён для того, чтобы упростить взаимодействие двух модулей. Второму модулю, модулю синтеза звуков, необходимо, путём анализа xml файла, сгенерировать последовательность звуков, записать их в аудио файл, и воспроизвести результат пользователю.

4. Описание структуры xml файла

Для взаимодействия двух модулей была разработана собственная структура xml файла, содержащая теги:

- *tonality* - тэг, обозначающий тональность композиции.

Пример: `<tonality>d_minor</tonality>`. Вместо *d_minor* может быть любая другая тональность (*c_minor*, *g_sharp_major* и др.).

- *harmony* - тэг, содержащий описание гармонии. Внутри этого тэга находится тэг *chord*

- *chord* - тэг, внутри которого находятся тэги *type*, *chord_name* и *mode*. Свойством тэга является *duration*, обозначающее длительность аккорда.

- *type* – тэг, важной частью которого является свойство *value*, обозначающее тип аккорда – стандартный (*standard*) или собственный (*own*)

- *chord_name* – тэг, обозначающий для стандартных аккордов, название ноты тональности (с, d, e, f, g, a, h и др.)

- *mode* – тэг, обозначающий музыкальный лад аккорда (*major* или *minor*).

- В случае, если тип аккорда *own*, то вместо

тэгов *type*, *chord_name* и *mode*, внутри тэга *chord* пишется тэг *notes*, в свойстве *value* которого, пишутся ноты и длительности создаваемого аккорда, идущие через запятую, пары разделены точкой с запятой.

Пример: `<notes value="f,2;c,3;f,3;g,3;c,4"/>`

- После закрытия тэга *chord* следует тэг *melody*, обозначающий мелодическую часть композиции. Этот тэг, также как и тэг *chord* требует явного закрытия тэгом `</chord>` и `</melody>` соответственно.

- Внутри тэга *melody* расположен тэг *note*, *note_name* и *octave*. Первый содержит свойство *duration*, обозначающее длительность ноты, второй содержит свойство *value*, которое обозначает имя аккорда (с, d, e, f, g, a, h и др.).

- Наконец третий тэг *octave* имеет свойство *value*, обозначающее октаву ноты.

Таким образом, рассмотренная выше структура, позволяет описать мелодию и гармонию произведения, однозначно определить каждый аккорд и ноту в произведении, а также произвести удобный обмен информацией между двумя модулями системы.

Заключение

В данной работе рассмотрен теоретической алгоритм перехода, от цветовых характеристик к музыкальным. В основу алгоритма положена светомузыкальная теория В.В. Афанасьева [Афанасьев, 2002], а также разработанная великим русским композитором Скрябиным, схема соотношения цветов и нот (рисунок 1).

Также рассмотрена структура программной системы, осуществляющая генерацию звуков по цветовой гамме изображений, а также подробно рассмотрен xml файл, для обмена информацией между модулями системы.

Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов №15-47-02149, 15-07-06322, 15-37-70014, 16-07-00453.

Библиографический список

[Афанасьев, 2002] Афанасьев В.В. Световой музыкальный строй. Элементарная теория аудиовизуальных стимулов/В.В. Афанасьев// М. : Издательство «Музыка», 2002.

[Заболеева-Зотова и др., 2007] Транслятор графического изображения в музыкальное произведение/ Заболеева-Зотова А.В., Лягин И.Д.// Известия Волгоградского государственного университета: межвуз. сб. науч. ст. №2(28)/ ВолгГТУ. - Волгоград, 2007.-120 с. [Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. Вып.2], С.36-39.

[Заболеева-Зотова и др., 2013] Formalization of initial stage of designing multi-component software / Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л., Фоменков С.А., Петровский А.Б. // Multi Conference on Computer Science and Information Systems 2013 (Prague, Czech Republic, July 23-26, 2013) : Proceedings of the IADIS International Conference Intelligent Systems and Agents 2013 / IADIS (International Association for Development of the Information Society). – [Prague], 2013. – P. 107-111.

[Розалиев и др., 2013] Розалиев, В.Л. Methods and Models for Identifying Human Emotions by Recognition Gestures and Motion / Розалиев В.Л., Заболеева-Зотова А.В. // The 2013 2nd International Symposium on Computer, Communication, Control and Automation

3CA 2013, December 1-2, 2013, Singapore : Papers. – [Amsterdam – Beijing – Paris] : Atlantis Press, 2013. – P. 67-71.

SOUND GENERATION BASED ON IMAGE COLOR SPECTRUM

Nikitin N.A. *, Rozaliev V.L. *, Orlova Yu.A. *, Zaboleeva-Zotova A.V. **

*Volgograd State Technical University,
Volgograd, Russia

nikitin.nikita@outlook.com

vladimir.rozaliev@gmail.com

yulia.orlova@gmail.com

** Russian Foundation for Basic Research,
Moscow, Russian Federation

zabzot@gmail.com

The document contains a theoretical description of the algorithm the ratio of color and music performance, based on the lightmusic theory by Afanasiev and scheme of reference colors and tonalities by Scriabin. It also describes the software system of sound generation based on image color spectrum, which consists of a module of image analysis and synthesis of sounds, and contains a description of the chosen method of sound synthesis.

Introduction

The problem of this work is the fact that the process of creating music has not been automated now, and the automated sound generation based on image color spectrum is the first approach to automate it. That's why the algorithm of image based sound generation has been developed and described.

Main Part

The algorithm the ratio of color and music performance, based on the lightmusic theory by Afanasiev, is the kernel of image based sound generation and contains of several items.

The first is the determination of priority color of image. The second item is determination tonality according to scheme by Scriabin. The third part is the definition of matching colors and music, based on the combination of the chromatic scale and the color wheel.

The fourth part of the algorithm is the definition of matching colors and music to build harmony, based on the correlation of functional relations in painting and music. The fifth part of the algorithm is to get a set of notes with corresponding characteristics, then the mixing and conversion of the set of notes.

Conclusion

This document contain a theoretical description of the algorithm the ratio of color and music performance, based on the lightmusic theory by Afanasiev.



UDC 004.81

PRINCIPLE OF AUTHENTIC EQUIVALENCE IN PROBLEMS OF ADAPTIVE CONTROLS IN THE CONDITIONS OF APRIORISTIC UNCERTAINTY

M.Z.Babamukhamedova, M.Yu.Doshchanova, F.A.Ergashev

*The Tashkent University of Information Technology,
Tashkent, Republic Uzbekistan*

malikadoshanova@mail.ru

abbs@mail.ru

*The Tashkent State Technical University,
Tashkent, Republic Uzbekistan*

qwerty@mail.ru

In article systems with control which function in the conditions of aprioristic uncertainty with unknown parameters are considered. On objects of control revolting influences with unknown functions of distribution of probability the mathematical description which it is set by means of inequalities operate. Therefore automatic control system engineering by these objects it is necessary to carry out on the basis of the theory of adaptive systems. Problems of adaptive control solve now on the basis of a principle of authentic equivalence.

Keywords: Dual control, aprioristic uncertainty, object of control, adaptive control systems, operating influence, principle of authentic equivalence.

Introduction

Main principles of optimal control in the conditions of aprioristic uncertainty of revolting influences and the processes occurring in object of control, has formulated A.A. Feldbaum also named them dual control. In systems of dual control at formation of optimal operating influence at the present time it is necessary to know all information about past and future values of variables of a condition of object of control and revolting influences.

Principle of authentic equivalence in problems of adaptive control

In the majority of practical problems this information is inaccessible to the designer, but it can be restored to some extent from the analysis of observable target signals of object of control in managerial process. At an enough effective completion of missing data the control system gets optimal properties, or close to the optimal. Such control system should solve following problem:

- To define functions of density of distribution of probability of entrance influences and target variables of object of control;

- To define efficiency indicators (cost function) criteria of an optimality of estimation of variables of a condition and control;

- To calculate optimal estimations of current values of variables of a condition of object of control;

- To define optimum trajectories of transition of variables of a condition of object of control from an initial condition in the set final condition;

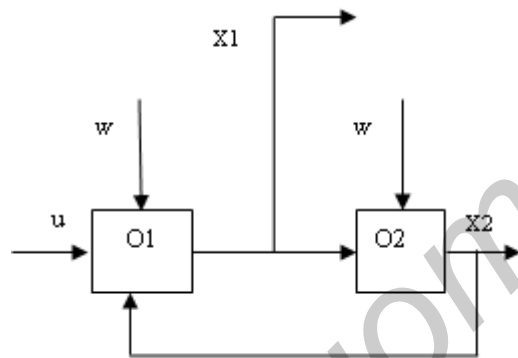
- To define algorithm of formation of operating influences, optimal by the chosen criterion of control;

The problem of dual control with functional adaptation is too difficult for practical realization as there is no the exact mathematical scheme, allowing to consider in algorithm of formation of operating influence of all future values of revolting influences and all future values of variables of a condition of nonlinear objects of control. Dual control can be realized only in neutral systems. Only in these cases optimal values of operating influences at the present situation do not depend on the future values of variables of a condition and the future values of revolting influences. Therefore, in neutral control systems the theorem of division of problems of optimum estimation of variables of a condition and formation of optimal operating influences is fair. The existing theory of adaptive control is based on a principle of authentic equivalence according to which at formation of optimal operating influences in

control systems instead of unknown variables of a condition and parameters, it is possible to use their estimations found in managerial process as a result of the analysis of measured values of target signals of object of control.

Coordinates of optimal trajectories of transition of operated variables in a demanded condition should be set in an explicit form or indirectly – by means of criterion of control efficiency. The control system realising a principle of authentic equivalence, contains object of control, system of measuring devices, the shaper of operating influences and the observer of variables of the condition, included in a feedback chain (Drawing 1).

In control problems the principle of authentic equivalence is carried out by nonlinear objects, if the mathematical description of optimal trajectories of transition of variables of a condition of object of control in the set final condition is known to within parameters, and revolting influences are additive white gausse noise. However in real systems, as a rule, these conditions are not carried out. So, for example, communication systems and radio navigation systems use non-stationary communication channels with unknown parametres and (or) condition variables in which noise with unknown functions of distribution of probability operation.



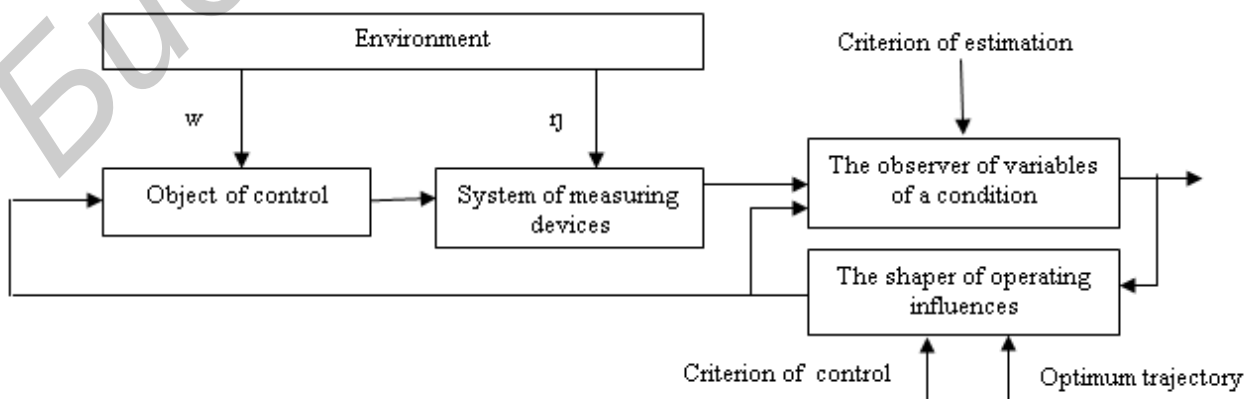
Drawing 2 – Example of multidimensional object of control

Thus, time of correlation of noise comparably in due course correlations of the report of information. The optimal trajectory of transfer of object of control from an initial condition in a demanded final condition, as a

rule, is not known. Restrictions on set of admissible trajectories of transfer of object of control from an initial condition in the demanded final condition, set by means of inequalities are usually known only. The object of control often consists of two subsystems O1 and O2 (Drawing 2), thus operating influences arrive on inputs only subsystems O1, and other part of variables of a condition of object of control is accessible to measurements. Therefore, in adaptive control systems use the approached methods based on regularization of initial statement of a problem, for example, by means of a method regulation A.N.Tikhonov. At correct application the method regulation guarantees reception of the approached decision regulation problems with accuracy, sufficient for application in engineering practice. In the theory of adaptive control there were two approaches to the description aprioristic uncertainty mathematical model of a control system: stochastic and determined (minimax).

Adaptive control in the conditions of aprioristic uncertainty

In connection with development of astronautics and other spheres of a science the new mathematical apparatus of the description of control systems in space of conditions with which help system movement describe system of the ordinary differential equations of the first order has been created. These equations contain functions with which help set processes of change in time of revolting and operating influences. Thus, processes of change in time of revolting influences approximated, aligned with white gausse noise. In the same time the fundamental idea of an optimality of synthesis of a control system has arisen and - the operating influence formed by system of optimum control was successfully developed, should optimise some indicator of quality of control taking into account the restrictions created by mathematical model of a control system. The problem of synthesis of control systems is considered as a problem of optimisation of criterion of quality of control, including all requirements to a control system. It gives the chance to use variation methods, a maximum principle, dynamic programming and the theory of stochastic processes for formation of operating influence.



Drawing 1- The system of adaptive control realizing a principle of authentic equivalence

3. The analysis of observers of variables of a condition of object of management

The correct choice of mathematical model of object of control is the solving condition guaranteeing success in the decision of a problem of the adaptive controls.

This choice should be based as on understanding of the procedure of identification of model, and on the substantial and formalized information on identified object. Quality of the received model can be estimated, in particular, by criterion of an average quadratic errors in which the set of design variables includes model structure. In stochastic problems of control usually use model of dynamics of object of control in space of conditions. The method of variables of a condition is based on concept a system condition, mathematical definition of which is given Zadeh L.A.. A condition of object of control at the moment of time describe set of the phase variables connected by it for entrance signals known functional dependences and characterising behaviour of object of control in the future provided that entrance signals and an object condition at time starting point are known. If in separate elements of object of control there are unknown processes, these processes describe that by means of numbers of Fure, polynoms, B - splines, vejvlet-numbers and others base functions with the unknown parametres which are subject to identification in control process.

Revolting influences consider colour gausses as noise with known likelihood characteristics. It is supposed that these revolting influences can be received on an exit of linear dynamic system on which inputs submit aligned white gauss noise. Unknown parametres of the bleaching filter are subject to identification in control process. It is supposed that initial estimations of parametres are random variables with known likelihood characteristics. Variable conditions of the bleaching filter and unknown parametres include in structure of the expanded vector of variables of a condition of object of control. The processes occurring in object of control and the bleaching filter, describe in space of conditions the matrix differential equation.

4. Synthesis of the shaper of operating influences at the stochastic the description uncertainty

In the theory of optimum estimation it is known that updating process filter Kalman intensity and a population mean is a vector white gauss noise with a matrix. It follows from this that evolution of a condition of object of control is described by the stochastic equation. In problems of stochastic management consider that operating influences belong to the set of the limited functions. Thus the management purpose usually formalize in the form of the chosen criterion of quality of management. Most often use average value of square-law function or average value of function of the generalized work. Thus, a problem of synthesis of the

shaper of operating influences adaptive control system consists in function minimization.

This problem решают by means of a stochastic variant of a principle of a maximum or by means of a stochastic variant of a principle of dynamic programming. In linear control systems with square-law function by means of a principle of a maximum and dynamic programming it was possible to receive algorithm of formation of optimum operating influences. This way of synthesis of linear control systems has received enough wide circulation also it is known as a method of analytical designing of optimum regulators of Kalman-Letov. However it was gradually found out that the theory of stochastic control adequately describes rather narrow circle of practical problems. In other situations there is a considerable quantity of the factors interfering application of this theory of control. So, in nonlinear problems of control with function of the generalized work the principle of a maximum and dynamic programming lead to the nonlinear regional problems which decision in real time (as it is required in automatic control systems) it is impossible to find.

In real problems of control there is the inevitable uncertainty connected with presence of non-stationary parameters of model of object of control and non-stationary revolting influences with unknown likelihood characteristics which cannot be defined precisely in control process. Besides, optimum operating influence form in the form of in advance set function of time (programmed control). It is obvious that such strategy of control is the extremely undesirable. It is more natural to form control by means of feedback, as function of target signals of system at time present situation.

5. The analysis of adaptive systems of automatic control

The made analysis shows that many systems with control, function in the conditions of aprioristic uncertainty. In these systems there are nonlinear processes to unknown parameters. On objects controls operate revolting influences with unknown functions probability distributions, thus time of correlation of revolting influences is often commensurable in due course correlations of variables of a condition. The mathematical description of revolting influences, sets of the admissible operating influences and set of admissible trajectories of transfer of object of control in a demanded condition are usually set by means of inequalities. Therefore automatic control system engineering by these objects is necessary for carrying out on the basis of the theory of adaptive systems.

Problems of adaptive control solve now on the basis of a principle of authentic equivalence according to which the problem of adaptive control includes a problem of synthesis of the observer of variables of a condition of object of control, a problem of definition of an optimum trajectory of transition of object of control in a demanded condition and a problem of

synthesis of a regulator operating influences. Synthesis of adaptive control systems carry out by means of the theory stochastic control, regulators of the set structure with optimised parameters and the minimax approach. However at use of these approaches there are the following problems complicating their realisation in adaptive systems of automatic control of nonlinear objects:

1) In the nonlinear systems functioning in the conditions of the aprioristic uncertainty, it is impossible to apply the division theorem at the problem decision optimum estimation of variables of a condition.

2) The Decision of an arising nonlinear problem of optimum control it can be not unique. In this case additional research is necessary, allowing to find out, what should be chosen from available decisions for optimum control.

3) The Decision of a problem of optimum control can appear not smooth enough for practical realisation.

4) Even if the decision of a problem of optimum control has the smooth the decision, the control found from this decision, can not belong to set of admissible controls.

5) The Decision of problems of adaptive control by means of stochastic variants of a principle of a maximum and dynamic programming leads to formation of optimum operating influence under the programmed control law. If the indignations operating in a control system, are distinct from colour white gauss noise such control systems possess a static error of control as operating influences form to proportionally current values of a vector of variables of a condition.

6) Optimum control by means of calculus of variations and maximum principle form by the decision of point-to-point regional problems, generally the nonlinear. However now there are no algorithms of the decision of nonlinear point-to-point regional problems in real time, as it is required in automatic control systems.

7) At approximation of real indignations by colour noise dimension of system of the equations for variables of a fortune of model of object of control that leads to increase in the computing is essentially increased complexities of the decision of a problem of estimation of variables of a condition.

8) For application of methods of the stochastic control based on generation up and the decision of the equation the Fokker-Plank-Kolmogorov, is required to know laws of distribution of all entrance and target signals of object of control.

9) If control systems contain nonlinear elements on which non-stationary revolting influences with unknown likelihood characteristics stochastic methods cannot be applied to synthesis of adaptive systems of automatic control operate.

Conclusion

The method of analytical designing of adaptive systems of automatic control by nonlinear objects in the conditions of aprioristic uncertainty when revolting influences are the signals limited on absolute size is considered, and the mathematical description of

trajectories of transition of operated variables in a final condition is set by inequalities. As a result the theory of optimal stochastic control which in aggregate with digital computers represents a powerful method of the decision of problems of control has been created. This approach was widely adopted due to its mathematical elegance and possibility to use computers for realization of algorithms of control.

The bibliographic list

[Фельдбаум, 1966] Фельдбаум А.А. Основы теории оптимальных автоматических систем // М.: Наука, 1966. 532 с.

[Саридис, 1980] Саридис Дж. Самоорганизующиеся стохастические системы управления // М.: Наука, 1980. – 400 с.

[Тихонов, 1979] Тихонов А.Н. Методы решения некорректных задач / В.Я.Арсенин // М.: Наука, 1979. – 285 с.

[Гольцов, 2004] Гольцов А.С. Адаптивное управление нестационарными объектами в условиях априорной неопределенности // Проблемы транспорта: Сб. науч. тр. СПб.: Международная академия транспорта, 2004. Вып. 11. 126-135 с.

[Гольцов, 2001] Гольцов А.С. Многомерная адаптивная система автоматического управления // Авиакосмическая техника и технология. 2001. №26. 272-274 с.

[Гольцов, 2004] Гольцов А.С. Синтез систем автоматического управления нелинейными объектами в условиях априорной неопределенности // Идентификация систем и задачи управления: Тр. Междунар. конф. М. Институт проблем управления им. В.А.Трапезникова РАН, 2004. 1142-1165 с.

[Zadeh, 1963] Zadeh L.A. Linear System Theory / C. Desoer // New York: McGraw-Hill, 1963. 375 p.

ПРИНЦИП ДОСТОВЕРНОЙ ЭКВИВАЛЕНТНОСТИ В ЗАДАЧАХ АДАПТИВНОГО УПРАВЛЕНИЯ В УСЛОВИЯХ АПРИОРНОЙ НЕОПРЕДЕЛЕННОСТИ

Бабамухамедова М.З., Дошанова М.Ю.,
Эргашев Ф.А.

*Ташкентский университет информационных
технологий, Ташкент, Узбекистан*

malikadoshanova@mail.ru

abbs@mail.ru

*Ташкентский государственный технический
университет, Ташкент, Узбекистан*

qwerty@mail.ru

В статье рассматриваются системы с управлением, которые функционируют в условиях априорной неопределенности с неизвестными параметрами. На объекты управления действуют возмущающие воздействия с неизвестными функциями распределения вероятности математическое описание, которое задается с помощью неравенств. Поэтому разработку систем автоматического управления этими объектами необходимо осуществлять на основе теории адаптивных систем. Задачи адаптивного управления в настоящее время решают на базе принципа достоверной эквивалентности.



УДК 519.876.5

МЕТОДИКА И ПРОГРАММНОЕ СРЕДСТВО АНАЛИЗА СЛОЖНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

Бурак Т.И., Лукашевич М.М.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

timburik@gmail.com

lukashevich@bsuir.by

Приведена методика компьютерного моделирования сложных динамических систем на основе системного подхода Дж. Форрестера. Представлено краткое описание и характеристика программного средства моделирования динамических систем.

Ключевые слова: компьютерное моделирование, системный подход, динамические системы.

Введение

Одной из важнейших научных проблем естествознания является решение задачи предсказания поведения изучаемого объекта (сложной системы) во времени и пространстве на основе определенных знаний о его начальном состоянии и характере влияния внешних факторов. Современный этап развития методов математического моделирования в эколого-биологической области во многом связан с развитием вычислительной техники и возможностью детализации свойств моделируемого объекта, т.е. открыта перспектива построения подробных имитационных моделей [Sarancha и др., 2012].

1. Системный подход Дж. Форрестера

В большинстве прикладных задач требуется работать со сложными динамическими системами, и в этом случае функционала типовых схем оказывается недостаточно. Один из подходов к моделированию динамических систем описан в работах Дж. Форрестера [Forrester, 2006]. Этот подход объединяет в себе черты типовых схем и является интуитивно-понятным и гибким инструментом при моделировании сложных динамических систем. Основная идея системного подхода заключается в том, что модель описывается как совокупность некоторых существенных для процесса моделирования характеристик, а изменение этих характеристик обуславливаются петлями обратных связей. Петля обратной связи - это замкнутая цепочка взаимодействия, которую можно описать следующим образом: изменение

характеристики системы влечет за собой изменение окружающих условий (других характеристик), что в свою очередь вызывает дальнейшее изменение самой характеристики. Такой подход позволяет наиболее точно отразить процессы, происходящие внутри сложной динамической системы. Для описания системы с петлями обратных связей используются два вида переменных: уровни и темпы. Уровень L - это численное представление некоторой существенной для моделирования характеристики системы. Темп D_L описывает изменение некоторого уровня L в процессе моделирования. В общем случае изменение уровней описывается формулой 1:

$$L(t_{i+1}) = L(t_i) + \Delta t \cdot \sum_{j=1}^N D_{L_j}(IL_D(t_i)). \quad (1)$$

где Δt - промежуток времени между двумя итерациями моделирования;

D_{L_j} - темпы, относящиеся к уровню L ;

N - количество темпов для уровня L ;

$IL_D = \{L_0, \dots, L_m\}$ - некоторый набор уровней системы, которые влияют на изменение уровня L . Здесь и далее будем считать, что $t_0, \dots, t_i, t_{i+1}, \dots, t_n$ - дискретные моменты времени, соответствующие итерациям моделирования (i - номер итерации), и $t_i - t_{i+1} = \Delta t \forall i \in [1, n]$. Если взять Δt достаточно малым и учесть, что $t_{i+1} = t_i + \Delta t$, получим следующее уравнение (формула 2):

$$\frac{L(t_{i+1}) - L(t_i)}{\Delta t} = \frac{\Delta L}{\Delta t} \approx L' = \sum_{j=1}^N D_{L_j}(IL_D(t_i)) \quad (2)$$

Формула 1 есть ни что иное, как метод Эйлера – простейший численный метод решения дифференциальных уравнений первого порядка точности. Для повышения точности вычислений в описываемой методике используется многошаговые метод Адамса-Башфорта [Hairer, 1993]. Введем следующее обозначение:

$$f_L(t_i) = \sum_{j=1}^N D_{L_j}(IL_D(t_i)). \quad (3)$$

Тогда метод Эйлера из формулы 1 преобразуется в формулу 4. Метод Эйлера совпадает с методом Адамса-Башфорта первого порядка, методы Адамса-Башфорта второго, третьего и четвертого порядков приведены в формулах 5-7 соответственно.

$$L(t_{i+1}) = L(t_i) + \Delta t \cdot f_L(t_i). \quad (4)$$

$$L(t_{i+1}) = L(t_i) + \Delta t \cdot \left(\frac{3}{2} f_L(t_i) - \frac{1}{2} f_L(t_{i-1}) \right). \quad (5)$$

$$L(t_{i+1}) = L(t_i) + \Delta t \cdot \left(\frac{23}{12} f_L(t_i) - \frac{4}{3} f_L(t_{i-1}) + \frac{5}{12} f_L(t_{i-2}) \right). \quad (6)$$

$$L(t_{i+1}) = L(t_i) + \Delta t \cdot \left(\frac{55}{24} f_L(t_i) - \frac{59}{24} f_L(t_{i-1}) + \frac{37}{24} f_L(t_{i-2}) - \frac{3}{8} f_L(t_{i-3}) \right). \quad (7)$$

Так как вычисление значений уровней модели идет последовательно, то использование методов Адамса-Башфорта не несет в себе дополнительных вычислительных нагрузок – предыдущие несколько значений $f_L(t)$ для каждого уровня сохраняются в буфере для использования в дальнейших вычислениях.

Таким образом, модель может быть представлена в виде D-схемы как система дифференциальных уравнений. В зависимости от того, используют ли темпы D_L вероятностную составляющую при описании законов изменения уровней L , подход Дж. Форрестера может быть использован для описания как детерминированных, так и стохастических моделей. Для описания уровня L системы достаточно указать начальное значение уровня $L(t_0)$, а также набор темпов D_{L_i} , которые будут отвечать за изменение уровня в процессе моделирования. Такой подход к описанию уровней даёт возможность введения дополнительного класса переменных – констант. Константой называется уровень, для которого не определено ни одного темпа, то есть значение такого уровня не изменяется в процессе моделирования. Для упрощения описания темпов и получения дополнительной информации о модели введем временные переменные Tmp . Временные

переменные представляют собой промежуточные результаты вычислений, которые используются при расчетах темпов. Также при выводе результатов моделирования временные переменные могут использоваться как дополнительные показатели состояния модели. При описании модели временные переменные задаются как функциональные зависимости от уровней и других временных переменных (формула 8):

$$Tmp(t_i) = f(IL(t_i), IT(t_i)). \quad (8)$$

где $IL = \{L_0, \dots, L_m\}$ – некоторый набор уровней системы, от которых зависит значение данной временной переменной;

$IT = \{Tmp_0, \dots, Tmp_k\}$ – некоторый набор временных переменных, от которых зависит значение данной временной переменной.

Так как экобиологические системы являются комплексными и состоят из множества взаимодействующих компонент, необходимо предусмотреть возможность разбиения большой системы на составляющие подсистемы меньшего размера, что упростит процесс описания модели и обеспечит её иерархическую структуру. Для этого введем понятие объекта модели. Объект Obj – это структурный элемент модели, который является условным обозначением компонент системы. Каждый объект представляет собой подмодель, компоненты которой также могут взаимодействовать с компонентами других подмоделей. Разбиение сложных систем на отдельные компоненты значительно упрощает процесс описания модели. Иерархическая структура модели представлена на рисунке 1.

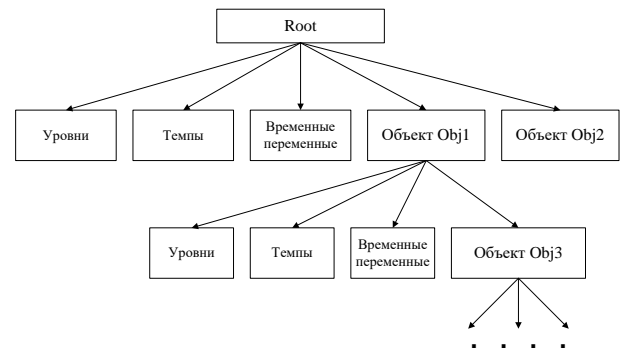


Рисунок 1 – Иерархическая структура модели

2. Программное средство моделирования сложных динамических систем

Подход, предложенный Дж. Форрестером, был реализован в виде программного средства моделирования сложных эколого-биологических систем, в рамках белорусско-российского проекта «Методы и программные средства компьютерного моделирования сложных динамических систем (экологических, биологических)» поддержанного Республиканским фондом фундаментальных исследований (№Ф14Р-042 от 23.05.2014 г.).

Научная идея проекта состоит в создании инструментария для количественного обоснования гипотез о ведущих механизмах изучаемых экологических процессов, опираясь на всю палитру известных математических и биофизических методов. При этом использованы современные компьютерные технологии для поддержки междисциплинарных исследований, позволяющих при создании математических моделей эколого-биологических объектов использовать знания экспертов – биологов. Такой подход позволяет описывать изучаемый объект в «пространстве и во времени», как целиком, так и на уровне индивидов, что позволит значительно повысить адекватность моделируемых процессов. Научная новизна проекта заключается в разработке новых математических моделей и алгоритмического аппарата функционирования динамических систем (в том числе эколого-биологических) и разработке на их основе программного средства моделирования для междисциплинарных исследований в области эко-биологического прогнозирования и регулирования.

Разработан программный интерфейс библиотеки, созданной для демонстрации работы алгоритма моделирования. Программное средство, предоставляющее данный интерфейс пользователю, реализовано в виде динамической библиотеки на языке C#. Программное средство доступно по следующей гиперссылке <http://modeling-timburik.rhcloud.com>.

Структурная схема реализованного программного средства представлена на рисунке 2. Программное средство реализовано в виде клиент-серверного приложения. Серверная часть реализована в виде сервлета для Apache Tomcat Server. Клиентская часть реализована на языке JavaScript с использованием визуального языка программирования Google Blockly для описания формул и библиотеки JointJS для отображения и навигации по иерархической структуре модели. Клиент и сервер обмениваются сообщениями в формате JSON.

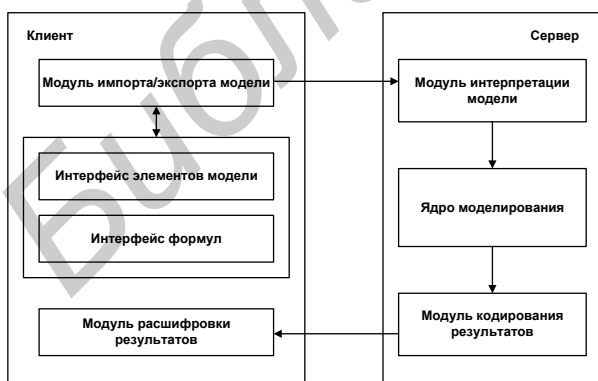


Рисунок 2 – Структурная схема программного средства

Апробация разработанного ПС выполнена на достоверных моделях эко-биологических систем. На рисунках 3 и 4 приведены примеры описания отдельных элементов имитационной модели тундрового сообщества «растительность-лемминги-песцы» [Глушков и др., 2013] с использованием реализованного ПС.

песцы» [Глушков и др., 2013] с использованием реализованного ПС.

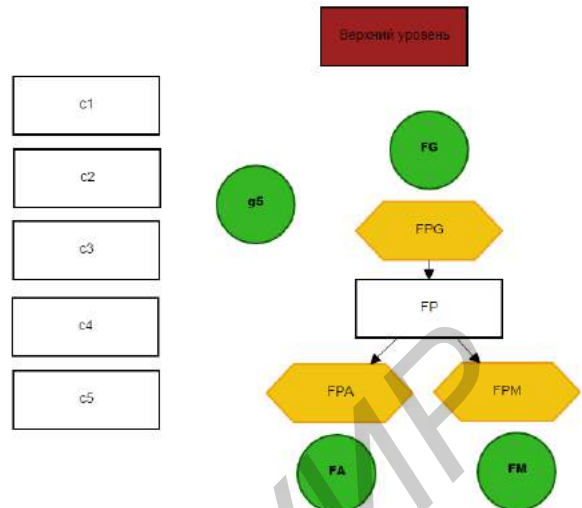


Рисунок 3 – Описание объекта «Песцы»



Рисунок 4 – Пример описании переменной

3. Апробация методики и программного средства

В результате программной реализации данной имитационной модели сообщества «растительность-лемминги-песцы» получены результаты моделирования, которые подтверждают гипотезу о ведущих механизмах функционирования рассматриваемого эколого-биологического объекта [Трапеев и др., 2014], рисунки 5-7.

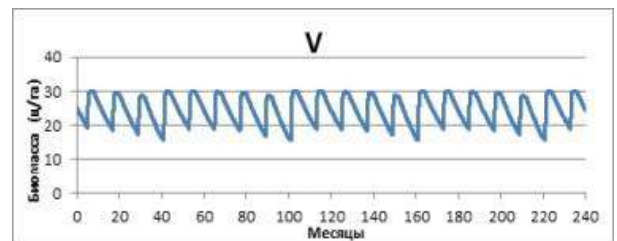


Рисунок 5 – Уровень растительности

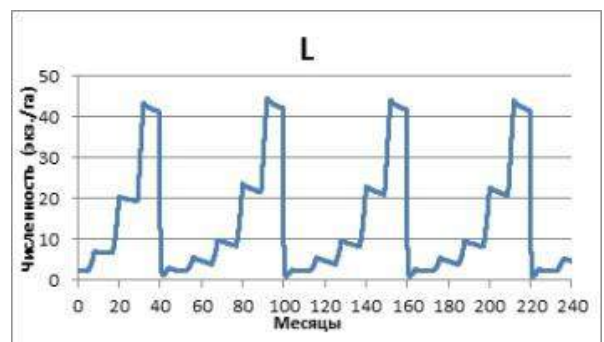


Рисунок 6 – Численность леммингов

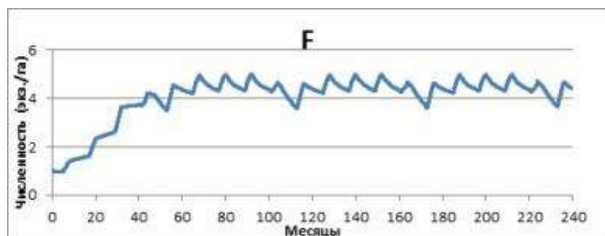


Рисунок 7 – Численность песцов

Заклучение

Предложенная методика является основой для создания систем динамического моделирования. Разработан программный инструментарий для обоснования гипотез о ведущих механизмах функционирования различных сложных динамических систем. Разработанная методика отличается от известных тем, что в ее основу положены работы Дж. Форрестера в области моделирования сложных динамических систем. Работа выполнена в рамках гранта фонда фундаментальных исследований «Методы и программные средства компьютерного моделирования сложных динамических систем (экологических, биологических)» (№Ф14Р-042 от 23.05.2014 г.).

Библиографический список

- [Sarancha и др., 2012] Sarancha, D. A., Lyulyakin, O. P., Trashcheev R. V. Interaction of simulation and analytic methods in modelling of ecological and biological objects. / D. A. Sarancha, O. P. Lyulyakin, R. V. Trashcheev // Russian Journal of Numerical Analysis and Mathematical Modelling.- 2012, Vol. 27, No. 5, pp. 479–492.
- [Глушков и др., 2013] В.Н. Глушков, Д.А. Саранча. Комплексный метод математического моделирования биологических объектов. Моделирование тундрового сообщества. Автоматика и телемеханика, 2013, № 2, стр. 94–108.
- [Форрестер, 2006] Форрестер, Д. Мирровая динамика. - М.: АСТ, 2006. - С. 384.
- [Колесов, 2012] Колесов, Ю. Б. Компонентные технологии математического моделирования: учеб. пособие / Ю. Б. Колесов, Ю. Б. Сениченков. - СПб.: Изд-во Политехн. ун-та, 2012. - 223 с.
- [Советов и др., 2001] Советов Б. Я., Яковлев С. А., Моделирование систем: Учеб. для вузов — 3-е изд., перераб. и доп. — М.: Высш. шк., 2001. — 343 с.
- [Самарский и др., 1997] Самарский А. А., Михайлов А. П. Математическое моделирование: Идеи. Методы. Примеры. — М.: Наука, 1997. — 320 с.
- [Forrester, 2006] Forrester, Jay W. Principles of Systems. (2nd ed.). — MA: Pegasus Communications, Waltham, 2006. — P. 391.
- [Hairer, 1993] Hairer E., Norsett S.P. & Wanner G. Solving ordinary differential equations I: Nostiff problems. (2nd ed.). — Berlin: Springer Verlag, 1993. — P. 391.
- [Трашчев и др., 2014] Трашчев Р.В., Люлякин О.П., Саранча Д.А., Юрезанская Ю.С. Метод комплексных исследований на примере моделирования популяций леммингов — М.: ВЦ РАН, 2014. — 115 с.

METHODOLOGY AND SOFTWARE FOR SEMANTIC ANALYSIS OF COMPLEX DYNAMICAL SYSTEMS

Burak T.I., Lukashevich M.M.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

timburik@gmail.com

lukashevich@bsuir.by

In this paper presented methodology for computer modeling of complex dynamic systems, based on system approach of J. Forrester. A brief description and characteristics of software tools for modeling dynamic systems.

Introduction

The problem of predicting behavior of the object of interest (usually a complex system) is one of the major problems of natural science. Modern level of mathematical modeling development is connected to ability of property specification of modeling objects, which allows us to build complex imitational models.

Main Part

One of the approaches for modeling complex dynamic systems introduced in the papers of J. Forrester, the founder of system dynamics. The main idea of system approach is that model represented as a set of main characteristics, whose changes are driven by feedback loops.

Using system approach as a basis the software tool for modeling complex dynamic systems was developed. This tool has the form of a web application available via this URL: <http://modeling-timburik.rhcloud.com>.

Software tool was used to create model of eco-biological system “Vegetation-Lemmings-Arctic Foxes”, and obtained testing results proves the hypothesis about the leading functioning mechanisms of examined system.

Conclusion

Developed methodology for semantic analysis of complex dynamical systems. Modeling algorithm is based on J.Forrester system approach. Developed software tool was tested on complex eco-biological system.



УДК 004.89

КАЧЕСТВЕННОЕ МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКИХ СИСТЕМ НА ОСНОВЕ МЕТОДОВ РАСПРОСТРАНЕНИЯ ОГРАНИЧЕНИЙ

Зуенко А.А.

Институт информатики и математического моделирования КНЦ РАН,

г. Анатиты, Россия,

zuenko@iimm.ru

В статье описан разработанный автором метод распространения ограничений, предназначенный для решения задач качественного моделирования технических систем. В отличие от аналогов, использующих для описания ограничений над конечными доменами структуры наподобие реляционных таблиц, предлагаемый метод опирается на применение оригинальных матрицеподобных структур, которые позволяют “сжато” представлять нечисловые зависимости между параметрами модели и эффективно их обрабатывать.

Ключевые слова: качественное моделирование, задача удовлетворения ограничений, ограничения с конечными доменами, матрицы ограничений.

Введение

На практике бывает трудно выявить точные соотношения между переменными в моделируемой системе, но все еще остается зависимость установить некоторые качественные зависимости между ними [Братко, 2004]. Кроме того, даже при наличии строгой аналитической модели, для ее применения требуется знать значения всех параметров модели, измерение которых может быть сильно осложнено. В такой ситуации пользователь пытается заменить значения неизвестных параметров предполагаемыми значениями, надеясь, что они не слишком отличаются от истинных. Применение качественных моделей позволяет избежать необходимости выдвижения необоснованных предположений.

Качественная машинная модель позволяет решить задачу путем перебора всех возможных качественных вариантов поведения, которые соответствуют всем возможным сочетаниям значений параметров в модели.

При решении многих задач точные числовые данные не требуются. К типичным задачам, где качественное моделирование является более предпочтительным, относятся функциональные рассуждения, диагностика и проектирование на основе «исходных принципов», например законов физики. Заметим также, что с точки зрения конечного пользователя задача разработки качественных моделей является более простой по сравнению с созданием количественных моделей.

Функциональные рассуждения связаны с поиском ответов на вопросы, каким образом работает устройство или система. Причем, интерес представляет (качественный) механизм работы системы: если числовые значения параметров системы немного изменяются, обычно основной функциональный механизм остается тем же.

При решении диагностической задачи интерес представляют неисправности, которые вызвали наблюдаемое аномальное поведение системы.

Проблема структурного синтеза на основе использования «исходных принципов» состоит в следующем. Даны некоторые основные блоки. Требуется найти такую комбинацию этих блоков (структуру), которая позволяет реализовать заданную функцию.

Для использования качественных моделей требуются определенные методы проведения качественных рассуждений, изложению которых и посвящена настоящая статья.

Для решения обозначенного круга задач предлагается использовать разработанные автором методы распространения ограничений, основанные на матричном представлении ограничений в виде *S*- и *D*- систем [Зуенко, 2013], [Зуенко, 2014], [Зуенко и др., 2015]. Другими словами, перечисленные выше задачи предлагается решать в виде задач удовлетворения ограничений (Constraint Satisfaction Problem – CSP) [Ruttkay, 1998], [Bartak, 1999], [Рассел и др., 2006]. В отличие от предыдущих работ цикла в настоящей статье для

формализации ограничений используются не D -системы (матрицы дизъюнктов), а C -системы (матрицы конъюнктов), а также разработан оригинальный метод распространения ограничений для этого случая.

1. Пример формализации технической системы в виде матриц ограничений

В качестве примера выберем простую электрическую схему, состоящую из выключателей, ламп и источников питания (рисунок 1). Качественная модель, описывающая работу данной схемы, взята из [Братко, 2004]. В приведенной книге языком моделирования служит Prolog.

В настоящей работе представлена интерпретация данной модели в терминах матриц ограничений, а, именно, в виде набора C -систем, а возникающие задачи моделирования предлагается рассматривать как задачи удовлетворения ограничений.

Выключатели могут быть разомкнутыми или замкнутыми (т.е. выключенными или включенными), а лампы могут быть светящимися или темными, сгоревшими или исправными. Нас, как и в [Братко, 2004], интересует поиск ответов на вопросы, которые относятся к области прогнозирования состояний, диагностики или управления такими электрическими схемами.

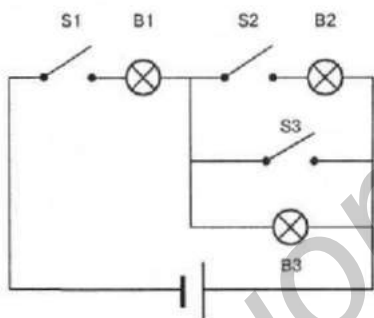


Рисунок 1 - Пример электрической схемы

Создание модели схемы сводится к заданию следующих ограничений:

1. Законы функционирования лампы.
2. Законы функционирования выключателя.
3. Закон Кирхгофа (в качественной формулировке): напряжение на выключателе + напряжение на лампе = напряжение источника питания.

Также неявно посредством общих атрибутов (переменных) должны учитываться физические соединения между компонентами.

В рассматриваемой качественной модели электрические токи и напряжения имеют лишь качественные значения "pos" (положительный), "zero" (нулевой) и "neg" (отрицательный).

Компоненты исследуемой электрической схемы делятся на два типа: выключатели и лампы. Качественное поведение этих компонентов

описывается с помощью типовых отношений. Для выключателя – это отношение $SWITCH(SP, V, C)$:

$$\begin{matrix} SP & V & C \\ \{on, off\} & \{neg, zero, pos\} & \{neg, zero, pos\} \\ 1 \left[\begin{matrix} \{on\} & \{zero\} & * \\ \{off\} & * & \{zero\} \end{matrix} \right] \end{matrix} \quad (1)$$

Атрибуты SP , V , C – это положение выключателя, напряжение и ток, соответственно.

Например, из строки 2 следует, что на разомкнутом (*off*) выключателе ток является нулевым (*zero*), а напряжение может иметь любое значение.

Закон функционирования лампы может быть выражен в виде отношения $BULB(B, L, V, C)$:

$$\begin{matrix} B & L & V & C \\ \{ok, blown\} & \{dark, light\} & \{neg, zero, pos\} & \{neg, zero, pos\} \\ 1 \left[\begin{matrix} \{blown\} & \{dark\} & * & \{zero\} \\ \{ok\} & \{light\} & \{pos\} & \{pos\} \\ \{ok\} & \{light\} & \{neg\} & \{neg\} \\ * & \{dark\} & \{zero\} & \{zero\} \end{matrix} \right] \end{matrix} \quad (2)$$

Атрибуты B , L , V , C – это признак исправности лампы, наличие/отсутствие свечения, напряжение на лампе и ток, соответственно.

Согласно строке 1, сгоревшая (*blown*) лампа остается темной (*dark*), через нее не проходит ток, а напряжение может иметь любое значение. Исправная (*ok*) лампа светится постоянно, за исключением того случая, когда и напряжение, и ток в лампе равны нулю. При этом предполагается, что любой ненулевой ток является достаточно большим для того, чтобы заставить лампу светиться.

Напряжение и ток могут одновременно либо равняться нулю, либо быть положительными, либо быть отрицательными. По сути, это качественная абстракция закона Ома, который формулируется следующим образом:

$$\text{Напряжение} = \text{Сопротивление} * \text{Сила_тока.}$$

Поскольку сопротивление лампы является положительным, то напряжение (V) и ток (C) должны иметь одинаковый знак и поэтому одно и то же качественное значение.

В числовых моделях электрических схем, помимо закона Ома, используются некоторые фундаментальные законы, такие как законы Кирхгофа. Законы Кирхгофа гласят: во-первых, сумма всех напряжений вдоль любого замкнутого контура в схеме равна 0, во-вторых, сумма всех токов в любом соединении в схеме равна 0. Чтобы применить эти законы в качественной модели, необходимо предусмотреть качественную версию операции арифметического суммирования. В настоящей работе вместо обычного арифметического суммирования $X + V = Z$ применяется сокращенный вариант в виде операции

качественного суммирования, которая реализована в форме следующего отношения $QSUM[X, Y, Z]$:

X	Y	Z
$\{neg, zero, pos\}$	$\{neg, zero, pos\}$	$\{neg, zero, pos\}$
1	$\{pos\}$	$\{pos\}$
2	$\{zero\}$	$\{pos\}$
3	$\{neg\}$	$\{pos\}$
4	$\{neg\}$	$\{zero\}$
5	$\{neg\}$	$\{neg\}$
6	$\{pos\}$	$\{pos\}$
7	$\{zero\}$	$\{zero\}$
8	$\{neg\}$	$\{neg\}$
9	$\{pos\}$	$\{pos\}$
10	$\{neg\}$	$\{zero\}$
11	$\{pos\}$	$\{neg\}$
12	$\{zero\}$	$\{neg\}$
13	$\{neg\}$	$\{neg\}$

Данное отношение можно рассматривать как некий тип отношения (типовое отношение), на основе которого в процессе подстановки вместо X, Y, Z конкретных атрибутов получается конкретное отношение. Подобным образом в логических исчислениях конкретизируется схема аксиомы, превращаясь в конкретную аксиому в ходе подстановки.

Строки этой матрицы представляют собой некоторые факты, например, в строке 1 утверждается, что сумма двух положительных чисел представляет собой положительное число. Строки 3, 4, 5, что сумма положительного и отрицательного чисел может представлять собой любое значение из множества $\{neg, zero, pos\}$.

Такая операция суммирования является недетерминированной (неопределенной). Из-за отсутствия точной информации, потерянной в процессе качественного абстрагирования, иногда невозможно определить, каковым фактически является результат суммирования. Такого рода недетерминированность является довольно типичной для качественных рассуждений.

Отношение $QSUM$ может быть задано и в более "компактной" форме:

X	Y	Z
$\{neg, zero, pos\}$	$\{neg, zero, pos\}$	$\{neg, zero, pos\}$
1	$\{zero\}$	$\{zero\}$
2	*	$\{pos\}$
3	$\{neg\}$	*
4	$\{pos\}$	$\{pos\}$
5	$\{neg\}$	$\{neg\}$
6	$\{pos\}$	*
7	*	$\{neg\}$

После того как определены отдельные компоненты, задача определения всей схемы

становится несложной. Каждая конкретная схема может быть определена с помощью системы отношений, записанных в виде C-систем (и/или D-систем).

Например, для моделирования электрической схемы, приведенной на рисунок 1, нужно рассмотреть следующие отношения.

№ 1. Выключатель "S1" – $SWITCH[Sw1, VSw1, C1]$:

Sw1	VSw1	C1
$\{on, off\}$	$\{neg, zero, pos\}$	$\{neg, zero, pos\}$
1	$\{zero\}$	*
2	*	$\{zero\}$

№ 2. Лампа "B1" – $BULB[B1, L1, VB1, C1]$:

B1	L1	VB1	C1
$\{ok, blown\}$	$\{dark, light\}$	$\{neg, zero, pos\}$	$\{neg, zero, pos\}$
1	$\{dark\}$	*	$\{zero\}$
2	$\{light\}$	$\{pos\}$	$\{pos\}$
3	$\{light\}$	$\{neg\}$	$\{neg\}$
4	*	$\{zero\}$	$\{zero\}$

№ 3. Выключатель "S2" – $SWITCH[Sw2, VSw2, C2]$:

Sw2	VSw2	C2
$\{on, off\}$	$\{neg, zero, pos\}$	$\{neg, zero, pos\}$
1	$\{zero\}$	*
2	*	$\{zero\}$

№ 4. Лампа "B2" – $BULB[B2, L2, VB2, C2]$:

B2	L2	VB2	C2
$\{ok, blown\}$	$\{dark, light\}$	$\{neg, zero, pos\}$	$\{neg, zero, pos\}$
1	$\{dark\}$	*	$\{zero\}$
2	$\{light\}$	$\{pos\}$	$\{pos\}$
3	$\{light\}$	$\{neg\}$	$\{neg\}$
4	*	$\{zero\}$	$\{zero\}$

№ 5. Выключатель "S3" – $SWITCH[Sw3, V3, CSw3]$:

Sw3	V3	CSw3
$\{on, off\}$	$\{neg, zero, pos\}$	$\{neg, zero, pos\}$
1	$\{zero\}$	*
2	*	$\{zero\}$

№ 6. Лампа "B3" – $BULB[B3, L3, V3, CB3]$:

$$\begin{array}{cccc}
 B3 & L3 & V3 & CB3 \\
 \{ok,blown\} & \{dark,light\} & \{neg,zero,pos\} & \{neg,zero,pos\} \\
 1 \left[\begin{array}{cccc}
 \{blown\} & \{dark\} & * & \{zero\} \\
 \{ok\} & \{light\} & \{pos\} & \{pos\} \\
 \{ok\} & \{light\} & \{neg\} & \{neg\} \\
 * & \{dark\} & \{zero\} & \{zero\}
 \end{array} \right] & (10)
 \end{array}$$

№ 7. Закон “Напряжения на выключателе S1 и лампе B1 складываются”:

$$\begin{array}{ccc}
 VSw1 & VB1 & V1 \\
 \{neg,zero,pos\} & \{neg,zero,pos\} & \{neg,zero,pos\} \\
 1 \left[\begin{array}{ccc}
 \{zero\} & \{zero\} & \{zero\} \\
 \{pos\} & * & \{pos\} \\
 \{pos\} & \{neg\} & * \\
 * & \{pos\} & \{pos\} \\
 * & \{neg\} & \{neg\} \\
 \{neg\} & \{pos\} & * \\
 \{neg\} & * & \{neg\}
 \end{array} \right] & (11)
 \end{array}$$

№ 8. Закон “Напряжения на выключателе S2 и лампе B2 складываются”:

$$\begin{array}{ccc}
 VSw2 & VB2 & V3 \\
 \{neg,zero,pos\} & \{neg,zero,pos\} & \{neg,zero,pos\} \\
 1 \left[\begin{array}{ccc}
 \{zero\} & \{zero\} & \{zero\} \\
 \{pos\} & * & \{pos\} \\
 \{pos\} & \{neg\} & * \\
 * & \{pos\} & \{pos\} \\
 * & \{neg\} & \{neg\} \\
 \{neg\} & \{pos\} & * \\
 \{neg\} & * & \{neg\}
 \end{array} \right] & (12)
 \end{array}$$

№ 9. Закон “Напряжение в сети положительно”

$$\begin{array}{cc}
 V1 & V3 \\
 \{neg,zero,pos\} & \{neg,zero,pos\} \\
 1 \left[\begin{array}{cc}
 \{pos\} & * \\
 \{zero,neg\} & \{pos\}
 \end{array} \right] & (13)
 \end{array}$$

№ 10. Закон “Токи на выключателе S3 и лампе B3 суммируются”

$$\begin{array}{ccc}
 CSw3 & CB3 & C3 \\
 \{neg,zero,pos\} & \{neg,zero,pos\} & \{neg,zero,pos\} \\
 1 \left[\begin{array}{ccc}
 \{zero\} & \{zero\} & \{zero\} \\
 \{pos\} & * & \{pos\} \\
 \{pos\} & \{neg\} & * \\
 * & \{pos\} & \{pos\} \\
 * & \{neg\} & \{neg\} \\
 \{neg\} & \{pos\} & * \\
 \{neg\} & * & \{neg\}
 \end{array} \right] & (14)
 \end{array}$$

№ 11. Закон 5 “Сила тока всей электрической цепи”

$$\begin{array}{ccc}
 C2 & C3 & C1 \\
 \{neg,zero,pos\} & \{neg,zero,pos\} & \{neg,zero,pos\} \\
 1 \left[\begin{array}{ccc}
 \{zero\} & \{zero\} & \{zero\} \\
 \{pos\} & * & \{pos\} \\
 \{pos\} & \{neg\} & * \\
 * & \{pos\} & \{pos\} \\
 * & \{neg\} & \{neg\} \\
 \{neg\} & \{pos\} & * \\
 \{neg\} & * & \{neg\}
 \end{array} \right] & (15)
 \end{array}$$

Процесс моделирования электрической схемы завершен. Теперь рассмотрим примеры задач, которые в работе предлагается ставить и решать в форме задач удовлетворения ограничений.

Вопросы прогностического типа

Какими будут наблюдаемые результаты некоторого "входного воздействия" на систему (изменения положений выключателей), если дано некоторое функциональное состояние системы (лампы — исправные или сгоревшие). Например, что произойдет, если будут включены (on) все выключатели, притом что все лампы исправны (ok)?

Вопросы диагностического типа

Если известны входные воздействия на систему и некоторые результаты наблюдений, то каково состояние функционирования системы (исправна она или неисправна, и в чем состоит неисправность?). Например, если лампа 1 светится, лампа 3 остается темной, а выключатель 3 выключен, то каковы состояния ламп?

Вопросы управленческого типа

Каким должно быть управляющее воздействие, позволяющее достичь желаемого результата? Например, какими должны быть положения выключателей, чтобы заставить светиться лампу 3, при условии, что все лампы исправны?

Далее описывается метод распространения ограничений, позволяющий получать ответы на подобные вопросы.

2. Организация вывода на ограничениях

Приведем утверждения, позволяющие реализовывать эквивалентные преобразования системы ограничений для случая, когда ограничения представлены в виде набора C-систем. Целью преобразований является приведение системы ограничений к более простому виду, где содержится меньшее количество C-систем, строк C-систем, столбцов (атрибутов) C-систем, значений в доменах атрибутов и т.п.

Утверждение 1 (У1). Если все строки (кортежи) C-системы пусты, то есть содержат хотя бы по одной пустой компоненте каждая, то C-система пуста (соответствующая задача CSP несовместна).

Утверждение 2 (У2). Если все компоненты некоторого атрибута (столбца *C*-системы) являются полными, то данный атрибут можно удалить из *C*-системы (удаляются все компоненты стоящие в соответствующем столбце), а пара “удаляемый атрибут – его домен” сохраняется в векторе частичного решения.

Утверждение 3 (У3). Если домен некоторого атрибута *C*-системы содержит значения, не встречающиеся в соответствующем столбце, то эти значения удаляются из данного домена.

Утверждение 4 (У4). Если строка *C*-системы содержит хотя бы одну пустую компоненту (строка пуста), то строка удаляется.

Утверждение 5 (У5). Если компонента некоторого атрибута содержит значение, не принадлежащее соответствующему домену, то это значение удаляется из компоненты.

Утверждение 6 (У6). Если одна строка *C*-системы полностью доминирует (покомпонентно содержит) другую строку, то доминируемая строка удаляется из *C*-системы.

Продемонстрируем поиск ответа на вопрос прогностического типа, который приводился ранее. Дано: $Sw1 - \{on\}$, $Sw2 - \{on\}$, $Sw3 - \{on\}$, $B1 - \{ok\}$, $B2 - \{ok\}$, $B3 - \{ok\}$. Требуется определить значения параметров $L1, L2, L3$.

Процесс получения решения отображен в таблице 1. Рассмотрим строку 1, которая соответствует первому шагу. В список на обработку поступают *C*-системы (ограничения) с номерами 1-6, которые имеют в своих схемах атрибуты, чьи значения заданы в условии. В качестве текущего, то есть рассматриваемого на данном шаге, берется ограничение 1:

$$\begin{array}{ccc}
 Sw1 & VSw1 & C1 \\
 \{on\} & \{neg, zero, pos\} & \{neg, zero, pos\} \\
 1[\{on\} & \{zero\} & * \\
 2[\{off\} & * & \{zero\}]
 \end{array} \quad (16)$$

Заметим, что домен атрибута $Sw1$ конкретизирован с учетом условий задачи. К данной *C*-системе применяется сначала утверждение **У5** и в строке 2 появляется пустая компонента, соответствующая атрибуту $Sw1$. Затем, используя **У4**, элиминируем вторую строку:

$$\begin{array}{ccc}
 Sw1 & VSw1 & C1 \\
 \{on\} & \{neg, zero, pos\} & \{neg, zero, pos\} \\
 1[* & \{zero\} & *]
 \end{array} \quad (17)$$

Теперь, согласно, **У3** конкретизируем домен атрибута $VSw1$: $VSw1 - \{zero\}$. Затем строка 1 удаляется из *C*-системы, поскольку по **У2** элиминируются все ее атрибуты.

В последнем столбце таблицы, жирным цветом показаны атрибуты, чьи значения удалось

конкретизировать на текущем шаге, а без выделения приводятся атрибуты, значения которых были уже известны до выполнения шага.

Теперь исключаем из списка ограничение 1 и добавляем в него ограничение 7, содержащее конкретизированный атрибут $VSw1$ (это видно из строки 2 таблицы). Остальные шаги выполняются по аналогии.

Процесс распространения ограничений оканчивается вычеркиванием всех атрибутов из всех *C*-систем. То есть в результате распространения ограничений нам удалось конкретизировать значения всех переменных рассматриваемой задачи CSP. На каждом шаге происходит усечение доменов одних атрибутов на основе известных доменов других атрибутов. В результате решается задача конкретизации значений целевых параметров.

Таблица 1 – Распространение ограничений

Шаг	Текущее ограничение	Список ограничений	Домены
1	1	1, 2, 3, 4, 5, 6	$Sw1 - \{on\}$, $VSw1 - \{zero\}$
2	3	2, 3, 4, 5, 6, 7	$Sw2 - \{on\}$, $VSw2 - \{zero\}$
3	5	2, 4, 5, 6, 7, 8	$Sw3 - \{on\}$, $V3 - \{zero\}$
4	9	2, 4, 6, 7, 8, 9	$V1 - \{pos\}$, $V3 - \{zero\}$
5	6	2, 4, 6, 7, 8	$B3 - \{ok\}$, $L3 - \{dark\}$, $V3 - \{zero\}$, $CB3 - \{zero\}$
6	7	2, 4, 7, 8, 10	$VSw1 - \{zero\}$, $VB1 - \{pos\}$, $V1 - \{pos\}$
7	2	2, 4, 8, 10	$B1 - \{ok\}$, $L1 - \{light\}$, $VB1 - \{pos\}$, $C1 - \{pos\}$
8	8	4, 8, 10, 11	$VSw2 - \{zero\}$, $VB2 - \{zero\}$, $V3 - \{zero\}$
9	4	4, 10, 11	$B2 - \{ok\}$, $L2 - \{dark\}$, $VB2 - \{zero\}$, $C2 - \{zero\}$
10	11	10, 11	$C2 - \{zero\}$, $C3 - \{pos\}$, $C1 - \{pos\}$
11	10	10	$CSw3 - \{pos\}$, $CB3 - \{zero\}$, $C3 - \{pos\}$

Итак, ответ: $L1 - \{light\}$, $L2 - \{dark\}$, $L3 - \{dark\}$. Если будут включены (*on*) все выключатели, притом что все лампы исправны (*ok*), то светиться будет только первая лампа.

Заключение

В работе рассмотрена возможность применения методов программирования в ограничениях для качественного моделирования технических систем. Представление ограничений в матричной форме, а именно в виде набора C -систем, позволяет “сжато” описывать множества кортежей, содержащихся в многоместных отношениях, с помощью которых часто выражаются ограничения (особенно в Prolog-системах). Помимо экономии памяти, использование предложенного подхода к обработке ограничений позволяет существенно ускорить традиционные алгоритмы-пропагаторы.

Работа выполнена при поддержке РФФИ (проекты № 14-07-00205-а «Информационная технология моделирования гибких недоопределенных динамических систем на основе логико-аналитических методов», № 14-07-00256-а «Алгебраический подход к параллельной обработке данных и знаний на основе алгебры кортежей»).

Библиографический список

[Братко, 2004] Братко, И Алгоритмы искусственного интеллекта на языке PROLOG. 3-е изд. / И Братко // пер. с англ. — М.: Изд. дом «Вильямс», 2004. - 640 с.

[Зуенко, 2013] Зуенко, А.А. Матрицеподобные вычисления в задачах удовлетворения ограничений / А.А. Зуенко // Шестая Всероссийская конференция по проблемам управления, 30 сентября – 5 октября 2013 г., г. Геленджик, с. Дивноморское: материалы конференции в 4 т. – Ростов-на-Дону: Изд-во Южного федерального университета, 2013. -Т.1. – С.30-34.

[Зуенко, 2014] Зуенко, А.А. Вывод на ограничениях с применением матричного представления конечных предикатов / А.А. Зуенко // Искусственный интеллект и принятие решений, 2014. – Вып. 3. – С.21-31.

[Зуенко и др., 2015] Зуенко, А.А. Эвристический метод удовлетворения ограничений на основе их матричного представления / А.А. Зуенко, А.А. Очинская // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2015): материалы Пятой междунар. научн.-техн. конф. (Минск, 19-21 февраля 2015 г.) – Минск: БГУИР, 2015. – С. 297-302.

[Рассел и др., 2006] Рассел, С. Искусственный интеллект: современный подход. 2-е изд. / С. Рассел, П. Норвиг // пер. с англ.; ред. К.А. Птицына. – М.: Изд. дом «Вильямс», 2006. -1408 с.

[Bartak, 1999] Bartak R. Constraint Programming: In Pursuit of the Holy Grail // Proceedings of the Week of Doctoral Students (WDS99), Part IV. – Prague: MatFyzPress, 1999. P. 555–564.

[Ruttkey, 1998] Ruttkey Zs. Constraint satisfaction a survey // CWI Quarterly. 1998. V. 11. P. 163–214.

QUALITATIVE MODELING OF TECHNICAL SYSTEMS BASED ON THE CONSTRAINT PROPAGATION METHODS

Zuenko A.A.

Institute for Informatics and Mathematical Modelling, Kola Science Centre of RAS, Apatity, Russia

zuenko@iimm.ru

The constraint propagation technique developed by the author is presented in the article. The technique is developed for tasks of qualitative modeling of technical systems. Despite of prototypes that use such structures as relational tables for constraints representation over finite domains, the proposed technique relies on using of original matrix-like structures allowing represent qualitative regularities between parameters of the model in compressed form and speed up their processing.

Introduction

Many tasks don't require accurate numeric data for its solution. Typical tasks where qualitative modeling is more preferable include such tasks as functional reasoning, diagnostics, structural synthesis.

For solution of tasks mentioned above the author developed constraint propagation techniques, based on matrix representation of constraints in the form of C -systems and D -systems. The tasks mentioned above are proposed to solve as constraint satisfaction problems. Despite of earlier publications of the author in the article C -systems are used to formalize constraints instead of D -systems. An original constraints propagation technique is developed for this case as well.

Main Part

The affirmations that allow realize equivalent transformations of constraint system are presented in the article. The purpose of the transformations is to reduce of constraint system to the form that contains less number of C -systems, rows of C -systems, columns (attributes) of C -systems, members of attribute domains and etc.

Simple electric circuit that contains of switchers, bulbs and power is chosen for illustration of the inference. A qualitative model of the electric circuit is constructed in the form of constraints system. A forecasting task is solved on the model as well.

The solving is presented in the special table. Domains of attributes are reduced on each step basing on the known domains of other attributes. As a result of such reduction the values of required parameters are inferred.

Conclusion

The matrix representation of constraints in the form of the set of C -systems allow compressively write a set of n -tuples of n -ary relations. Such relations are used for representation of constraints (as in Prolog-systems). Besides the space economy the suggested technique for constraints handling gives opportunity to significantly accelerate typical algorithms-propagators.

The authors would like to thank the Russian Foundation for Basic Research (grants 14-07-00205-a, 14-07-00256-a).



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891.3

ПРИМЕНЕНИЕ СЕТЕЙ БАЙЕСА ДЛЯ АНАЛИЗА ЭКГ

Саган В.Ю., Фридман Р.

*Институт прикладного системного анализа Национального технического университета Украины
«Киевский политехнический институт», г. Киев, Украина*

Vitaliysagan14@gmail.com

Ronfridman1995@gmail.com

В данной работе описаны исследования в области автоматизированного анализа работы сердца по электрокардиограммам с применением машинного обучения. В частности описаны результаты применения байесовских сетей и наивного байесовского классификатора для определения наличия кардиологических патологий. Полученные результаты указывают на то, что данный подход применим к поставленной задаче. Наилучшие результаты были получены с использованием улучшенной байесовской сети.

Ключевые слова: электрокардиограмма; байесовский классификатор; байесовские сети; машинное обучение; генетический алгоритм; сердечнососудистые заболевания.

Введение

Проблема сердечнососудистых заболеваний (ССЗ) крайне остро стоит в современном обществе. По данным отчета всемирной организации здравоохранения ССЗ являются причиной более 30% всех смертей в мире, что делает их основной мировой причиной смертности. Таким образом, профилактика сердечнососудистых заболеваний является крайне важной. [FridmanR., SaganV., 2015]

На данный момент на смену классическому для медицины подходу, предусматривавшему в основном реагирование врачей на возникшие заболевания, приходит новая парадигма. Она называется Predictive, Preventive and Personalized Medicine (PPPM) что означает прогностическая профилактическая персонализированная медицина. Основной акцент в этой парадигме сделан на профилактике заболеваний и их лечении на ранних стадиях. В частности она предусматривает возможности внедрения профилактических и диагностических средств и методов повседневную жизнь пациентов, что с одной стороны позволит снизить нагрузку на врачей, а с другой – увеличит качество контроля здоровья пациентов.

На данный момент специалисты используют различные методы анализа состояния сердца, например УЗИ и МРТ, но базовым методом остается электрокардиография (ЭКГ). ЭКГ – это методика регистрации и исследования

электрических полей, образующихся при работе сердца. ЭКГ записывается при помощи специального устройства – электрокардиографа. Он считывает данные о разности электрических потенциалов в различных четко определенных точках тела человека и затем строит по этим данным электрокардиограмму.

Данные для электрокардиографа поступают с установленных на теле человека электродов. Традиционно они расположены на специальных присосках, которые должны быть размещены в четко определенных местах. Однако в последнее время появилось множество решений использующих носимые датчики типа Holter, которые позволяют считывать данные даже человеку, не имеющему специальной подготовки. Из этого следует, что в случае создания качественной системы, анализирующей данные полученные электрокардиографом, в ближайшее время контроль данных ЭКГ может стать ежедневной профилактической процедурой, особенно для людей, которые находятся в зоне риска.

1. Анализ ЭКГ

1.1. Схема записи данных на ЭКГ

Классическая схема записи ЭКГ подразумевает установку 12 электродов, по данным с которых формируется 12 отведений. Каждое отведение представляет собой временной ряд, который описывает некоторую картину электрической активности сердца. Таким образом, отведения

одинаковы по своей природе, однако могут содержать в себе разную информацию. На практике для анализа состояния сердца чаще всего используют 6 отведений, которые обозначаются как I, II, III, VR, VL и VF. Основным объектом анализа при обработке ЭКГ являются зубцы, которые отражают различные этапы сердечного сокращения. Признаки, используемые методами машинного обучения при обработке ЭКГ, могут быть получены непосредственно из зубцов (длина S-T интервала, высота R-зубца), однако это не единственный способ их получения (рис. 1).

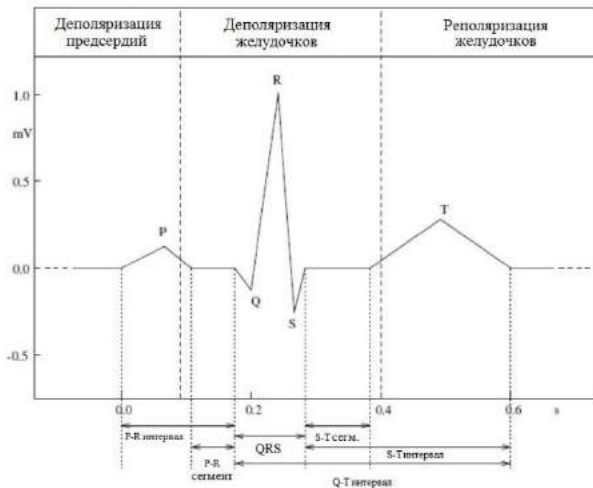


Рисунок 1 – Отображение удара сердца на ЭКГ человека

1.2. Выделение признаков из ЭКГ

Методы машинного обучения не могут быть непосредственно применены к сигналу ЭКГ и требуют преобразования исходного сигнала. В дальнейшем алгоритмы машинного обучения позволят принять решение, насколько здоров пациент. Однако на начальном этапе необходима предобработка сигнала по схеме, показанной на рисунке 2.

Авторами представлена классификация различных типов признаков: Признаки выделяемые из ЭКГ классифицируются на: интервальные признаки, морфологические признаки, основанные на изучении структуры ЭКГ сигнала, признаки, в основе которых лежит вейвлет-преобразование, признаки, полученные при переводе ЭКГ сигнала в частотную область (с использованием быстрого преобразования Фурье) и признаки, сформированные на основе формы сигнала (shapematching).



Рисунок 2 – Этапы предварительной обработки сигнала ЭКГ

Каждый класс, может содержать от нескольких десятков до нескольких сотен различных признаков, которые захватывают некоторые специфические характеристики кривой ЭКГ и содержат информацию о сердечных патологиях.

Во время фазы обучения модели выбраны

наиболее информативные признаки среди всех признаков и используются для дальнейшего обучения модели. Все остальные признаки были отброшены. Детальнее об этой фазе в 3 разделе. Во время рабочей фазы все извлеченные признаки из каждого комплекса для каждого отведения хранятся в векторах. Затем каждый заданный вектор для каждого периода времени (например, для 5 комплексов) упакован в столбец матрицы признаков (featurematrix). Матрица признаков используется в дальнейшем в качестве входа для байесовской сети. [Chereda H.O., Nikolaiev S.S., 2015]

При выполнении исследования была использована база данных, собранная специалистами из университетов BilkentUniversity и BaskentUniversity в Анкаре, Турция. В ней содержатся данные о кардиограммах 452 людей. Каждой кардиограмме соответствует 279 различных параметров. В первую очередь это основные данные о пациенте, например пол, возраст, рост и вес. В выборке не содержатся записи ЭКГ, однако в ней указаны некоторые характеристики каждой из них. Например, указаны средние значения длин некоторых интервалов, амплитуды некоторых зубцов и также присутствие или отсутствие зубцов характерной формы (например, оборванных или двухфазных). В выборке содержатся все вышеназванные характеристики каждого из 12 отведений. Помимо этого каждой записи соответствует класс, к которому она относится. Например, первый класс соответствует записям здоровых людей, второй - ишемической болезни сердца а пятый - синусовой тахикардии. Всего таких классов 12.

2. Применение байесовских сетей

2.1. Описание байесовского классификатора

В медицинском контексте, условная вероятность – это вероятность того, каким будет заключение C , учитывая некоторые свидетельства / наблюдения, E , где существует зависимость между C и E . Эта вероятность обозначается как

$P(C|E)$, где

$$P(C|E) = \frac{P(E|C) \times P(C)}{P(E)} \quad (1)$$

Теорема Байеса - метод нахождения апостериорной вероятности с использованием условной

$$P(E|C) = \frac{P(C|E) \times P(E)}{P(C)} = \frac{P(C, E)}{P(C)} \quad (2)$$

Этот условная связь позволяет исследователю получить информацию о вероятности C или E с известным результатам другого. Теперь рассмотрим сложную проблему с n переменными, где связь между ними не известна для целей прогнозирования

выходной переменной одного класса (например, узел 1 на рисунке 3).

Если все переменные относились к одному общему распределению, эквивалентному всем узлам, которые являются родителями первого уровня, число возможных комбинаций переменных будет равно $(2^n - 1)$.

Для каждой комбинации, должно быть задействовано достаточное количество образцов, чтобы получить реальную оценку вероятности. Это приводит к необходимости использования очень большого количества данных. Если зависимость отношений между переменными может быть определена, то в результате независимые переменные будут удаляться, и меньше узлов будет находиться рядом с целевым узлом. Эти удаления родительских узлов приводят к значительному снижению количества комбинаций переменных, тем самым уменьшая количество необходимых данных. При наличии пропусков в данных условная вероятность определяется с помощью переменных, которые присутствуют. Способность байесовских сетей справиться с пропусками в данных и ее низкие требования к качеству информации, полученной на основе априорных переменных, являются ее основными преимуществами. [M. Wiggins and others, 2008]

Для этого исследования, были разработаны два метода сетей: первый с помощью классификатора поиска «a greedyhill-climbsearch» и второй, основанный на эволюционных вычислениях с использованием классификатора. Была измерена производительность обоих методов и построено сравнение с точностью классификации, полученной обычным байесовским классификатором.

2.2. Применение генетического алгоритма для выделения наиболее весомых признаков

При решении комбинаторной задачи исследования байесовской сети также может быть использован генетический алгоритм. Общий вид усовершенствованного генетического алгоритма показан на рисунке 3.

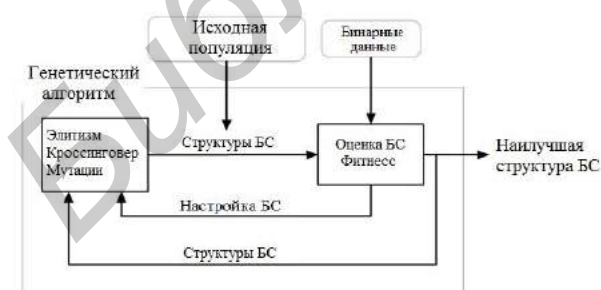


Рисунок 3 – Общий вид усовершенствованного генетического алгоритма

Алгоритм может быть использован для построения новых структур по результатам, полученным для начальной структуры. Итерационные случайные мутации и кроссоверы сетей внутри популяции проверяются и наиболее приспособленные из них сохраняются для

следующих поколений. После прохода поколения популяция эволюционирует, оставляя наилучшие структуры. Плохие структуры удаляются. Улучшение достигнуто в том числе потому, что структура полученной сети является динамической. Это позволяет реализовать метод интеллектуального построения модели без исчерпывающего поиска всех возможных комбинаций структуры узлов. [Ernest w. Lau and others, 2000]

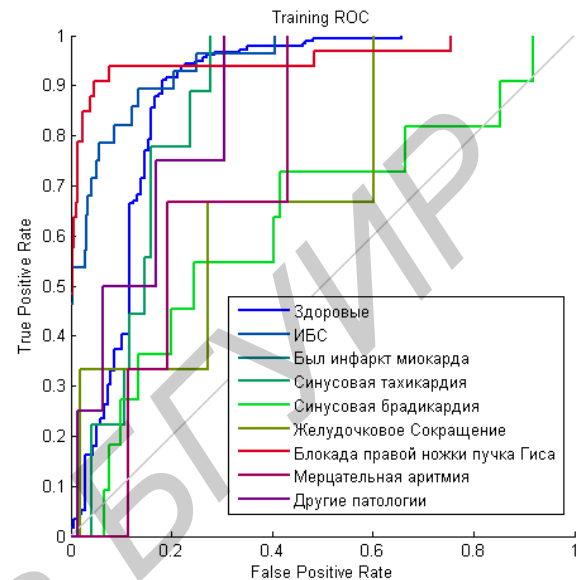


Рисунок 4- ROC-кривые, построенные для классификации патологий по ЭКГ с использованием байесовской сети и генетического алгоритма

3. Анализ полученных результатов

В ходе проведения исследования были получены достаточно интересные и многообещающие результаты. С применением генетического алгоритма была построена улучшенная Байесовская сеть. Она показала довольно хорошие результаты на некоторых из классов, представленных в выборке. Результаты применения улучшенной байесовской сети представлены на рисунке 4. Были выделены 9 классов, на которых классификатор работает лучше всего и для них были построены ROC-кривые. Наиболее приятными являются неплохие результаты при определении ишемической болезни сердца (ИБС), которая является основной причиной смертности в мире и следовательно наиболее опасным из ССЗ. Также получены хорошие результаты для блокады правой ножки пучка Гиса, синусовой тахикардии и класса здоровых людей.

Заключение

Во время проведения данного исследования была подтверждена возможность применения алгоритмов машинного обучения, а именно байесовских сетей, в автоматизированном анализе ЭКГ. В совокупности с возникновением новых технических средств, позволяющих упростить считывание ЭКГ и распространением доступных

вычислительных машин достаточно высокой мощности, это позволяет рассматривать совершенно новый подход к профилактике ССЗ.

Этот проект был разработан при поддержке
IHSTOP IASA Outsourcing Training and Production
Centre "ITSEA".

Библиографический список

[ERNEST W. LAU and others, 2000] ERNEST W. LAU. The Bayesian Approach Improves the Electrocardiographic Diagnosis of Broad Complex Tachycardia / ERNEST W. LAU [and others] // PACE 2000; 23 [Pt. I]: 1519–1526

[M. Wiggins and others, 2008] Evolving a Bayesian classifier for ECG-based age classification in medical applications / M. Wiggins [and others]; Applied Soft Computing 8 (2008) 599–608

[Парадинец, 2008] Парадинец, А. В. Генетический алгоритм диагностики сердечнососудистых заболеваний / А. В. Парадинец // Сборник работ 59-й научной конференции студентов и аспирантов БГУ 2002 год

[Fridman R., Sagan V., 2015] Fridman R., Sagan V. Approach to ischemia detection using artificial neural network / Fridman R., Sagan V. // Kyiv-Cherkasy, Ukraine. – Cherkasy, 2015. – P. 155-156.

[Chereda H.O., Nikolaiev S.S., 2015] Chereda H.O., Nikolaiev S.S. Architecture of automatic machine learning system for heart pathologies detection using ECGs / Chereda H.O., Nikolaiev S.S. // SAIT 2015, Kyiv, Ukraine, June 22-25, 2015. Proceedings. – ESC "IASA" NTUU "KPI", 2015. – P. 27-28.

APPLICATION OF BAYESIAN NETWORKS FOR ANALYSIS OF ECG

Fridman R., Sagan V.Y.,

*National Technical University of Ukraine "Kyiv
Polytechnic Institute"
Institute for Applied System Analysis,
Kyiv, Ukraine*

Vitaliysagan14@gmail.com
Ronfridman1995@gmail.com

Introduction

This paper describes the research in the field of automated analysis of the heart by electrocardiograms (ECGs) using machine learning algorithms. In particular, it describes the results of the application of Bayesian networks and naive Bayesian classifier in detection of the presence of cardiac pathologies. The results that were obtained indicate that this approach can be applied to the task. The best results were obtained by using the improved Bayesian network.

Main Part

Machine learning methods cannot be directly applied to the raw ECG signal and require specific ECG features to make a decision how healthy the patient is.

ECG signal may contain from dozens to a couple of hundreds of different features that capture some specific characteristics of the ECG complex curve and contain information about heart pathologies.

During model learning phase the most informative features are selected among all the features and are

saved for the model training. All the rest features are truncated. During working phase all extracted features from each complex for each lead are stored in the vectors. Then each given vector for each time frame (for e.g. 5 complexes) is packed as the row of the feature matrix. Then this matrix is used further as an input for machine learning methods.

More than 200 heart pathologies exist and they have different influence into ECG. Some of them (like arrhythmias) influence on temporal pattern of ECG activity and are found with temporal pipeline of the system. Other pathologies influence on complexes' parameters (influence complexes' parameters) and are detected by pattern matching pipeline.

To achieve better performance a set of models which consist of Bayesian network and a greedy hill-climb search is applied to feature matrix.

Conclusion

This study has confirmed the possibility of application of machine learning algorithms, namely, Bayesian networks, for automated ECG analysis. In conjunction with the emergence of new technical means to simplify the collection of the ECG data and the spread of affordable computing devices of sufficiently high power, it allows us to consider a new approach to prevention of cardiovascular diseases.



УДК 004.8

ПОСТРОЕНИЕ БАЗЫ ЗНАНИЙ ДЛЯ ХРАНЕНИЯ И АНАЛИЗА БИМЕДИЦИНСКИХ КТ ИЗОБРАЖЕНИЙ ПОЯСНИЧНОГО ОТДЕЛА ПОЗВОНОЧНИКА

Курочка К.С., Карабчикова Е.А.

*Гомельский государственный технический университет имени П.О.Сухого,
г. Гомель, Республика Беларусь*

kurochka@gstu.by

karabchicova.ru@mail.ru

Предлагается архитектура базы знаний для осуществления хранения и анализа снимков компьютерной томографии нижнего отдела позвоночника, а также алгоритмы построения уникальных дескрипторов.

Ключевые слова: база знаний; медицинская информационная система; поясничный отдел позвоночника.

Введение

В настоящее время лечение заболеваний опорно-двигательного аппарата невозможно без использования и анализа соответствующих биомедицинских изображений. Отдельное место среди которых занимают результаты компьютерной томографии (КТ), и, в особенности, компьютерная томография позвоночника человека [1].

Помимо типовых задач организации хранения и поиска биомедицинских изображений, при лечении заболеваний позвоночника выделяется задача построения биомеханической модели на основе аппарата математического моделирования. Такие биомеханические модели могут быть использованы для выбора оптимальной фиксирующей конструкции, вариантов крепления её к позвонку, прогнозирования влияния имплантируемых элементов на позвоночный отдел пациента. При этом можно проводить анализ состояния позвоночника, варьируя параметры (свойства) исследуемых элементов, таких как плотность костной ткани, модуль упругости, геометрические параметры позвоночника, что позволяет оценить развитие различных заболеваний и патологий. Например, таких как остеопороз. Для этого необходимо на основе биомеханической модели иметь возможность оценить напряжённо-деформированное состояние (НДС) элементов позвоночника в различных состояниях при функциональных нагрузках [2].

Эффективное решение комплекса подобных задач возможно посредством проектирования и

разработки базы знаний, обрабатывающей результаты компьютерной томографии позвоночника человека как набора изображений, так и формирующей виртуальную биомеханическую модель, которую можно будет использовать для определения напряжённо-деформированного состояния в различных условиях.

В связи с этим к проектируемой базе знаний можно определить следующие требования:

- а) хранение данных компьютерной томографии как серии изображений в формате *DICOM (Digital Imaging and Communications in Medicine)*;
- б) построение уникальных дескрипторов на основе данных компьютерной томографии;
- в) построение виртуальной биомеханической модели;
- г) осуществление поиска определенного случая или подобных с заданной степенью точности как по дескрипторам КТ, так и по параметрам биомеханической модели;
- д) использование биомеханической модели для моделирования НДС и хранения результатов.

1. Архитектура проектируемой базы знаний

На рисунке 1 представлена архитектура предлагаемой базы знаний, которая включает в себя следующие компоненты:

- программный комплекс;
- база данных;
- интерфейс взаимодействия со сторонним программным обеспечением для моделирования



Рисунок 1 – Архитектура базы знаний

различных случаев НДС.

Программный комплекс представлен набором правил и внутренних алгоритмов. Набор правил необходим для реализации обработки данных или иными словами перебора данных в соответствии с существующими в базе описателями. А набор внутренних алгоритмов осуществляет формирование уникальных дескрипторов базы знаний, реализуют взаимодействие с базой данных и сторонним программным обеспечением.

База данных обеспечивает хранение информации. В ней содержатся исходные изображения компьютерной томографии, аппроксимационная узловая поверхность поясничного отдела позвоночника и его биомеханическая модель, набор различных описателей.

Взаимодействия со сторонним программным обеспечением необходимо для осуществления моделирования различных вариантов напряженно-деформируемого состояния нижнего отдела позвоночника. Посредством взаимодействия программного комплекса с генераторами конечноэлементных сеток, таких как *Netgen Mesh Generator* [3], из аппроксимационной узловой поверхности создается биомеханическая модель. Она представлена набором конечных элементов для каждого из которых установлены физические параметры (модуль Юнга, коэффициент Пуассона). Впоследствии данная модель взаимодействует с программными системами конечно-элементного анализа и результаты моделирования напряженно-деформируемого состояния сохраняются в базу знаний.

2. Создание уникальных дескрипторов

Отличительной особенностью проектируемого комплекса является использование в качестве дескриптора аппроксимационной узловой поверхности нижнего отдела позвоночника.

На основе КТ изображений, хранящихся в специализированном медицинском формате DICOM, содержащем рентгеновскую плотность, которая зависит от физической плотности тканей, происходит синтез необходимой анатомической структуры – поясничного отдела позвоночника.

С целью подавления помех, которые возникают в изображениях на этапе их регистрации аппаратными средствами или при передаче сигналов по линиям связи, производится цифровая обработка изображения [4]. На каждом обработанном снимке происходит распознавание костной ткани позвоночника и определяется ее геометрия. Для этих целей предлагается определить границы позвонка на каждом из изображений компьютерной томографии. Поскольку цвет на снимке представляет собой плотность изображенного объекта, нахождение костной ткани осуществляется путем подбора оптимального порогового значения и определения по нему принадлежности пикселей изображения к требуемому объекту.

Для получения четкого контура объектов на обработанном снимке используется алгоритм контурной сегментации – оператор Лапласа, реализованный средствами открытой библиотеки *OpenCV*. Применяя оператор Лапласа на каждое изображение, находят контуры позвоночника, которые в подавляющем большинстве случаев представляют собой замкнутые кривые. На основании полученной геометрии определяются опорные точки путем наложения на каждый слой мелкой конформной сетки. Пересечения узлов сетки с границами позвоночника обозначаются как опорные точки. В завершении построения опорные точки соседних слоев объединяются между собой в конечные элементы.

3. Реализация поиска

Эффективность работы с базой знаний определяется вводимыми дескрипторами, поскольку они являются ключевыми элементами при

получении знаний. На основании существующих дескрипторов и их комбинаций формируются различные запросы поиска.

Условно дескрипторы данной системы можно разделить на три группы.

Первая группа – информационные. Представляют собой описатели, используемые при поиске информации, содержащейся в файлах DICOM без каких-либо преобразований. Как правило это либо нахождение результатов одного конкретного медицинского обследования, либо группы схожих, путем выборки по статическим полям. Например, результаты всех обследований определенного пациента, случаи с определенным заболеванием и связанные с ним патологии, случаи с учетом возрастной группы, половой принадлежности и других обобщенных параметров.

Вторая группа – вычислительные. Дескрипторы, вычисляемые один раз на этапе добавления случая в базу знаний. Формируются для каждого слоя КТ изображений отдельно, но помимо этого дополнительно могут вычисляться на основе геометрии имеющейся аппроксимационной узловой поверхности или биомеханической модели, путем построения горизонтальных срезов.

Для достижения адекватности получаемых результатов при построении данной группы дескрипторов должно осуществляться масштабирование сопоставляющихся изображений или объектов.

Предлагаемые варианты дескрипторов вычислительной группы:

- опорные точки, вычисляемые по алгоритму, предлагаемому авторами в пункте 3;
- отношение площади к периметру;
- метод градиентного анализа контурного изображения.

Поиск информации на основании опорных точек, по которым выполняется построение аппроксимационной узловой поверхности, осуществляется путем сравнения их взаимного расположения. Два снимка с выделенными опорными точками сводятся к единой системе координат и сравниваются по условию:

$$\forall_{i,j} \exists |x_{i,j} - y_{i,j}| < \varepsilon, \quad (1)$$

где ε - заданная точность.

Сущность дескриптора “отношение площади к периметру” заключается в вычислении некоторого числа N :

$$N = \frac{S}{P}, \quad (2)$$

где S – количество пикселей на изображении, образующее площадь анатомической структуры

(рисунок 2), P – количество пикселей, образующее ее периметр (рисунок 3).

Сопоставление данного числа различных снимков позволяет судить о геометрической схожести сравниваемых объектов.



Рисунок 2 – Пример представление площади на КТ снимке нижнего отдела позвоночника.

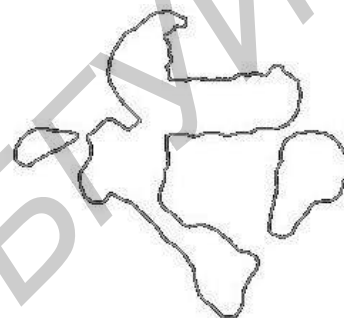


Рисунок 3 – Пример представление периметра на КТ снимке нижнего отдела позвоночника

Суть дескриптора на основе градиентного анализа контурного изображения заключается в следующем. Производится обход по контуру в результате которого выявляются “особые точки”, в которых вектор градиента меняет свое направление. На основе двух точек определяются текущие знаки приращения Δx и Δy по двум соответствующим координатным осям x и y . Если хотя бы одно из приращений в процессе перехода к следующей точке меняет знак и выполняется хотя бы одно условие:

$$|\Delta x| > \varepsilon, \quad (3)$$

или

$$|\Delta y| > \varepsilon, \quad (4)$$

то есть происходит достаточно значительный скачок по одной из координат, то данная точка определяется как “особая”. Преимуществом такого подхода является снижение объёма вычислений за счёт использования небольшого числа особых точек вместо всех точек контура. А добившись одинакового количества “особых точек” на двух сравниваемых изображениях, можно провести анализ их расположений на основании условия 1.

Третья группа дескрипторов – сложные сформированные. Данные дескрипторы – это результаты моделирования различных случаев

напряженно-деформируемого состояния биомеханической модели. К одной и той же конкретной биомеханической модели прилаживаются различные варианты механических нагрузок и варьируются параметры. Как результат, для конкретного случая создается своеобразная база поведения исследуемого нижнего отдела позвоночника. Задавая при поиске определенные варианты параметров моделирования и результата, можно осуществлять сравнение схожих случаев.

4. Пример построения аппроксимационной поверхности

На рисунке 4 приведен пример исходного КТ изображения.

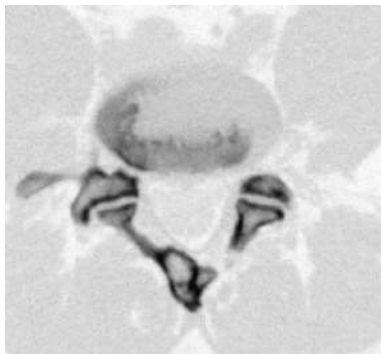


Рисунок 4 – Пример исходного КТ изображения

Примеры определения геометрии на КТ снимке в результате распознавания костной ткани и применения оператора Лапласа приведены на рисунках 2 и 3 соответственно. Результаты нахождения опорных точек представлены на рисунке 5.



Рисунок 5 – Результат определения опорных точек

На рисунке 6 представлена полученная аппроксимационная поверхность нижнего отдела позвоночника.



Рисунок 6 – Построенная аппроксимационная поверхность нижнего отдела позвоночника

Заключение

Предлагаемая архитектура базы знаний и алгоритмы обработки данных могут быть использованы для построения информационных медицинских систем в области лечения заболеваний опорно-двигательного аппарата. Достоинством подхода является возможность оценки НДС элементов позвоночника в различных состояниях при типичных функциональных нагрузках.

Библиографический список

- [Чуйко, 2014] Чуйко, А.Н. Приближенный анализ анатомии, механических характеристик и напряженно-деформированного состояния позвоночника человека / А.Н. Чуйко // Травма. – 2014. – № 6. – Том 15
- [Зенкевич, 1976] Зенкевич, О. Метод конечных элементов в технике. – М.: Мир, 1975. – 541 с.
- [Netgen, 2015] Netgen Mesh Generator [Электронный ресурс]. – Режим доступа: <http://sourceforge.net/projects/netgen-mesher/>, свободный.
- [Красильников, 2011] Красильников, Н. Н. Цифровая обработка 2D- и 3D-изображений / Н. Н. Красильников // БХВ-Петербург, 2011.

KNOWLEDGE BASE DESIGN FOR STORAGE AND ANALYSIS BIOMEDICAL CT SCAN OF LUMBAR SPINE

Kurochka K.S., Karabchikova E.A.

*P.O. Sukhoi State Technical University of Gomel,
Gomel, Republic of Belarus*

kurochka@gstu.by

karabchikova.ru@mail.ru

Introduction

Treatment of musculoskeletal system diseases is impossible without the use and analysis of relevant biomedical images. Apart from typical tasks of organizing storage and search biomedical imaging in the treatment of the spine diseases, allocated the task of building a biomechanical model based on mathematical modeling. An effective solution of such complex problems is designing and developing the knowledge base.

Main Part

The architecture of the knowledge base for storing and analyzing CT scans of the lumbar spine and algorithms for creating unique descriptors are suggested.

Conclusion

The proposed architecture of the knowledge base and data processing algorithms, that can be used for the building medical information in the treatment of musculoskeletal system diseases. The advantage of the approach is the opportunity to evaluate stress-strain state of parts of the human spine under typical functional loads.

АВТОРСКИЙ УКАЗАТЕЛЬ

А			
Аверкин А.Н.	319	Житко В.А.	383
Андреев И.А.	191	З	
Андрущенко В.Б.	233	Заболеева-Зотова А.В.	401, 561
Афанасьева Т.В.	271, 301, 305	Заварзин Д.В.	305
Б		Загорюлько Г.Б.	61
Бабамухамедова М.З.	565	Загорюлько Ю.А.	61
Балагура И.В.	233	Зубцов Р.О.	83
Барбанова Е.Ю.	517	Зуенко А.А.	573
Барлыбаев А.Б.	197	И	
Бегимтай У.Х.	393	Иванова А.В.	45
Бексаева Е.А.	191	Ивашенко В.П.	145
Бикмуллина И.И.	213	К	
Богатиков В.Н.	539	Каешко А.И.	151
Бойко В.К.	535	Казак Е.А.	545
Бойко И.М.	115	Карабчикова Е.А.	583
Боргест Н.М.	421	Карпаев С.А.	441
Бурак Т.И.	569	Карпов А.А.	477
Бурдо Г.Б.	449	Карпов В.Э.	471
В		Качан Я.С.	397
Вагин В.Н.	275	Клейн В.В.	191
Валькман Ю.Р.	165	Клещев А.С.	31
Виноградов Г.П.	457, 539	Клюшин А.Ю.	465
Виноградова Н.Г.	457	Кобринский Б.А.	89
Вишняков В.А.	545	Коваленко Т.В.	171
Власов С.А.	427	Козлов О.А.	511
Войтукевич Ю.А.	535	Коровин М.Д.	427
Воронин Ю.А.	539	Королев Ю.И.	283
Г		Корончик Д.Н.	155
Гаврилова Т. А.	27	Коршунов С.А.	355
Галочкин М.В.	359	Крощенко А.А.	313
Гатауллин Р.Р.	389	Кривальцэвіч А.В.	397
Гецэвіч Ю.С.	397	Крючков Б.И.	477
Гильмуллин Р.А.	389	Кудрявцев Д. В.	27
Гладун А.Я.	239	Кузенбаев Б.А.	527
Глоба Л.С.	51, 57	Куликов А.М.	177
Головко В.А.	313	Кулинич А.А.	259
Горбачёв Н.Н.	531	Курбатов С.С.	159
Гракова Н.В.	93	Куриленко И.Е.	289
Грибова В.В.	45, 327	Курманбакиев М.И.	371
Грищенко М.А.	209	Курочка К.С.	583
Губаревич А.В.	125	Л	
Гулякина Н.А.	275	Ландэ Д.В.	171, 233, 253
Д		Ларин С.Н.	435, 441
Давыденко И.Т.	93, 107	Лахин О.И.	487, 493
Данилин Н.С.	483	Левчик А.В.	377
Димитров Д.М.	483	Лобанов Б.М.	383
Додонов А.Г.	171, 253	Лобзин А.П.	159
Додонов В.А.	253	Лукашевич М.М.	569
Долбин А.В.	401	Лукашевич Н.В.	377
Донсков А.В.	487	Лунецкас А.А.	359
Дородных Н.О.	203, 209	Лысы С.И.	397
Дошанова М.Ю.	565	Лявшук В.Е.	535
Е		М	
Егерев И.А.	445	Мальковский М.Г.	183
Еремеев А.П.	275, 283	Маракуліна П.А.	397
Ефименко И.В.	223	Массель А.Г.	65, 71
Ж		Массель Л.В.	65, 71
Желепов А.С.	191	Меньшаков П.А.	411
Жилякова Л.Ю.	323	Митянок В.В.	405
		Михайлов Ю.Ф.	511

Мишурова Н.В.	487	Скорюпина Е.Г.	487
Моросин О.Л.	125	Славский В.В.	549
Москаленко Ф.М.	39	Смирнов С.В.	83
Мошкин В.С.	77, 265	Соловьев С.Ю.	183
Мурашко И.А.	411	Сорокин О.Л.	431
Мутовкина Н.Ю.	465	Соснин П.И.	359
Мухамедшин Д.Р.	371	Степашко П.В.	165
Н		Стратнев П.Ю.	553
Наместников А.М.	415	Сысоев Д.В.	487
Нгуен Тхи Минь Ву	295	Т	
Невзорова О.А.	371	Тарасов В.Б.	331
Никитин Н.А.	561	Татур М.М.	145
Ниязова Р.С.	527	Тимченко В.А.	39
Новогрудская Р.Л.	57	Тузикова А.В.	187
О		Турмаганбетова Ш.К.	557
Орлова А.А.	421	Тякунов А.С.	549
Орлова Ю.А.	401, 561	У	
П		Усов В.М.	477
Палюх Б.В.	445, 465	Ф	
Петров С.В.	535	Федорищев Л.А.	327
Пивоварчик О.В.	521	Федотова А.В.	93, 449
Плесневич Г.С.	295	Филиппов А.А.	77
Погорельый С.Д.	247	Фоминых И.Б.	349
Полников А.С.	493	Фридман Р.	579
Порядин А.Е.	367	Х	
Потебня А.В.	247	Харламов А.А.	177
Р		Хахалин Г.К.	159
Редькина К.В.	493	Хацкевич М.В.	313
Ровба Е.А.	535	Хорошевский В.Ф.	223
Рогушина Ю.В.	239	Хусаинов А.Ф.	507
Розалиев В.Л.	401, 561	Ш	
Романко О.Р.	309	Шалаев Д.О.	77
Русецкий К.В.	507	Шалфеева Е.А.	31
Рыбин В.М.	497	Шарипбай А.А.	197, 527
Рыбина Г.В.	497	Шихнабиева Т.Ш.	503
С		Шункевич Д.В.	125, 137
Сабиров И.Х.	483	Э	
Сабыров Т.С.	197	Эргашев Ф.А.	565
Савкин Л.В.	219	Ю	
Саган В.Ю.	579	Юрин А.Ю.	203, 209
Сапунков А.А.	301	Юрыгина Ю.С.	487
Святкина М.Н.	125, 331	Я	
Семенова В.А.	83	Ярушев С.А.	319
Сергиенко Е.С.	93, 497	Ярушкина Н.Г.	77, 265
Серков И.П.	191		
Сибирев И.В.	271		
Сидоркина И.Г.	367, 431		

AUTHOR INDEX

A			
Afanasjeva T.V.	271, 301, 305	Iurygina I. S.	487
Andreev I.A.	191	Ivanova A.V.	45
Andrushchenko V.B.	233	Ivashenko V.P.	145
Averkin A.N.	319	K	
B		Kachan E.S.	397
Babamukhamedova M.Z.	565	Karabchikova E.A.	583
Balagura I.V.	233	Karpaev S.A.	441
Barabanova E. Yu.	517	Karpov A.A.	477
Barlybayev A.B.	197	Karpov V.E.	471
Begimtay U.H.	393	Kayeshko A.I.	151
Beksaeva E.A.	191	Kazak E.A.	545
Bikmullina I.I.	213	Khakhalin G.K.	159
Bogatikov V.N.	539	Kharlamov A.A.	177
Boika V.K.	535	Khoroshevsky V.F.	223
Borgest N.M.	421	Khusainov A.F.	507
Boyko Igor	115	Klein V.V.	191
Burak T.I.	569	Kleshev A.	31
Burdo G.B.	449	Klyushin A. Yu.	465
C		Kobrinskii B.A.	89
Curbatow S.S.	159	Korolev Y.I.	283
D		Koronchik D. N	155
Danilin N.S.	483	Korovin M.D.	427
Davydenko I.T.	93, 107	Korshunov S.A.	355
Dimitrov D.M.	483	Kovalenko T.V.	171
Dodonov A.G.	171, 253	Kozlov O.A.	511
Dodonov V.A.	253	Krivaltsevich A.V.	397
Dolbin A.V.	401	Kroshchanka A.A.	313
Donskov A.V.	487	Kryuchkov B.I.	477
Dorodnykh N.O.	203, 209	Kudryavtsev D.V.	27
Doshchanova M. Yu.	565	Kulikov A.M.	177
E		Kulinich A. A.	259
Efimenko I.V.	223	Kurilenko I.E.	289
Egereva I.A.	445	Kurmanbakiev M.I.	371
Eremeev A.P.	275, 283	Kurochka K.S.	583
Ergashev F.A.	565	Kuzembaev B.A.	527
F		L	
Fedorischev L.A.	327	Lakhin O. I.	487, 493
Fedotova A.V.	93, 449	Lande D.V.	171, 233, 253
Filippov A.A.	77	Larin S. N.	435, 441
Fominykh I.B.	349	Levchik A.V.	377
Fridman R.	579	Liaushuk U.J.	535
G		Lobanov B.M.	383
Galochkin M.V.	359	Lobzin A.P.	159
Gataullin R.R.	389	Loukachevitch N.V.	377
Gavrilova T.A.	27	Lukashevich M.M.	569
Gilmullin R.R.	389	Lunecaks A.A.	359
Gladun A.	239	Lysy S.I.	397
Globa L.S.	51, 57	M	
Goloyko V.A.	313	Malkovsky M.G.	183
Gorbatchev N.N.	531	Marakulina P.A.	397
Grakova N.V.	93	Massel A.G.	65, 71
Gribova V.V.	45, 327	Massel L.V.	65, 71
Grishenko M.A.	209	Menshakov P.A.	411
Guliakina N.A.	275	Mikhailov J.F.	511
H		Mishurova N. V.	487
Hackevich M.V.	313	Mitsianok V.V.	405
Hetsevich Y.S.	397	Morosin O.L.	125
Hubarevich A.V.	125	Moshkin V.S.	77, 265
I		Moskalenko Ph.M.	39
		Mukhamedshin D.R.	371

Murashko I.A.	411	Sibirev I.V.	271
Mutovkina N.Yu.	465	Sidorkina I.G.	367, 431
N		Skoryupina E.G.	487
Namestnikov A.M.	415	Slavsky V.V.	549
Nevzorova O.A.	371	Smirnov S.V.	83
Nguyen Thi Minh Vu	295	Soloviev S.Y.	183
Nikitin N.A.	561	Sorokin O.L.	431
Niyazova R.S.	527	Sosnin P.I.	359
Novogradskaya R.L.	57	Stepashko P.V.	165
O		Stratnev P.U.	553
Orlova A.A.	421	Svyatkina M.N.	125, 331
Orlova Y.A.	401, 561	Sysoev D.V.	487
P		T	
Palyukh B.V.	445, 465	Tarassov V.B.	331
Piatrou S.V.	535	Tatur M.M.	145
Pivovarchyk O.	521	Timchenko V.A.	39
Plesniewicz G.S.	295	Turmaganbetova Sh.K.	557
Pogorilyy S.D.	247	Tuzikova A.V.	187
Polnikov A.	493	Tyakunov A.S.	549
Poryadin A.Y.	367	U	
Potebnia A.V.	247	Usov V.M.	477
R		Vagin V.N.	275
Redkina K.	493	Vaitukevich Y.A.	535
Rogushina J.	239	Valkman Y.R.	165
Romanko A.R.	309	V	
Rouba Y.A.	535	Vinogradov G.P.	457, 539
Rozaliev V.L.	401, 561	Vinogradov N.G.	457
Ruseckiy K.V.	507	Vishniakou U.A.	545
Rybin V.M.	497	Vlasov S.A.	427
Rybina G.V.	497	Voronin Yu. A.	539
S		Y	
Sabirov I.H.	483	Yarushev S.Y.	319
Sabyrov T.S.	197	Yarushkina N.G.	77, 265
Sagan V.Y.	579	Yurin A.Yu.	203, 209
Sapunkov A.A.	301	Z	
Savkin L.V.	219	Zaboleeva-Zotova A.V.	401, 561
Semyonova V.A.	83	Zagorulko G.B.	61
Sergienko E.S.	93, 497	Zagorulko Yu.A.	61
Serkov I.P.	191	Zavarzin D.V.	305
Shalaev D.O.	77	Zhelepov A.S.	191
Shalfeeva E.	31	Zhilyakova L.Yu.	323
Sharipbay A.A.	197, 527	Zhitko V.A.	383
Shikhnabieva T.Sh	503	Zubtcov R.O.	83
Shunkevich D.V.	125, 137	Zuenko A.A.	573

Научное издание

Открытые семантические технологии проектирования интеллектуальных систем

OSTIS-2016

Open Semantic Technologies for Intelligent Systems

**МАТЕРИАЛЫ
VI МЕЖДУНАРОДНОЙ
НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ**

(Минск, 18–20 февраля 2016 года)

В авторской редакции
Ответственный за выпуск *В. В. Голенков*
Компьютерная верстка *Н. В. Гракова, А. В. Губаревич*

Подписано в печать **10.02.2015**. Формат 60×84 1/8. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 72,54. Уч.-изд. л. 63,3. Тираж 140 экз. Заказ 30.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».

Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,

№2/113 от 07.04.2014, №3/615 от 7.04.2014.

ЛИ №02330/264 от 14.04.2014.

220013, Минск, П. Бровки, 6

Библиотека БГУИР



VII международная научно-техническая конференция «Открытые семантические технологии проектирования интеллектуальных систем»

Open Semantic Technologies for Intelligent Systems

OSTIS-2017

16 – 18 февраля 2017 г. Минск. Республика Беларусь

И Н Ф О Р М А Ц И О Н Н О Е П И С Ь М О

Приглашаем принять участие в VII Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2017).

Конференция пройдет в период с **16 по 18 февраля 2017** года в Белорусском государственном университете информатики и радиоэлектроники, г. Минск, Республика Беларусь.

Рабочие языки конференции: русский, белорусский, английский.

ОСНОВНЫЕ ОРГАНИЗАТОРЫ КОНФЕРЕНЦИИ

- Министерство образования Республики Беларусь
- Российская ассоциация искусственного интеллекта (РАИИ)
- Белорусский государственный университет информатики и радиоэлектроники (БГУИР)
- Государственное учреждение «Администрация Парка высоких технологий» (Республика Беларусь)

НАПРАВЛЕНИЯ РАБОТЫ КОНФЕРЕНЦИИ:

- *Принципы, лежащие в основе семантического представления знаний, и их унификация. Типология знаний и особенности семантического представления различного вида знаний и метазнаний. Связи между знаниями и отношения, заданные на множестве знаний. Семантическая структура глобальной базы знаний, интегрирующей различные накапливаемые знания*
- *Языки программирования, ориентированные на параллельную обработку семантического представления баз знаний*
- *Модели решения задач, в основе которых лежит обработка знаний, осуществляемая непосредственно на уровне семантического представления обрабатываемых знаний. Семантические модели информационного поиска, интеграции знаний, анализа корректности и качества баз знаний, сборки информационного мусора, оптимизации баз знаний, дедуктивного и индуктивного вывода в базах знаний, правдоподобных рассуждений, распознавания образов, интеллектуального управления. Интеграция различных моделей решения задач*
- *Семантические модели восприятия информации о внешней среде и отображения этой информации в базу знаний*
- *Семантические модели мультимодальных пользовательских интерфейсов интеллектуальных систем, в основе которых лежит семантическое представление используемых ими знаний, и унификация этих моделей*
- *Семантические модели естественно-языковых пользовательских интерфейсов интеллектуальных систем. Структура семантического представления лингвистических баз знаний, описывающих естественные языки и обеспечивающих решение задач понимания естественно-языковых текстов и речевых сообщений, а также задач синтеза естественно-языковых текстов и речевых сообщений, семантически эквивалентных заданным фрагментам баз знаний*
- *Интегрированные комплексные логико-семантические модели интеллектуальных систем, основанные на семантическом представлении знаний, и их унификация*
- *Различные технические платформы и варианты реализации интерпретаторов унифицированных логико-семантических моделей интеллектуальных систем, основанных на семантическом представлении знаний*
- *Средства и методы, основанные на семантическом представлении знаний и ориентированные на проектирование различных типовых компонентов интеллектуальных систем (баз знаний, программ, решателей задач, интерфейсов)*
- *Средства и методы, основанные на семантическом представлении знаний и ориентированные на комплексное проектирование различных классов интеллектуальных систем (интеллектуальных справочных систем, интеллектуальных обучающих систем, интеллектуальных систем управления, интеллектуальных робототехнических систем, интеллектуальных систем поддержки проектирования и др.)*
- *Прикладные интеллектуальные системы, основанные на семантическом представлении используемых ими знаний*

ЦЕЛЬ И ФОРМАТ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ

Целью конференции является обсуждение проблем создания **открытой комплексной семантической технологии компонентного проектирования интеллектуальных систем**. Этим определяется и формат её проведения, предполагающий (1) пленарные доклады, (2) секционные заседания; (3) круглые столы, посвященные обсуждению различных вопросов создания указанной технологии; (4) выставочные презентации докладов.

Выставочная презентация докладов даёт возможность каждому докладчику продемонстрировать результаты своей разработки на выставке. Формат проведения конференции предполагает точное время начала каждого доклада и точное время его выставочной презентации.

Важнейшей задачей конференции является привлечение к её работе не только учёных и аспирантов, но и студенческой молодежи, интересующейся проблемами искусственного интеллекта, а также коммерческих организаций, готовых сотрудничать с научными коллективами, работающими над интеллектуальными системами и созданием современных технологий и их проектированием.

УСЛОВИЯ УЧАСТИЯ В КОНФЕРЕНЦИИ

В конференции имеют право участвовать все те, кто интересуется проблемами искусственного интеллекта, а также коммерческие организации, готовые сотрудничать с научными коллективами, работающими над созданием современных технологий проектирования интеллектуальных систем.

Для участия в конференции OSTIS-2017 необходимо до 1 декабря 2016 года на электронную почту конференции ostisconf@gmail.com отправить:

- **статью** для публикации в Сборнике материалов конференции OSTIS-2017. Статья на конференцию должна быть оформлена в соответствии с правилами оформления публикуемых материалов;
- **заявку доклада** на конференцию OSTIS-2017. Каждое поле заявки обязательно для заполнения, в том числе указание того автора, кто будет представлять доклад. Заполняя регистрационную форму, Вы подтверждаете согласие на обработку Оргкомитетом конференции персональных данных, публикацию статей и информации об авторах в печатном и электронном виде. В заявке доклада должна содержаться информация по каждому автору. К заявке доклада должны быть прикреплены **цветные фотографии** всех авторов статьи (это необходимо для публикации Программы конференции).

Если доклад представляется на конкурс докладов молодых учёных или на конкурс программных продуктов молодых учёных, это должно быть явно указано в заявке доклада.

Отбор статей для публикации в Сборнике и участия в работе конференции осуществляется рецензентами из числа членов Программного комитета конференции.

Заявки и статьи, оформленные без соблюдения предъявляемых требований, не рассматриваются.

До 20 января 2017 года, авторам статей, включённых в Программу конференции, направляются приглашения для участия в конференции.

Участие в конференции не предполагает организационного взноса.

ПОРЯДОК ПРЕДСТАВЛЕНИЯ НАУЧНЫХ СТАТЕЙ

Статьи (только по перечисленным выше направлениям) представляются в готовом для публикации виде (правила оформления опубликованы на сайте конференции: http://www.conf.ostis.net/images/d/d3/Правила_оформления_публикуемых_материалов_OSTIS-2017.zip). Текст статьи должен быть логически законченным и содержать новые научные и практические результаты. От одного автора допускается не более двух статей.

Оргкомитет оставляет за собой право отказать в приеме статьи в случае, если статья не будет соответствовать требованиям оформления и тематике конференции, а также, если будет отсутствовать заявка доклада, соответствующая этой статье.

КОНКУРС ДОКЛАДОВ МОЛОДЫХ УЧЁНЫХ

Среди авторов доклада, представляемого на конкурс докладов молодых учёных, могут входить учёные со степенями и званиями, но непосредственно представлять доклад должны авторы, не имеющие степеней и званий в возрасте до 35 лет.

Для того, чтобы принять участие в конкурсе научных докладов молодых учёных необходимо:

- 1) заполнить заявку на участие в конференции (http://www.conf.ostis.net/images/7/7a/Заявка_на_участие_в_OSTIS-2017.zip), в которой чётко указать своё желание принять участие в данном конкурсе;
- 2) написать статью на конференцию и отправить её по электронному адресу ostisconf@gmail.com (правила оформления публикуемых материалов размещены на сайте конференции http://www.conf.ostis.net/images/d/d3/Правила_оформления_публикуемых_материалов_OSTIS-

2017.zip);

3) лично представить доклад на конференции.

КОНКУРС ПРОЕКТОВ МОЛОДЫХ УЧЁНЫХ

Принимать участие в конкурсе проектов молодых учёных могут проекты прикладных интеллектуальных систем и систем ориентированных на поддержку проектирования интеллектуальных систем, при этом представлять проект на конкурсе должен молодой учёный в возрасте до 35 лет, не имеющие учёных степеней.

Для того, чтобы принять участие в конкурсе программных продуктов молодых учёных необходимо:

- 1) заполнить заявку на участие в конференции (http://www.conf.ostis.net/images/7/7a/Заявка_на_участие_в_OSTIS-2017.zip), в которой чётко указать своё желание принять участие в данном конкурсе;
- 2) написать статью на конференцию и отправить её по электронному адресу ostisconf@gmail.com (правила оформления публикуемых материалов размещены на сайте конференции http://www.conf.ostis.net/images/d/d3/Правила_оформления_публикуемых_материалов_OSTIS-2017.zip);
- 3) лично представить доклад на конференции;
- 4) провести выставочную презентацию, разработанного программного продукта.

КОНКУРС СТУДЕНЧЕСКИХ ПРОЕКТОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

В конкурсе студенческих проектов могут принимать участие проекты, разработчиками которых являются студенты и магистранты высших учебных заведений, консультантами и руководителями проекта могут быть лица, имеющие научную степень и звание. Для того, чтобы принять участие в данном конкурсе необходимо:

- 1) ознакомиться с положением о конкурсе студенческих проектов (http://www.conf.ostis.net/images/9/94/Положение_о_конкурсе_студенческих_проектов.pdf http://conf.ostis.net/images/9/9a/Положение_о_конкурсе_студенческих_проектов_2017.pdf);
- 2) заполнить заявку на участие в конкурсе студенческих проектов (http://www.conf.ostis.net/images/4/44/Заявка_на_участие.zip);
- 3) подготовить описание проекта (http://www.conf.ostis.net/images/9/9f/Описание_проекта.zip).
- 4) выслать заявку на участие в конкурсе и описание проекта по электронному адресу конкурса студенческих проектов: ostis.stud@gmail.com.

ПУБЛИКАЦИЯ МАТЕРИАЛОВ КОНФЕРЕНЦИИ

Оргкомитет конференции предполагает публикацию статей, отобранных Программным комитетом по результатам их рецензирования, в Сборнике материалов конференции и на официальном сайте конференции <http://conf.ostis.net>. Неимущественные права принадлежат авторам статей, поэтому публикация и распространение материалов статей на иных информационных ресурсах допускается только с согласия авторов статей.

КЛЮЧЕВЫЕ ДАТЫ КОНФЕРЕНЦИИ

<i>1 октября 2016г.</i>	начало подачи материалов для участия в конференции
<i>1 декабря 2016г.</i>	срок получения материалов для участия в конференции Оргкомитетом
<i>25 января 2017г.</i>	срок принятия решения о публикации присланных материалов и рассылки приглашений для участия в конференции и сообщение о включении статьи в Сборник материалов конференции OSTIS
<i>5 февраля 2017г.</i>	размещение на сайте конференции http://conf.ostis.net/index.php?title=OSTIS-2017 проекта программы конференции
<i>10 февраля 2017г.</i>	размещение на сайте конференции http://conf.ostis.net/index.php?title=OSTIS-2017 Сборника материалов и Программы конференции OSTIS-2017
<i>16 февраля 2017г.</i>	регистрация участников и открытие конференции OSTIS-2017
<i>16-18 февраля 2017г.</i>	работа конференции OSTIS-2017

23 февраля 2017г. публикация фоторепортажа и отчёта о проведённой конференции на сайте конференции: <http://conf.ostis.net/index.php?title=OSTIS-2017>

ФОРМИРОВАНИЕ ПРОГРАММЫ КОНФЕРЕНЦИИ

Программа конференции формируется Программным комитетом по результатам рецензирования, представленных статей, а также на основании подтверждения автора(-ов) статьи о прибытии на конференцию.

КОНТАКТНЫЕ ДАННЫЕ ОРГАНИЗАТОРОВ КОНФЕРЕНЦИИ OSTIS

Вся необходимая информация по предстоящей и предыдущих конференциях OSTIS находится на сайте конференции <http://conf.ostis.net>.

Материалы для участия в конференции представляются в Оргкомитет конференции по электронной почте ostisconf@gmail.com.

Методическая и консультативная помощь участникам конференции осуществляется только через электронную почту конференции.

Конференция проходит в Республике Беларусь, г. Минск, ул. Платонова, 39 (5-ый учебный корпус Белорусского государственного университета информатики и радиоэлектроники).

Библиотека БГУИР