

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

***ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ.
КУРСОВОЕ ПРОЕКТИРОВАНИЕ***

*Рекомендовано УМО по образованию в области
информатики и радиоэлектроники
для специальности 1-40 03 01 «Искусственный интеллект»
в качестве пособия*

Минск БГУИР 2017

УДК 004.8(075.9)

ББК 32.813я75

И86

Авторы:

В. В. Голенков, Н. А. Гулякина,
Н. В. Гракова, И. Т. Давыденко, Д. В. Шункевич

Рецензенты:

кафедра интеллектуальных систем
Белорусского государственного университета
(протокол №14 от 17.05.2016);

заведующий лабораторией идентификации систем государственного научного учреждения «Объединенный институт информатики Национальной академии наук Беларуси», доктор технических наук, профессор А. А. Дудкин

Искусственный интеллект. Курсовое проектирование : пособие /
И86 В. В. Голенков [и др.]. – Минск : БГУИР, 2017. – 62 с. : ил.
ISBN 978-985-543-333-1.

Сформулированы основные положения, касающиеся выполнения курсового проектирования на кафедре интеллектуальных информационных технологий, даны рекомендации по решению типовых задач в рамках курсового проекта (работы), а также рассмотрены типичные ошибки и проблемы, возникающие в процессе выполнения курсового проекта.

УДК 004.8(075.9)

ББК 32.813я75

ISBN 978-985-543-333-1

© УО «Белорусский государственный университет информатики и радиоэлектроники», 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 ЦЕЛИ, ЗАДАЧИ И ФОРМА ОРГАНИЗАЦИИ КУРСОВОГО ПРОЕКТИРОВАНИЯ НА КАФЕДРЕ ИИТ	6
2 ТЕМАТИКА, СОДЕРЖАНИЕ И ПРАВИЛА ОФОРМЛЕНИЯ РЕЗУЛЬТАТОВ СТУДЕНЧЕСКИХ ПРОЕКТОВ	9
3 ОБЩАЯ МЕТОДИКА ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ПО ТЕХНОЛОГИИ OSTIS.....	11
3.1 Выбор и установка sc-хранилища.....	11
3.2 Установка scr-машины	12
3.3 Установка ядра базы знаний дочерней sc-системы.....	12
3.4 Установка ядра машины обработки знаний дочерней sc-системы.....	13
3.5 Установка ядра пользовательского интерфейса дочерней sc-системы.....	14
3.6 Установка ядра подсистемы поддержки проектирования дочерней sc-системы.....	16
4 МЕТОДИКА ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ СИСТЕМ, ПОСТРОЕННЫХ ПО ТЕХНОЛОГИИ OSTIS.....	17
4.1 Построение системы предметных областей и их онтологий, входящих в состав базы знаний проектируемой OSTIS-системы.....	17
4.2 Разработка фрагментов используемых предметных областей	19
4.3 Разработка структурных спецификаций для каждой используемой предметной области	20
4.4 Разработка терминологических онтологий для каждой используемой предметной области	21
4.5 Разработка теоретико-множественных онтологий для каждой используемой предметной области	23
4.6 Разработка логических онтологий для каждой используемой предметной области.....	24
4.7 Разработка логической системы понятий и их определений для каждой используемой предметной области	25
4.8 Разработка логической системы утверждений для каждой используемой предметной области	26
4.9 Разработка онтологий задач и решений для каждой используемой предметной области	30
4.10 Разработка онтологий классов задач и способов решения задач для каждой используемой предметной области	32
5 МЕТОДИКА ПРОЕКТИРОВАНИЯ МАШИН ОБРАБОТКИ ЗНАНИЙ.....	34
5.1 Расширение машины обработки знаний с применением многократно используемых компонентов OSTIS	34
5.2 Расширение машины обработки знаний на основе собственных разработок.....	36
5.3 Этап задания условия геометрической задачи	37

5.4 Этап работы поисковых операций	37
5.5 Этап вывода по аналогии	38
5.6 Этап применения стратегий решения задач	39
5.7 Этап применения правил логического вывода	40
5.8 Этап оптимизации сгенерированных знаний и сборки мусора	41
6 МЕТОДИКА ПРОЕКТИРОВАНИЯ КОМПОНЕНТОВ ИНТЕРФЕЙСА	43
7 ОРГАНИЗАЦИЯ ВЫПОЛНЕНИЯ СТУДЕНЧЕСКИХ ПРОЕКТОВ	44
8 ЗАЩИТА СТУДЕНЧЕСКИХ ПРОЕКТОВ, ИНДИВИДУАЛЬНАЯ АТТЕСТАЦИЯ СТУДЕНТОВ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ	48
9 ПЛАНИРОВАНИЕ КУРСОВОГО ПРОЕКТИРОВАНИЯ НА СЛЕДУЮЩИЙ СЕМЕСТР	52
ПРИЛОЖЕНИЕ А СТРУКТУРА БАЗЫ ЗНАНИЙ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ	53
ПРИЛОЖЕНИЕ Б СПИСОК ВОЗМОЖНЫХ ВАРИАНТОВ ТЕМ КУРСОВОГО ПРОЕКТА	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	60

Библиотека БГУИР

ВВЕДЕНИЕ

Данное пособие полностью соответствует Положению об организации и проведении курсового проектирования в БГУИР и уточняет специфику организации курсового проектирования на кафедре ИИТ.

Полный перечень документов, регламентирующих организацию курсового проектирования на кафедре ИИТ, включает в себя:

- положение об организации курсового проектирования в БГУИР;
- учебный план специальности «Искусственный интеллект» пятилетнего цикла;
- учебный план специальности «Искусственный интеллект» четырехлетнего цикла;
- правила оформления документации разрабатываемой системы (спецификация, обоснование, описание текущего состояния);
- правила оформления индивидуальных отчетов по курсовому проектированию (пояснительных записок);
- список студенческих проектов, выполняемых в текущий момент (с указанием команд разработчиков);
- список приостановленных студенческих проектов (с их участниками);
- список предлагаемой тематики студенческих проектов;
- общие рекомендации начинающим разработчикам сложных компьютерных систем.

1 ЦЕЛИ, ЗАДАЧИ И ФОРМА ОРГАНИЗАЦИИ КУРСОВОГО ПРОЕКТИРОВАНИЯ НА КАФЕДРЕ ИИТ

Основная цель курсового проектирования на кафедре ИИТ – максимально приблизить курсовое проектирование к реальной профессиональной деятельности по промышленной разработке, внедрению и сопровождению интеллектуальных систем различного назначения. Только так можно сформировать профессиональные и человеческие качества, необходимые для работы в коллективных сложных наукоемких проектах (мотивированность, целеустремленность, результативность, активность, самостоятельность, ответственность, умение участвовать в выработке согласованной позиции всей команды, проектная дисциплина, адекватная самооценка, коммуникативность и многое другое).

Условно можно выделить три уровня квалификации разработчиков сложных наукоемких (в том числе и интеллектуальных) информационных систем.

– Разработчик способен индивидуально выполнять четко поставленные перед ним проектные задачи.

– Разработчик имеет навыки формирования команды, перед которой ставится только конечная цель (планирование работ, контроль исполнения, интеграция, коллективная ответственность). Переход с первого на второй уровень существенно сокращает сроки и повышает качество разрабатываемой системы, так как каждый лично заинтересован в обеспечении целостности и качества разработки.

– Разработчик имеет навыки доработки технологии, если в ней не хватает необходимых компонентов, для реализации разрабатываемого приложения. Переход на третий уровень дает возможность существенно повысить уровень конкурентоспособности информационных систем (поскольку они разрабатываются на основе усовершенствованных технологий) и сократить сроки новых аналогичных разработок за счет накопленного в рамках развивающейся технологии собственного проектного опыта.

Цель курсового проектирования на кафедре ИИТ – развить все три уровня квалификации за счет формирования у студентов и магистрантов профессионального опыта работы на всех указанных выше уровнях.

Организационной формой курсового проектирования (то есть выполнение курсовых проектов, курсовых работ и расчетных работ) являются коллективно или индивидуально выполняемые **студенческие проекты**, каждый из которых имеет достаточно длительный срок разработки (как минимум

несколько лет). Они направлены на разработку интеллектуальных систем различного назначения либо на разработку компонентов таких систем и являются основной формой приобретения профессиональных навыков по специальности «Искусственный интеллект». Их основной целью должен быть переход от стартового состояния проекта к приобретению финансовых ресурсов для дальнейшего развития этого проекта.

Студенческие проекты выполняются студентами и магистрантами, которые несут полную ответственность за результативность и качество своей деятельности, уточнение постановки задачи, планирование работы, сборку и тестирование разрабатываемой системы. Студенческие проекты также могут иметь научного руководителя, технического консультанта, эксперта по предметной области.

Право участия в коллективных студенческих проектах предоставляется только успевающим и ответственным студентам.

Жизненный цикл каждого студенческого проекта может охватывать весь период обучения, включая дипломный проект и магистерскую диссертацию.

Студенческие проекты условно делятся на две категории:

- проекты прикладных интеллектуальных систем по различным предметным областям;
- проекты интеллектуальных систем, входящих в состав *Метасистемы IMS* или являющихся ее подсистемами либо различными компонентами библиотек.

Студенческие проекты преследуют три основные цели:

- практическая реализация некоторой конкретной интеллектуальной системы на базе *Технологии OSTIS*;
- организация внедрения (использования) разработанной интеллектуальной системы;
- внесение вклада в развитие *Технологии OSTIS* (точнее *Метасистемы IMS*) в тех направлениях, которые были необходимы для реализации разрабатываемой интеллектуальной системы.

Конечной целью каждого студенческого проекта является:

- промышленная разработка и внедрение некоторой интеллектуальной системы;
- включение накапливаемого проектного опыта в состав постоянно развивающейся технологии проектирования интеллектуальных систем, осуществляющей поддержку проектирования различных интеллектуальных систем.

В результате проектной деятельности должны сформироваться навыки коллективной, результативной, самостоятельной, ответственной и профессиональной деятельности, сочетающей творческую инициативу и проектную дисциплину.

Библиотека БГУИР

2 ТЕМАТИКА, СОДЕРЖАНИЕ И ПРАВИЛА ОФОРМЛЕНИЯ РЕЗУЛЬТАТОВ СТУДЕНЧЕСКИХ ПРОЕКТОВ

Связь студенческих проектов с учебными дисциплинами по четырехлетнему плану обучения осуществляется следующим образом:

– В индивидуальный план и, соответственно, индивидуальный отчет студентов 2-го курса в осеннем семестре должно быть включено задание, касающееся решения **индивидуальной теоретико-графовой задачи** в семантической памяти и по крайней мере одно индивидуальное задание, касающееся **разработки базы знаний интеллектуальной системы**, проектируемой в рамках соответствующего студенческого проекта. Это связывает курсовое проектирование с такими дисциплинами, как «Проектирование программ в интеллектуальных системах», «Математические основы интеллектуальных систем», «Традиционные интеллектуальные информационные технологии», «Основы дискретной математики».

– В индивидуальный план и, соответственно, индивидуальный отчет студентов 2-го курса в весеннем семестре должно быть включено по крайней мере одно индивидуальное задание, касающееся **разработки информационно-поисковых операций интеллектуальной системы**, проектируемой в рамках соответствующего студенческого проекта. Это связывает курсовое проектирование с такими дисциплинами, как «Математические основы интеллектуальных систем», «Общая теория систем», «Модели решения задач в интеллектуальных системах».

– В индивидуальный план и, соответственно, индивидуальный отчет студентов 3-го курса в осеннем семестре должно быть включено по крайней мере одно индивидуальное задание, касающееся **разработки машины обработки знаний интеллектуальной системы**, проектируемой в рамках соответствующего студенческого проекта. Это связывает курсовое проектирование с такими дисциплинами, как «Модели решения задач в интеллектуальных системах», «Проектирование баз знаний».

– В индивидуальный план и, соответственно, индивидуальный отчет студентов 3-го курса в весеннем семестре должно быть включено по крайней мере одно индивидуальное задание, касающееся **разработки пользовательского интерфейса интеллектуальной системы**, проектируемой в рамках соответствующего студенческого проекта. Это связывает курсовое проектирование с такими дисциплинами, как «Логические основы

интеллектуальных систем», «Языковые процессоры интеллектуальных систем», «Естественно-языковой интерфейс интеллектуальных систем».

– В индивидуальный план и, соответственно, индивидуальный отчет студентов 4-го курса в осеннем семестре должно быть включено по крайней мере одно индивидуальное задание, касающееся **разработки многократно используемых компонентов интеллектуальной системы**, проектируемой в рамках соответствующего студенческого проекта и включения этих компонентов в соответствующие библиотеки. Это связывает курсовое проектирование с такими дисциплинами, как «Технология проектирования интеллектуальных систем», «Естественно-языковой интерфейс интеллектуальных систем», «Графический интерфейс интеллектуальных систем».

Темы коллективных студенческих проектов могут предлагаться как кафедрой, так и самими студентами. Темы, предлагаемые кафедрой, отражены в списке тем.

Результаты исследований должны быть оформлены:

– в виде индивидуального отчета по курсовому проектированию (пояснительная записка) в соответствии с СТП-2013 (стандарт предприятия). Из индивидуального отчета должно быть понятно, что было сделано за отчетный период;

– в виде документации разрабатываемой системы (разрабатываемого проекта), пригодной для размещения в соответствующем разделе базы знаний самой системы (приложение А).

Практический результат в виде интеллектуальной системы соответствующей предметной области должен быть размещен на специализированном и доступном для экспертов интернет-ресурсе.

3 ОБЩАЯ МЕТОДИКА ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ПО ТЕХНОЛОГИИ OSTIS

При создании стартовой версии *дочерней sc-системы* по *Технологии OSTIS* можно выделить четыре основных этапа:

- выбор и установка платформы реализации *дочерней sc-системы*;
- установка *Ядра sc-моделей баз знаний*, то есть набора базовых *многократно используемых компонентов sc-моделей баз знаний*, необходимых для работы даже первого прототипа *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 *дочерней sc-системы*.

После того, как собрана самая базовая конфигурация *дочерней sc-системы*, можно начинать постепенное ее наполнение либо с использованием *Библиотеки многократно используемых компонентов Технологии OSTIS*, либо путем создания своих собственных компонентов, которые, в свою очередь, могут в дальнейшем быть включены в состав указанной библиотеки.

3.1 Выбор и установка *sc-хранилища*

В качестве платформы для разработки справочной системы по геометрии в данной работе выбрана *Web-ориентированная платформа реализации sc-систем*, основанная на специальном формате кодирования *sc-текстов*, входящая в состав *Библиотеки платформ реализации sc-систем* [9]. Согласно документации по данному *платформенно-зависимому многократно используемому компоненту OSTIS* [10] данный вариант реализации платформы ориентирован на операционные системы семейства Linux и сопровождается скриптами, позволяющими собрать и установить выбранную платформу на нужном сервере.

Данный вариант реализации платформы включает в себя собственно *sc-хранилище*, а также набор трансляторов.

3.2 Установка scr-машины

После того, как на сервер была установлена выбранная реализация sc-памяти, разработчик может установить также совместимую с выбранной реализацией платформы реализацию абстрактной scr-машины. На семантическом уровне совместимость задается при помощи отношения *совместимый компонент**. Таким образом, для выбранной реализации sc-памяти подойдет реализация абстрактной scr-машины на базе *web-ориентированной платформы реализации sc-систем*. Исходные коды и инструкцию по установке данного платформенно-зависимого многократно используемого компонента OSTIS согласно его описанию в библиотеке платформ реализации sc-моделей можно найти в источнике [10]. В данном варианте компонента для реализации sc-агентов используются языки C и C++.

Для корректной работы какого-либо варианта реализации абстрактной scr-машины необходимо наличие в системе соответствующего многократно используемого компонента sc-моделей баз знаний, описывающего *Предметную область программ базового языка программирования*, ориентированного на обработку sc-моделей баз знаний.

После того, как были установлены необходимые части платформы реализации sc-моделей, можно приступать к установке необходимых платформенно-независимых многократно используемых компонентов OSTIS.

3.3 Установка ядра базы знаний дочерней sc-системы

Ядро sc-моделей баз знаний рассматривается как один *неатомарный многократно используемый компонент OSTIS* и содержит в себе все фрагменты базы знаний, необходимые для корректной работы *платформы реализации sc-моделей интеллектуальных систем*, а также базовых *sc-агентов*, использование которых целесообразно в любой *дочерней sc-системе*.

Для понятности кратко перечислим фрагменты базы знаний, входящие в *Ядро sc-моделей баз знаний*, используя язык SCn [8]:

- *Раздел. Предметная область семантических окрестностей.*
- *Раздел. Предметная область ситуативно-событийных моделей изменения состояний баз знаний.*
- *Раздел. Предметная область моделей действий, направленных на изменение состояния баз знаний.*
- *Раздел. Предметная область чисел.*
- *Раздел. Предметная область сообщений, принимаемых и передаваемых интеллектуальной системой.*

- *Раздел. SC-код.*
 - *Раздел. Предметная область sc-элементов.*
 - *Раздел. Предметная область sc-множеств.*
 - *Раздел. Предметная область sc-отношений.*
 - *Раздел. Предметная область библиографических источников.*
 - *Раздел. Предметная область sc-структур.*
 - *Раздел. Предметная область sc-ссылок.*
 - *Раздел. Предметная область внешних sc-идентификаторов.*
 - *Раздел. Предметная область физических лиц и коллективов.*
- *Раздел. SCs-код.*
- *Раздел. SCn-код.*
- *Раздел. SCg-код.*
- *Базовый набор команд информационного поиска.*
 - *Команды запроса структуры:*
 - *запрос всех надклассов;*
 - *запрос декомпозиции;*
 - *запрос всех подклассов.*
 - *Команды базовых запросов:*
 - *запрос позитивных, константных выходящих дуг с отношениями;*
 - *запрос позитивных, константных входящих дуг с отношениями;*
 - *запрос позитивных, константных выходящих дуг;*
 - *запрос позитивных, константных входящих дуг.*
 - *Команды запроса семантических окрестностей:*
 - *запрос полной семантической окрестности.*
 - *Команды запроса идентификаторов:*
 - *запрос всех идентификаторов.*

3.4 Установка ядра машины обработки знаний дочерней sc-системы

Ядро машины обработки знаний *sc-систем* (*sc-машин*) включает в себя минимальный набор предметно независимых *sc-агентов информационного поиска*, необходимый для навигации по базе знаний дочерней *sc-системы*.

В текущем варианте указанное ядро включает в себя:

– *sc-агент поиска всех выходящих константных позитивных стационарных sc-дуг принадлежности;*

- *sc-агент поиска всех выходящих константных позитивных стационарных sc-дуг принадлежности с их ролевыми отношениями;*
- *sc-агент поиска всех входящих константных позитивных стационарных sc-дуг принадлежности;*
- *sc-агент поиска всех выходящих константных позитивных стационарных sc-дуг принадлежности с их ролевыми отношениями;*
- *sc-агент поиска связок декомпозиции для заданного sc-элемента;*
- *sc-агент поиска всех известных сущностей, являющихся частными по отношению к заданной;*
- *sc-агент поиска всех известных сущностей, являющихся общими по отношению к заданной;*
- *sc-агент поиска всех идентификаторов заданного sc-элемента;*
- *sc-агент поиска полной семантической окрестности заданного элемента;*
- *sc-агент поиска связок заданного отношения, компонентом которых является заданный sc-элемент;*
- *sc-агент поиска всех конструкций, изоморфных заданному образцу.*

Данный набор *sc-агентов* может быть реализован как на платформенно-независимом уровне, так и на уровне платформы. В нашем случае реализация всех перечисленных *sc-агентов* входит в состав выбранной платформы реализации *sc-моделей*, что, с одной стороны, упрощает их установку в дочернюю *sc-систему*, но с другой сильно осложняет процесс внесения изменений в состав ядра.

Чтобы обеспечить возможность задания более сложных поисковых запросов в систему необходимо добавить дополнительные *sc-агенты* из библиотеки многократно используемых *sc-агентов OSTIS*.

Для корректной работы данных *sc-агентов* дополнительно необходимо наличие в базе знаний определенного набора *сопутствующих компонентов**.

3.5 Установка ядра пользовательского интерфейса дочерней sc-системы

Ядро sc-моделей пользовательских интерфейсов, как и собственно платформа реализации *sc-моделей*, представляет собой платформенно-зависимый многократно используемый компонент *OSTIS*, который должен быть совместим с выбранным вариантом платформы. Подходящий вариант реализации ядра пользовательских интерфейсов, описанный в Библиотеке многократно используемых компонентов *sc-моделей интерфейсов* может быть

найден в источнике [7]. Согласно описанию, выбранный компонент содержит в себе помимо собственно ядра следующий набор компонентов:

- визуализатор *sc-конструкций* в *SCg-коде*;
- редактор *sc-конструкций* в *SCg-коде*;
- визуализатор *sc-конструкций* в *SCn-коде*;
- набор компонентов для отображения *sc-ссылок* различных форматов (PNG, GIF, PDF, HTML, TXT и др.).

Также в выбранный ранее вариант *платформы реализации sc-моделей* по умолчанию входят трансляторы из *sc-памяти* в формат, необходимый для SCg- и SCn-визуализации, а также транслятор из SCg-редактора в *sc-память*. Таким образом, установленных средств достаточно для базовой визуализации и редактирования *sc-конструкций* посредством web-интерфейса.

Примеры отображения некоторых произвольных конструкций посредством универсальных компонентов интерфейса приведены на рисунках 3.1–3.3.

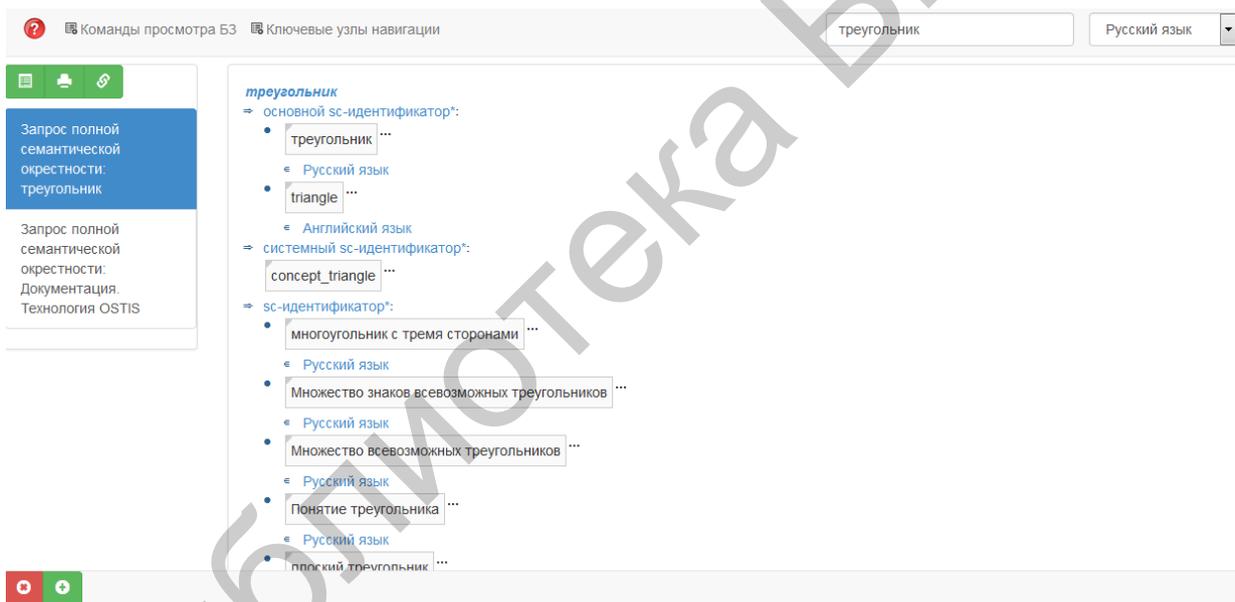


Рисунок 3.1 – Отображение фрагмента базы знаний в SCn-коде

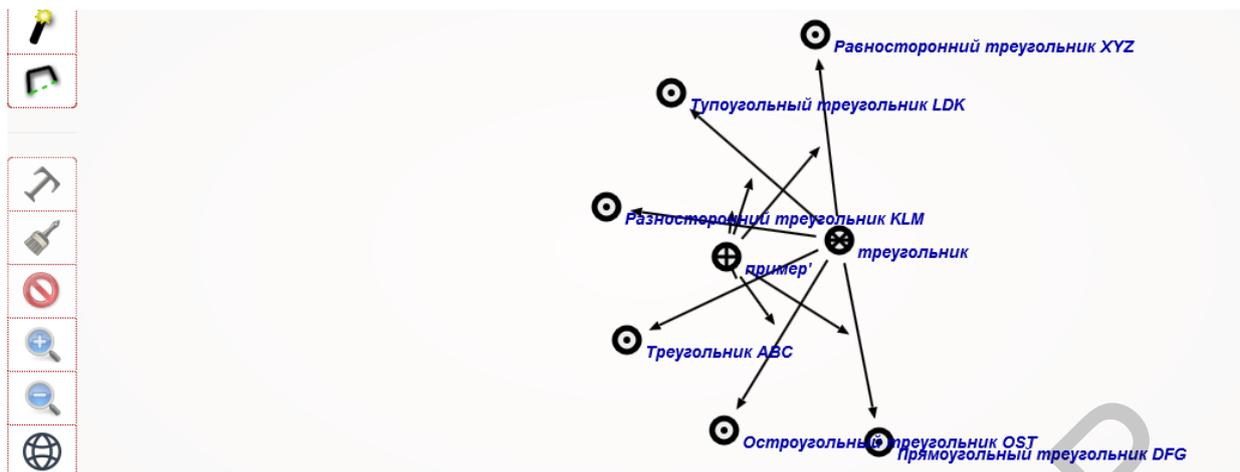


Рисунок 3.2 – Отображение фрагмента базы знаний в SCg-коде

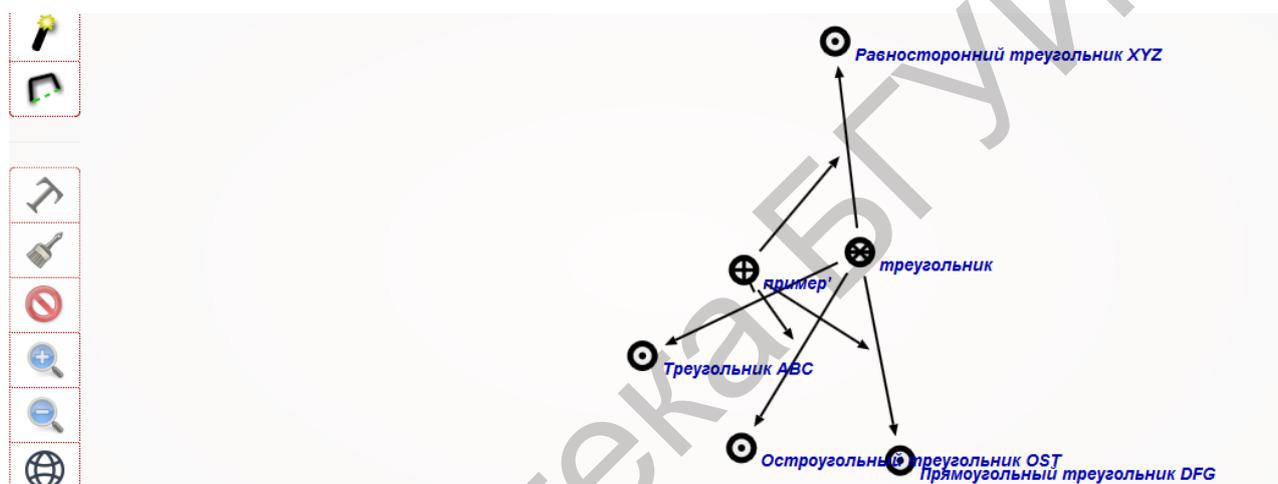


Рисунок 3.3 – Результат редактирования фрагмента базы знаний в SCg-коде

3.6 Установка ядра подсистемы поддержки проектирования дочерней sc-системы

Подсистема поддержки проектирования *дочерней sc-системы* в свою очередь состоит из двух взаимосвязанных подсистем, одна из которых входит в состав самой *IMS*, а другая является *многократно используемой типовой подсистемой интеллектуальных систем* и входит в состав минимального набора компонентов, необходимых для полноценного функционирования *дочерней sc-системы* в рамках *Глобальной базы знаний* [1].

Ядро данной типовой подсистемы включает в себя необходимые средства интеграции *многократно используемых компонентов OSTIS* в дочернюю *sc-систему* и средства управления проектированием *дочерней sc-системы*.

4 МЕТОДИКА ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ СИСТЕМ, ПОСТРОЕННЫХ ПО ТЕХНОЛОГИИ OSTIS

Помимо базовых многократно используемых компонентов баз знаний, входящих в состав ядра базы знаний, каждая дочерняя система может быть дополнена другими компонентами из *Библиотеки многократно используемых компонентов sc-моделей баз знаний*. Общие принципы проектирования баз знаний по *Технологии OSTIS* рассмотрены в [3]. Рассмотрим общую методику проектирования базы знаний на примере базы знаний справочной системы по геометрии.

4.1 Построение системы предметных областей и их онтологий, входящих в состав базы знаний проектируемой OSTIS-системы

Первым и важнейшим этапом проектирования базы знаний является уточнение структуры описываемой предметной области или нескольких взаимосвязанных предметных областей. Уточнение такой структуры – это, прежде всего, уточнение класса исследуемых объектов, предмета исследования, всего семейства ключевых узлов семантической сети, представляющей предметную область. В рамках предметной области возможно выделение *частных предметных областей* на основе выделения подмножества из семейства классов исследуемых объектов или изменения набора исследуемых отношений.

При описании структуры предметной области используются ключевые узлы, входящие в состав *Предметной области предметных областей*, входящей в *Библиотеку многократно используемых компонентов баз знаний* в виде компонента, следовательно, данный компонент необходимо добавить в состав базы знаний разрабатываемой системы.

Пример структуры *Предметной области геометрии Евклида*:

Предметная область геометрии Евклида

∈ предметная область

=> частная предметная область*:

- Предметная область геометрических точек
- Предметная область линий
- Предметная область планарных фигур

=> частная предметная область*:

- Предметная область прямолинейных фигур
- Предметная область планарных углов
- Предметная область многоугольников

=> частная предметная область*:

- Предметная область треугольников
 - Предметная область четырехугольников
 - Предметная область вписанных планарных фигур
 - Предметная область кругов и окружностей
 - Предметная область геометрических поверхностей
 - Предметная область геометрических тел
- => частная предметная область*:
- Предметная область многогранников и их поверхностей
 - Предметная область непланарных углов
 - Предметная область тел вращения и их поверхностей
 - Предметная область конгруэнтности геометрических фигур

Рассмотрение базы знаний с позиции ее соотношения с предметной областью позволяет рассматривать исследуемые объекты на различных уровнях детализации, которые отражаются в различных типах онтологий, описывающих определенное направление описания свойств объекта в рамках рассматриваемой предметной области. К типам таких онтологий относятся:

– *структурная спецификация предметной области* – описание связей рассматриваемой предметной области с другими предметными областями и ролей всех понятий, входящих в состав данной предметной области;

– *терминологическая онтология* – описание терминов и их синонимов, ключевых понятий рассматриваемой предметной области, близких между собой терминов, этимологии терминов и правила построения идентификаторов экземпляров понятий;

– *теоретико-множественная онтология* – описание теоретико-множественных связей между понятиями рассматриваемой предметной области;

– *логическая онтология* – описание всех высказываний рассматриваемой предметной области;

– *логическая система понятий и их определений* – это структура, 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;

– *логическая система утверждений и их доказательств* – это структура, являющаяся надстройкой над логической онтологией, включающая описание системы утверждений рассматриваемой предметной области с указанием набора утверждений, через которые доказывается каждое утверждение;

– *онтология задач и решений задач* – описание конкретных задач, рассматриваемых в заданной предметной области, и их решений;

– *онтология классов задач и способов решения задач* – описание классов задач, рассматриваемых в заданной предметной области, и способов их решения. Данная структура является надстройкой над *онтологией задач и решений задач*.

Для описания указанной структуры предметной области используются ключевые понятия, входящие в состав *Предметной области онтологий – многократно используемого компонента sc-моделей баз знаний*.

Таким образом, для всех разделов базы знаний, описывающих предметную область и ее спецификацию, имеет смысл задать типовую структуру. В качестве примера рассмотрим *Предметную область четырехугольников* и ее онтологии.

Раздел. Предметная область четырехугольников

∈ предметная область и ее онтологии

<= декомпозиция раздела:*

{

- *Предметная область четырехугольников*
- *Структурная спецификация предметной области четырехугольников*
- *Терминологическая онтология предметной области четырехугольников*
- *Теоретико-множественная онтология предметной области четырехугольников*
- *Логическая онтология предметной области четырехугольников*
- *Логическая иерархия понятий предметной области четырехугольников*
- *Логическая иерархия высказываний о предметной области четырехугольников*
- *Онтология задач и решений задач предметной области четырехугольников*
- *Онтология классов задач и способов решения задач предметной области четырехугольников*

}

Далее рассмотрим подробнее фрагменты указанных разделов спецификации предметной области.

4.2 Разработка фрагментов используемых предметных областей

Любая предметная область также может содержать примеры конкретных объектов исследования, то есть экземпляры классов исследования.

На рисунке 4.1 приведен пример отображения в системе семантической окрестности объекта исследования Параллелограмм ABCD.

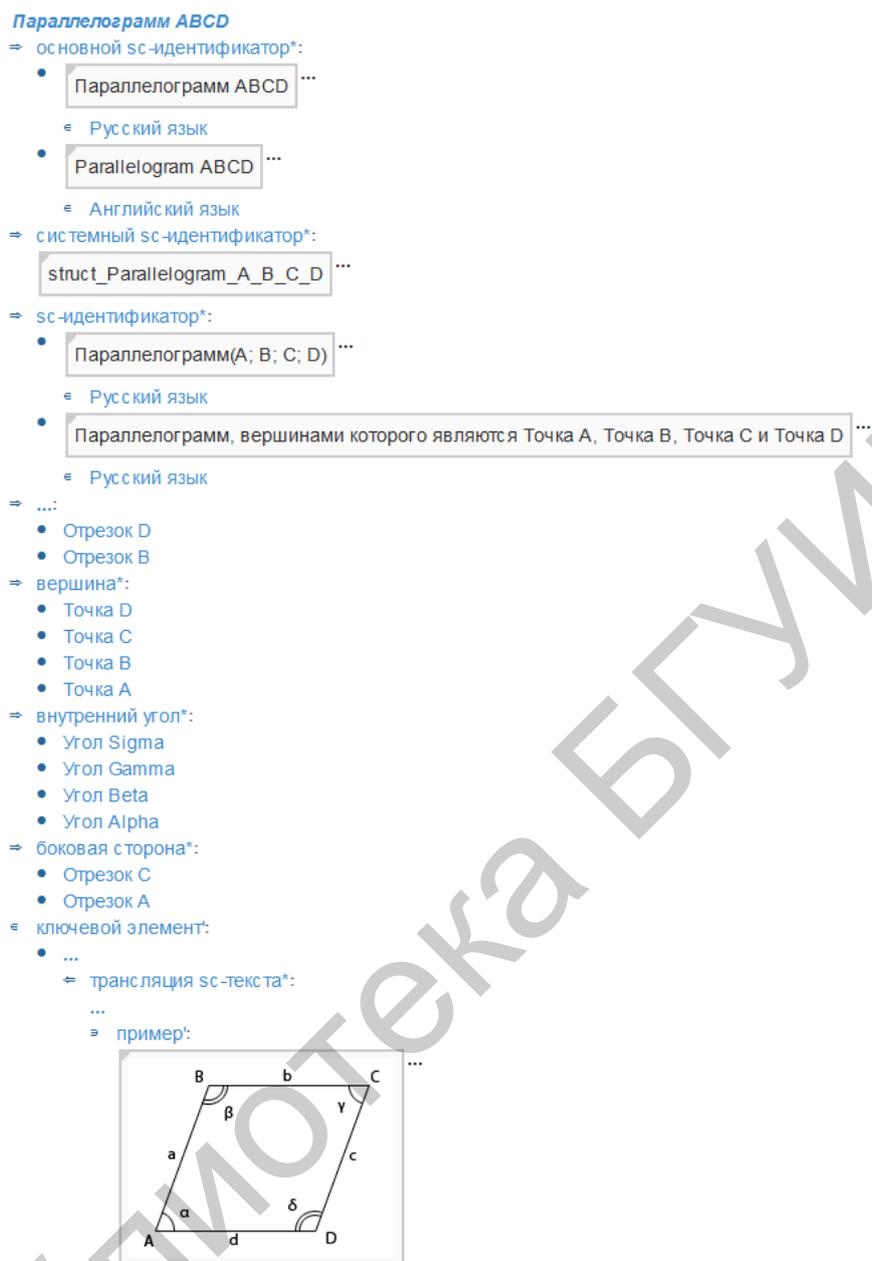


Рисунок 4.1 – Семантическая окрестность объекта исследования
Параллелограмм ABCD

4.3 Разработка структурных спецификаций для каждой используемой предметной области

Рассмотрим фрагмент *Структурной спецификации предметной области четырехугольников*:

Предметная область четырехугольников

=> онтология*:

Структурная спецификация предметной области четырехугольников

= [

Предметная область четырехугольников

<= частная предметная область*:

Предметная область многоугольников

⇒ *максимальный класс объектов исследования*':

четыреугольник

⇒ *немаксимальный класс объектов исследования*':

- *параллелограмм*
- *прямоугольник*
- *ромб*
- *квадрат*
- *трапеция*
- *равнобедренная трапеция*
- *прямоугольная трапеция*

⇒ *элемент структуры*':

- *квадрат ABCD*
- *четыреугольник KLMN*
- *трапеция BMNO*

⇒ *исследуемое отношение*':

- *внутренний угол**
- *сторона**
- *высота**
- *площадь**
- *периметр**
- *средняя линия**

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы OSTIS-системе:

- Что является объектом исследования данной предметной области?
- Какие отношения исследуются в данной предметной области?

4.4 Разработка терминологических онтологий для каждой используемой предметной области

Рассмотрим фрагмент *Терминологической онтологии предметной области четырехугольников*:

Предметная область четырехугольников

=> *онтология**:

Терминологическая онтология предметной области четырехугольников

=[

четыреугольник

= *многоугольник с четырьмя сторонами*

= *многоугольник с четырьмя углами*

<= правила идентификации элементов*:

- {
- [Четырехугольник < суффикс >]
- [Чет-к < суффикс >]
- [Четырехугольник (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>, <идентификатор точки>)]
- }

прямоугольник

= четырехугольник с прямыми внутренними углами

<= правила идентификации элементов*:

- {
- [Прямоугольник < суффикс >]
- [Прям-к < суффикс >]
- [Прямоугольник (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>, <идентификатор точки>)]
- }

ромб

= четырехугольник с равными сторонами

= четырехугольник с конгруэнтными сторонами

<= правила идентификации элементов*:

- {
- [Ромб < суффикс >]
- [Ромб (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>, <идентификатор точки>)]
- }

квадрат

= ромб с прямыми углами

= прямоугольник с равными сторонами

= прямоугольник \cap ромб

<= правила идентификации элементов*:

- {
- [Квадрат < суффикс >]
- [Квадрат (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>, <идентификатор точки>)]
- }

трапеция

<= правила идентификации элементов*:

- {
- [Трапеция < суффикс >]
- [Трапеция (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>, <идентификатор точки>)]
- }

равнобедренная трапеция

= трапеция с равными сторонами

прямоугольная трапеция

= трапеция с прямыми углами при боковой стороне

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *OSTIS-системе*:

- Какие синонимичные термины известны для заданного термина?
- По какому правилу идентифицируются элементы заданного множества?

4.5 Разработка теоретико-множественных онтологий для каждой используемой предметной области

Рассмотрим фрагмент *Теоретико-множественной онтологии предметной области четырехугольников*:

Предметная область четырехугольников

=> онтология*:

Теоретико-множественная онтология предметной области четырехугольников

=[

четыреугольник

▷ трапеция

трапеция

▷ параллелограмм

▷ равнобедренная трапеция

▷ прямоугольная трапеция

параллелограмм

▷ прямоугольник

▷ ромб

квадрат

⊂ прямоугольник

⊂ ромб

средняя линия*

∈ бинарное отношение

=> область определения*:

трапеция ∪ отрезок

=> первый домен*:

трапеция

=> второй домен*:

отрезок

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *OSTIS-системе*:

- Как классифицируется заданное понятие?
- Какие надклассы известны для заданного понятия?
- Являются ли два заданных понятия пересекающимися?
- Какова область определения заданного отношения?

4.6 Разработка логических онтологий для каждой используемой предметной области

Рассмотрим фрагмент *Логической онтологии предметной области четырехугольников*

Предметная область четырехугольников

=> онтология*:

Логическая онтология предметной области четырехугольников

=[

четыреугольник

∈ *ключевой sc-элемент*':

Опр.(четыреугольник)

∈ *определение*

<= *трансляция sc-текста**:

...

Э *пример*':

[**четыреугольник** – это многоугольник с четырьмя сторонами]

∈ *Русский язык*

...

∈ *утверждение*

<= *трансляция sc-текста**:

...

Э *пример*':

[Сумма мер внутренних углов четырехугольника равна 360 угловых градусов.]

∈ *Русский язык*

...

∈ *утверждение*

<= *трансляция sc-текста**:

...

Э *пример*':

[Сумма квадратов длин диагоналей четырехугольника равна удвоенной сумме длин квадратов отрезков, соединяющих середины его противоположных сторон.]

∈ *Русский язык*

...

∈ *утверждение*

<= трансляция sc-текста*:

...

Э пример':

[Если около четырехугольника можно описать окружность, то сумма мер его любых противоположащих углов равна 180 угловых градусов.]

∈ Русский язык

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *OSTIS-системе*:

- Какие высказывания известны в рамках заданной предметной области?
- Какие высказывания являются аксиомами в рамках заданной предметной области?
- Как определяется/поясняется то или иное понятие?

4.7 Разработка логической системы понятий и их определений для каждой используемой предметной области

Рассмотрим фрагмент *Логической системы понятий и их определений предметной области четырехугольников*

Логическая онтология предметной области четырехугольников

=> метазнание*:

Логическая система понятий и их определений предметной области четырехугольников

=[

четыреугольник

∈ ключевой sc-элемент':

...

∈ определение

<= трансляция sc-текста*:

...

Э пример':

[**четыреугольник** – это многоугольник с четырьмя сторонами]

∈ Русский язык

<= используемые константы*:

{

- *многоугольник*
- *сторона**

}

трапеция

∈ ключевой sc-элемент':

...

∈ определение

<= трансляция sc-текста*:

...

∃ пример':

[**трапеция** – это четырехугольник, у которого две противоположные стороны параллельны]

∈ Русский язык

<= используемые константы*:

{

- *четырёхугольник*
- *сторона**
- *параллельность**
- *противолежащий**

}

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *OSTIS-системе*:

- Через какие понятия определяется заданное понятие?
- Какие понятия определяются на основе заданного понятия?

4.8 Разработка логической системы утверждений для каждой используемой предметной области

В разрабатываемой системе по геометрии уместно привести типовые доказательства для некоторых теорем и типовые решения для некоторых примеров задач.

Для обеспечения такой возможности необходимо добавить в систему описание *Предметной области действий и задач*, содержащей все необходимые ключевые узлы, то есть *Раздел. Предметная область действий и задач*.

Рассмотрим фрагмент *Логической системы утверждений и их доказательств предметной области четырехугольников*.

Логическая онтология предметной области четырехугольников

=> *метазнание**:

Логическая система высказываний и их доказательств предметной области четырехугольников

=[

параллелограмм

∈ *ключевой sc-элемент'*:

Утв.(параллелограмм; диагональ; четырёхугольник)*

∈ *утверждение*


```

...
= результат*:
...
= трансляция sc-текста*:
...
  3 Прямая AD параллельна прямой BC. ...
    € Русский язык
= последовательность действий*:
...
= последовательность действий*:
...
  3 rrel_2:
    stat_parallelism_congruence_straight_line_internal_cross_lying_angles
  3 rrel_1:
    ...
    € ...
    = декомпозиция sc-действия*:
    ...
    € successfully_completed_action
    € sc-действие применения логического утверждения

```

Рисунок 4.3 – Пример отображения sc-текста шага доказательства

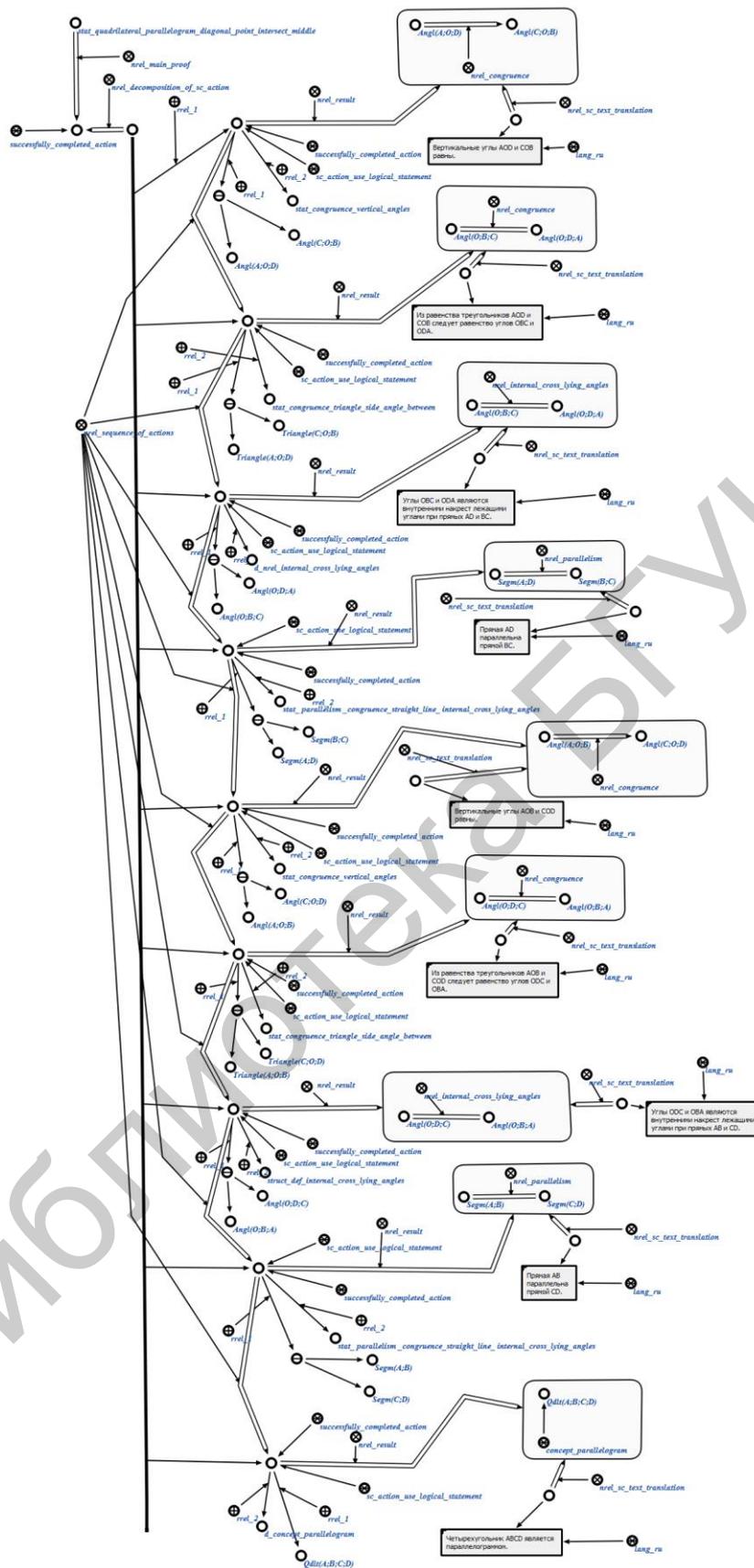


Рисунок 4.4 – Пример sc-текста доказательства, представленного в SCg-коде

4.9 Разработка онтологий задач и решений для каждой используемой предметной области

Аналогичным образом можно внести в систему образец решения, например, следующей задачи: «В параллелограмме ABCD $\sin(C) = 5/7$. AD = 7. Найдите высоту, опущенную на сторону AB».

Ниже приведен фрагмент *Онтологии задач и решений задач предметной области четырехугольников*.

Предметная область четырехугольников

=> онтология*:

Онтология задач и решений задач предметной области четырехугольников

=[

параллелограмм

∈ *ключевой sc-элемент*':

Задача. Нахождение высоты параллелограмма по длине стороны и синусу внутреннего угла

∈ *задача*

<= *трансляция sc-текста**:

...

∃ *пример*':

[В параллелограмме ABCD $\sin(C) = 5/7$. AD = 7. Найдите высоту, опущенную на сторону AB.]

∈ *Русский язык*

=> *решение**:

Решение. Задача. Нахождение высоты параллелограмма по длине стороны и синусу внутреннего угла

<= *используемые утверждения**:

{

- *Утв.(параллелограмм; противоположащий*; внутренний угол*; конгруэнтность*)*
- *Опр.(синус*)*

}

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *OSTIS-системе*:

- Примеры каких задач известны в рамках заданной предметной области?
- Как решается заданная задача?
- С помощью каких утверждений решается заданная задача?

Формальная запись условия этой задачи на языке SCg представлена на рисунке 4.5.

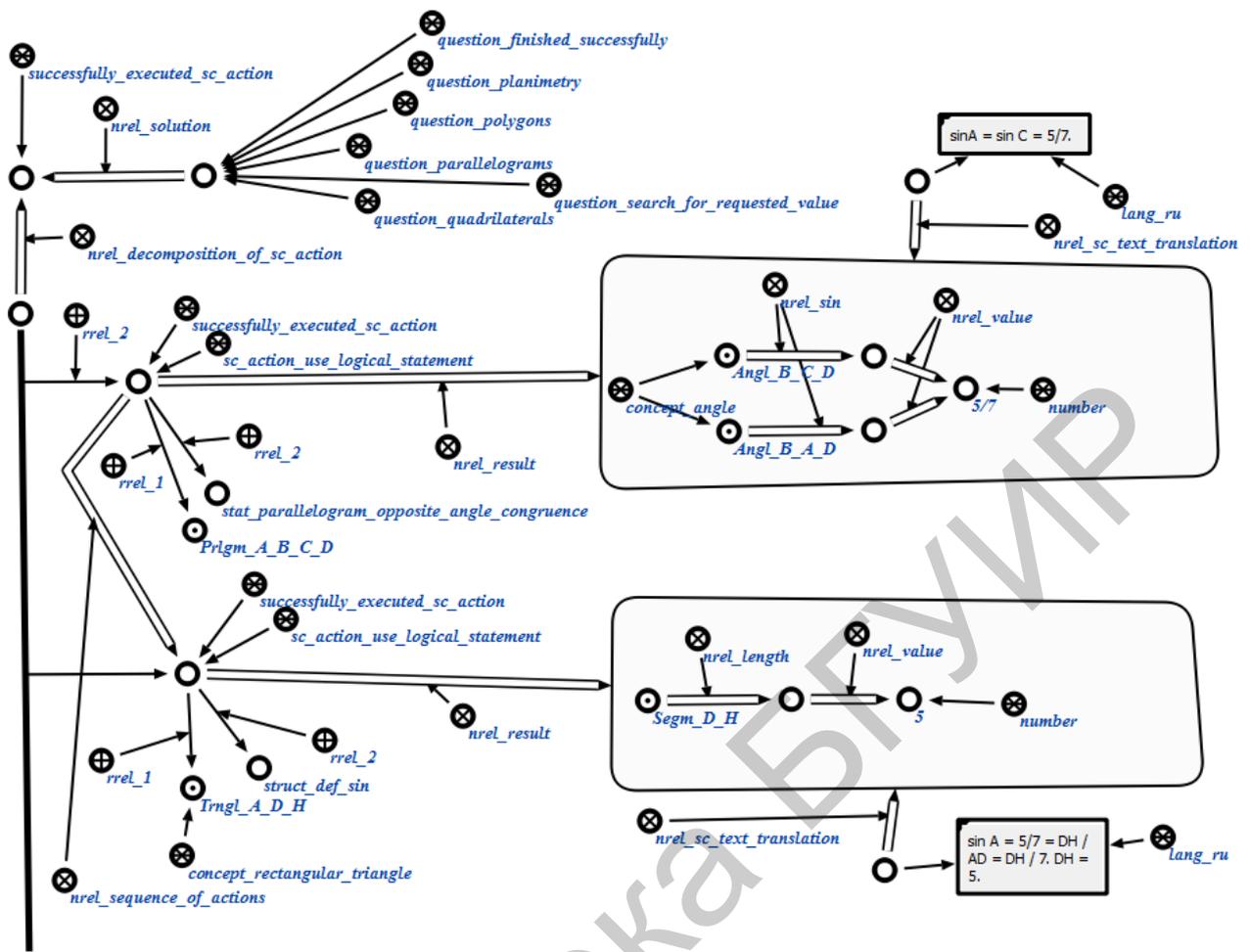


Рисунок 4.6 – Решение задачи, представленное в SCg-коде

В системе решение можно просмотреть по шагам при помощи базовых *sc-агентов информационного поиска*, однако, как и в случае с доказательствами, это не всегда удобно, поэтому имеет смысл добавить в систему *sc-агент поиска решения задачи*, позволяющий просмотреть все решение сразу.

4.10 Разработка онтологий классов задач и способов решения задач для каждой используемой предметной области

Рассмотрим фрагмент *Онтологии задач и решений задач предметной области четырехугольников*.

Онтология задач и решений задач предметной области четырехугольников

=> *метазнание**:

Онтология классов задач и способов решения задач предметной области четырехугольников

= [

Задача. Нахождение высоты параллелограмма по длине стороны и синусу внутреннего угла

∈ задача поиска заданной величины

∈ задача на параллелограммы
∈ задача на четырехугольники
∈ задача планиметрии

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *OSTIS-системе*:

- Какие классы задач известны для заданной предметной области?
- Какие способы решения класса задач известны для заданной предметной области?

Таким образом, база знаний *дочерней sc-системы* может быть пополнена как за счет использования компонентов из *Библиотеки многократно используемых компонентов sc-моделей баз знаний*, так и за счет добавления разработчиком в систему предметных знаний из рассматриваемой области.

Библиотека БГУИР

5 МЕТОДИКА ПРОЕКТИРОВАНИЯ МАШИН ОБРАБОТКИ ЗНАНИЙ

Машина обработки знаний любой системы, управляемой знаниями, определяет ее функциональные возможности и во многом обеспечивает удобство работы с ее контентом. Общие принципы проектирования машин обработки знаний по *Технологии OSTIS* рассмотрены в [4].

5.1 Расширение машины обработки знаний с применением многократно используемых компонентов OSTIS

В справочной системе по геометрии представлены типовые доказательства некоторых теорем и решения некоторых задач. Для удобства навигации по знаниям такого рода разработаны специальные *sc-агенты* и описаны соответствующие им компоненты в *Библиотеке многократно используемых компонентов sc-машин*:

– компонент библиотеки, *sc-агент* поиска доказательства для заданного утверждения;

– компонент библиотеки, *sc-агент* поиска решения заданной задач.

Для указанных компонентов *сопутствующими компонентами** являются:

– компонент библиотеки, команда поиска доказательства для заданного утверждения;

– компонент библиотеки, команда поиска решения заданной задач.

Также *сопутствующим компонентом** для указанных является *Раздел. Предметная область вопросов и задач*, но данный компонент уже был внедрен в систему ранее.

После того как в систему были добавлены указанные компоненты, конечный пользователь системы получил возможность увидеть полный *sc-текст доказательства* выбранного утверждения или решения выбранной задачи, как показано на рисунках 5.1 и 5.2.

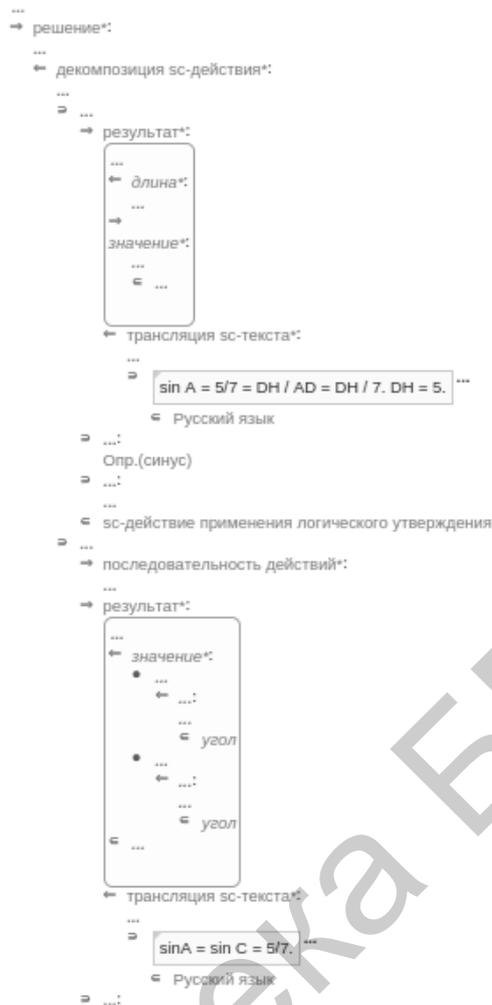


Рисунок 5.1 – Фрагмент результата работы *sc-агента поиска решения задачи*

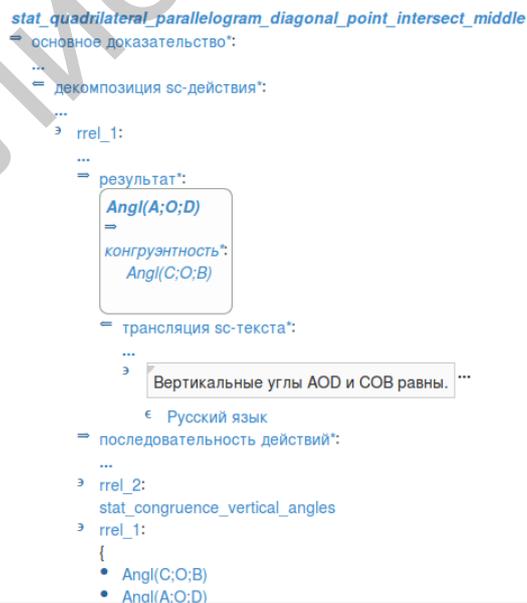


Рисунок 5.2 – Фрагмент результата работы *sc-агента поиска доказательства*

5.2 Расширение машины обработки знаний на основе собственных разработок

В процессе разработки конкретной дочерней системы по *Технологии OSTIS* может оказаться, что необходимый компонент такой системы, хотя и может использоваться в целом ряде систем, но пока еще не реализован другими разработчиками или реализован, но по каким-либо причинам пока еще не вошел в состав *Библиотеки многократно используемых компонентов OSTIS*. В таком случае разработчик дочерней *сс-системы* может самостоятельно разработать нужный компонент и, поскольку *Технология OSTIS* является открытой технологией, попробовать внедрить его в состав соответствующей *библиотеки многократно используемых компонентов*.

Одним из важнейших достоинств интеллектуальной справочной системы по геометрии является то, что она строит ответы на заданные пользователем вопросы (задачи) не только путем поиска и локализации (выделения) этих ответов в рамках текущего состояния базы знаний, но и путем генерации этих ответов, если их нет в текущем состоянии базы знаний. Интеллектуальная справочная система по геометрии, например, способна решать различные вычислительные задачи и задачи на доказательство. С этой целью был разработан интеллектуальный решатель задач, который, однако, разрабатывался без привязки к конкретной предметной области и может войти в состав *Библиотеки типовых подсистем интеллектуальных систем*.

В процессе решения конкретной задачи решателем были выделены **шесть ключевых этапов**:

- этап задания условия геометрической задачи;
- этап работы поисковых операций:
 - а) поиск нахождения значения указанной величины;
 - б) поиск нахождения доказательства для указанного утверждения;
- этап аналогии;
- этап применения стратегий решения задач:
 - а) «итерационный обход»;
 - б) «обход тезаурусом»;
- этап применения правил логического вывода:
 - а) применение имплицативных высказываний;
 - б) применение утверждений с эквивалентными частями;
- этап оптимизации сгенерированных знаний и сборки мусора.

5.3 Этап задания условия геометрической задачи

Выбранная нами модель представления знаний в виде семантических сетей предполагает, что все ключевые понятия базы знаний и связи между ними представлены в виде узлов и дуг, то есть принимают вид связного графа. Решатель опирается на граф условия и параметры запроса. Граф условия – условие задачи, представленное на языке семантических сетей (SC). Параметры запроса – узлы графа условия, к которым обращается пользователь при постановке задачи. Предположим, что нам известны длины одного из катетов и гипотенузы, и необходимо найти длину второго катета. Параметром запроса станет узел, содержащий длину неизвестного катета (рисунок 5.3).

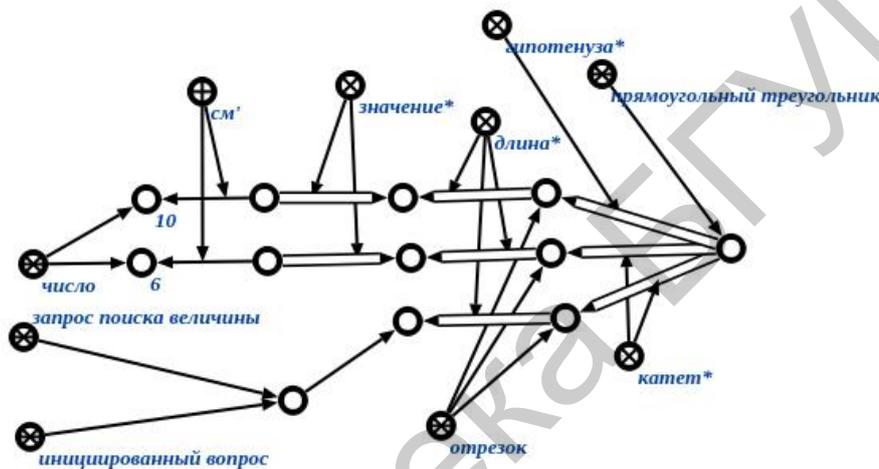


Рисунок 5.3 – Граф условия

5.4 Этап работы поисковых операций

Вне зависимости от типа задачи всегда имеется вероятность того, что данная задача уже была решена системой ранее или системе уже откуда-либо известен ответ на поставленный вопрос. Соответственно, первым шагом к решению поставленной задачи станет использование поисковой операции, соответствующей классу решаемой задачи.

– Класс задач, ориентированный на поиск значения неизвестной величины, решается путем вывода значения искомой величины с пошаговым выводом действий в виде примененных в ходе решения утверждений и результатов их применения. В случае, если граф условия соответствует одному из шаблонов решенных задач, то применение операции вывода решения для заданной задачи позволит решателю с минимальными затратами времени предоставить пользователю ответ.

– Класс задач, ориентированный на поиск доказательства, принимает в качестве параметра запроса не узел неизвестной величины, а *sc-структуру*, содержащую произвольное высказывание, которое в ходе решения задачи будет либо доказано, вследствие применения вспомогательных логических утверждений, либо опровергнуто, что будет свидетельствовать о его невыполнимости в текущем состоянии базы знаний. Если доказательство высказывания ранее было вынесено в качестве типового, то результат выполнения *операции поиска доказательства для заданного утверждения* предоставит пользователю ответ в виде декомпозиции шагов доказательства с указанием последовательности применения утверждений.

5.5 Этап вывода по аналогии

При решении задачи мы часто сталкиваемся с тем, что одну величину можно выразить через другую. Идеальным случаем является наличие между объектами бинарного неориентированного отношения типа *конгруэнтность** или *подобие**, позволяющего на основании знаний об одном объекте делать достоверные выводы о другом. Так как основной характеристикой разрабатываемых систем является их способность к осуществлению различного рода логического вывода, то это подразумевает под собой применение системой не только стандартных алгоритмов, но и других логических схем, таких как аналогия, что позволяет сократить процесс поиска решения в разы. Пример применения аналогии для рассмотренной в подразделе 5.3 задачи представлен на рисунке 5.4.

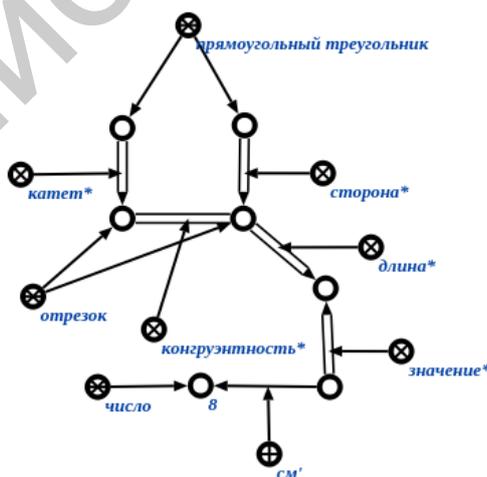


Рисунок 5.4 – Аналогия

5.6 Этап применения стратегий решения задач

На данном этапе осуществляется выбор между различными стратегиями решения задач, и при необходимости параллельный запуск различных стратегий. На данный момент интеллектуальный решатель реализует комбинированную стратегию, рассмотренную в работе [5].

Вначале рассматривается некоторый объект, для него осуществляется поиск всех классов объектов, которым он принадлежит. Далее для каждого класса осуществляется поиск утверждений, справедливых для данного класса объектов. При рассмотрении каждого утверждения осуществляется попытка применить его в рамках некоторой семантической окрестности рассматриваемого объекта.

Если ни одно из утверждений класса применить не удастся, то стратегия продолжает свою работу и переходит к другим объектам. Когда применения одного утверждения оказывается недостаточно для получения ответа, то стратегия запускается заново, с узла запроса, и поиск решения продолжается.

Были разработаны два возможных варианта обхода стратегии:

а) *«итерационный обход»*: попытавшись применить по разу каждое из утверждений, на основании того, был ли выявлен какой-то прогресс в ходе решения, то есть сгенерированы ли какие-либо новые знания в семантической окрестности задачи, мы определяем, стоит ли производить еще один обход;

б) *«обход тезаурусом»*: по мере единственного обхода мы применяем каждое из утверждений по разу и формируем словарь, в который войдут все утверждения, отсеченные на этапе оптимизации сгенерированных знаний по причине нехватки известных параметров; после успешного применения логического высказывания мы поочередно применяем каждое из утверждений, находящихся в словаре, с целью получить желаемый результат, сокращая таким образом количество утверждений, рассматриваемых на каждом шаге обхода.

Когда при *«итерационном обходе»* прогресс в решении остановится, а при *«обходе тезаурусом»* все логические утверждения окажутся пройденными, можно утверждать, что известных параметров для нахождения величины было недостаточно (класс задач, ориентированный на поиск неизвестной величины) или заданное утверждение не является истинным в текущем состоянии базы знаний (класс задач, ориентированный на поиск доказательства).

Для выигрыша во времени решатель пользуется специальным набором утверждений, связка которых соединена с классами объектов через отношение *основные утверждения** (рисунок 5.5).

Необходимость введения такой связки обоснована еще и тем, что не все логические утверждения имеют прикладной характер: часть из них используется для дидактических целей, например при обучении, а не для применения в рамках задач.

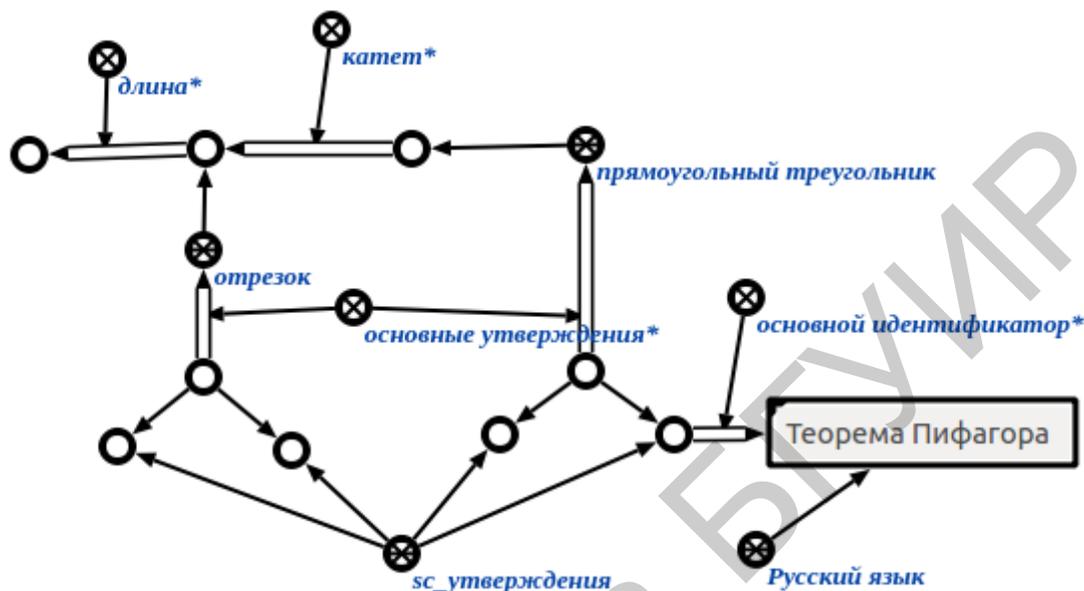


Рисунок 5.5 – Стратегия

5.7 Этап применения правил логического вывода

На данном этапе происходит попытка применения утверждения, полученного на предыдущем шаге, с целью генерации в системе новых знаний. Если такое применение справедливо (например, посылка истинна) и имеет смысл (в результате применения будут сгенерированы новые знания), то осуществляется генерация новых знаний на основе одного из правил логического вывода. Если в данном контексте вывод на основе данного утверждения невозможен или нецелесообразен, решение возвращается на предыдущий этап. В случае успешного применения утверждения происходит переход к следующему этапу решения.

На данный момент решатель поддерживает возможность применения двух типов утверждений:

а) *применение имплицативных высказываний*: граф условия соотносится с левой частью и в случае успеха дополняется *sc-конструкцией*, состоящей из *sc-констант*, содержащейся в правой части импликации;

б) *применение утверждений с эквивалентными частями*: сначала для графа условия находится часть логического утверждения, изоморфная ему,

удаление конструкций, ставших ненужными и по каким-либо причинам не удаленных на предыдущих этапах решения.

Если все этапы решения выполнены успешно, то решение возвращается ко второму этапу, и в результате выполнения поисковой операции пользователь видит полученный ответ. В противном случае процесс повторяется еще раз, то есть мы возвращаемся к этапу стратегии.

В конце решения задачи граф условия примет вид, представленный на рисунке 5.7.

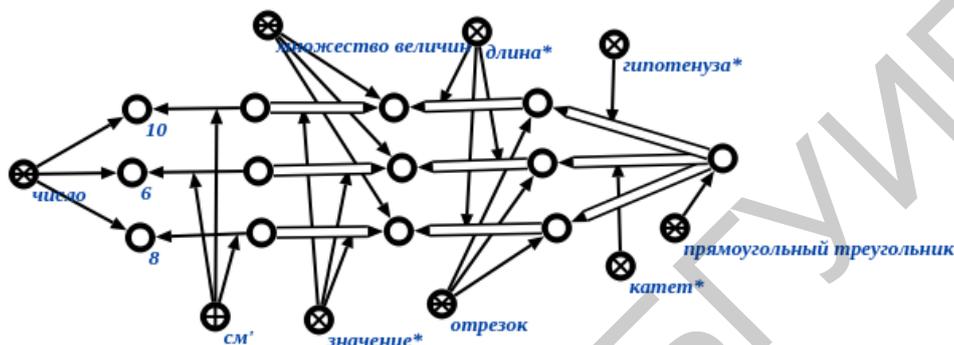


Рисунок 5.7 – Финальный граф

В будущем после тестирования и полной отладки решателя планируется его внедрить в *Библиотеку типовых подсистем интеллектуальных систем*.

Добавление компонента в *Библиотеку многократно используемых компонентов OSTIS* или обновление уже существующего может осуществляться двумя способами:

а) в случае *платформенно-зависимого многократно используемого компонента OSTIS* логично использовать стандартные системы контроля версий и с их помощью с учетом прав доступа вносить предложения по изменению компонента;

б) в случае *платформенно-независимого многократно используемого компонента OSTIS* предложение по добавлению или изменению компонента может формироваться прямо в *sc-памяти* средствами *SC-кода* и рассматриваться администратором базы знаний с учетом методики проектирования баз знаний (в общем случае любой *платформенно-независимый многократно используемый компонент OSTIS* рассматривается как часть базы знаний).

6 МЕТОДИКА ПРОЕКТИРОВАНИЯ КОМПОНЕНТОВ ИНТЕРФЕЙСА

В конкретных прикладных системах, разработанных по *Технологии OSTIS* помимо *Ядра sc-моделей пользовательских интерфейсов* и средств работы с базовыми языками представления *SC-кода* может оказаться целесообразным использование в системе дополнительных средств отображения или редактирования знаний какого-либо конкретного вида. Подробно методика и средства проектирования пользовательских интерфейсов по *Технологии OSTIS* рассмотрена в [2].

Для рассматриваемой в данной работе справочной системы по геометрии был разработан специализированный редактор геометрических чертежей, совместимый с выбранным вариантом реализации ядра пользовательских интерфейсов. Данный компонент поставляется в виде набора исходных кодов и может быть встроен в реализацию ядра по инструкции, приведенной в описании данной реализации [7]. После встраивания редактора в систему пользователь получит возможность создавать геометрические чертежи из доступных элементов, сохранять в файл и загружать из файла созданный чертеж (рисунок 6.1).

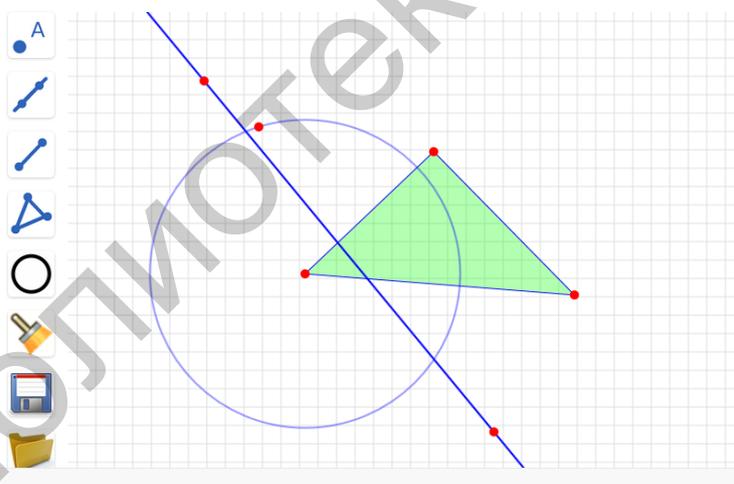


Рисунок 6.1 – Пример использования редактора геометрических чертежей

7 ОРГАНИЗАЦИЯ ВЫПОЛНЕНИЯ СТУДЕНЧЕСКИХ ПРОЕКТОВ

Список тем студенческих проектов и соответствующих команд исполнителей **предварительно формируется** в конце каждого семестра (во время защиты курсовых проектов), **уточняется и утверждается** кафедрой в течении первых двух недель от начала каждого семестра.

Переход студентов из одного коллективного проекта в другой возможен, но только после завершения семестра и до конца первой недели следующего семестра.

Каждый студент в первую неделю каждого семестра обязан согласовать и уточнить свою тему и задание с руководителем курсового проектирования в текущем семестре, а также проверить, как результаты этого согласования отражены в документах на кафедре.

Студенту выдается тема индивидуального проекта без согласования с ним (решением кафедры) при невыполнении в течение первых двух недель от начала семестра одного из следующих трех условий:

- студент должен быть официально сохранен командой в составе исполнителей студенческого проекта, в котором он участвовал в прошлом семестре;
- студент должен быть официально принят командой в состав исполнителей нового студенческого проекта;
- студент должен подтвердить или уточнить тему своего индивидуального задания.

Во время семестра каждый студент должен активно участвовать в реализации того проекта, в который он включен, но может также участвовать и в реализации других студенческих проектов. Вся указанная деятельность должна быть отражена в индивидуальном отчете каждого студента и учитывается при выставлении индивидуальных оценок.

Таким образом, участниками и разработчиками каждого студенческого проекта могут быть студенты любого курса (в том числе и первого) и магистранты. При этом в каждом отдельном проекте желательно, чтобы были представлены студенты всех курсов, принимающих участие в проектировании интеллектуальных систем. Это необходимо для обеспечения преемственности и сбалансированности команды разработчиков.

Научно-техническое руководство студенческими проектами осуществляется преподавателями (чаще всего аспирантами) кафедры. Научный руководитель проекта не должен оказывать влияние на организацию

выполнения проекта. Распределение заданий между членами команды, контроль исполнительской дисциплины, тестирование и интеграция результатов, отбор новых членов команды осуществляются самостоятельно, то есть вся ответственность за ход и результаты студенческого проекта лежит исключительно на членах команды. Научный руководитель исполняет роль консультанта и советника, но не организатора. При этом исполнители могут обращаться за консультацией к любому аспиранту и преподавателю кафедры, руководителю (консультанту), техническому консультанту, эксперту по тематике разрабатываемой системы. Все они оказывают содействие, но не несут никакой ответственности за результаты разработки.

Участники проекта должны активно искать любых полезных партнеров (консультантов, экспертов, инвесторов) и взаимодействовать с ними, а также активно содействовать развитию технологии проектирования интеллектуальных систем в тех направлениях, которые необходимы для реализации проекта.

Участники также должны взаимодействовать с близкими проектами, особенно с теми, интеграция с которыми логически обоснована. В идеале необходимо переходить от разработки отдельных интеллектуальных систем к разработке коллективных интеллектуальных систем.

Команда исполнителей проекта в самом общем случае имеет следующую структуру:

- научный руководитель;
- научно-технический консультант;
- эксперт предметной области;
- системный администратор (платформа и интерфейс с другими системами);
- администратор базы знаний;
- администратор машины обработки знаний;
- администратор пользовательского интерфейса;
- администратор подсистемы управления информационной безопасностью;
- администратор подсистемы информационного обеспечения и управления проектированием;
- администратор подсистемы адаптивного повышения квалификации пользователей системы (повышение эффективности эксплуатации);
- рядовые исполнители (разработчик базы знаний, разработчик машины обработки знаний, разработчик программного интерфейса и др.).

Формирование команды исполнителей проекта осуществляется по следующим правилам:

- войти в состав существующего студенческого проекта можно до начала семестра и по результатам собеседования с представителями коллективного проекта;

- возможно одновременное участие в двух проектах, но один из них должен быть основным;

- выход из состава разработчиков студенческого проекта может быть осуществлен на основании экспертного заключения и/или решения кафедры, кроме того, на основании коллективного решения команды разработчиков проекта, а также по личной инициативе с указанием, в какой проект осуществляется переход;

- переход в другой проект может быть осуществлен только в конце семестра до начала экзаменационной сессии;

- переход разработчика в течение семестра в другой проект запрещен;

- состав исполнителей для вновь созданного студенческого проекта должен быть сформирован до начала семестра;

- на каждом этапе развития студенческого проекта должны быть четко определены обязанности основных разработчиков: системного администратора, администратора базы знаний, администратора машин обработки знаний, администратора пользовательского интерфейса.

По результатам каждого семестра могут происходить следующие организационные изменения:

- состав разработчиков каждого студенческого проекта может изменяться (могут появляться новые разработчики, а некоторые разработчики могут быть переведены на другие проекты по собственной инициативе, решению команды либо решению кафедры);

- некоторые студенческие проекты по решению кафедры могут быть приостановлены, а команды разработчиков распущены;

- могут появляться новые студенческие проекты по инициативе студентов или кафедры.

Календарный план по организации и контролю разработки студенческих проектов в рамках каждого семестра включает в себя следующие мероприятия:

- корректировка набора студенческих проектов и состава исполнителей в течение первых двух недель от начала семестра (осуществляется на заседании кафедры с учетом мнения команд разработчиков студенческих проектов);

– формирование индивидуальных заданий по курсовому проектированию в соответствии с участием каждого студента в индивидуальном и/или коллективном студенческом проекте:

✓ до 15 сентября (осенний семестр);

✓ до 24 февраля для 1–3 курсов и до 15 февраля для 4–5 курсов (весенний семестр);

– 1-я контрольная точка о ходе выполнения индивидуальных заданий студентами:

✓ до 15 октября (осенний семестр);

✓ до 15 марта (весенний семестр);

– 2-я контрольная точка о ходе выполнения индивидуальных заданий студентами:

✓ до 15 ноября (осенний семестр);

✓ до 15 апреля (весенний семестр);

– 3-я контрольная точка – оформление документации разрабатываемой системы в ее базе знаний, написание индивидуальных отчетов и предоставление полученных результатов руководителю курсового проектирования в текущем семестре:

✓ до 15 декабря для 1–3 курсов и до 5 декабря для 4–5 курсов (осенний семестр);

✓ до 20 мая для 1–3 курсов и до 30 апреля для 4–5 курсов (весенний семестр);

– экспертиза полученных результатов рабочей комиссии кафедры (оценка качества написанной документации и индивидуальных отчетов, тестирование разработанной версии системы рабочей комиссией кафедры):

✓ до 20 декабря для 1–3 курсов и до 11 декабря для 4–5 курсов (осенний семестр);

✓ до 1 июня для 1–3 курсов и до 5 мая для 4–5 курсов (весенний семестр);

– защита студенческих проектов с участием широкого круга экспертов:

✓ до 30 декабря для 1–3 курсов и до 21 декабря для 4–5 курсов (осенний семестр),

✓ до 10 июня для 1–3 курсов и до 15 мая для 4–5 курсов (весенний семестр).

8 ЗАЩИТА СТУДЕНЧЕСКИХ ПРОЕКТОВ, ИНДИВИДУАЛЬНАЯ АТТЕСТАЦИЯ СТУДЕНТОВ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ

Защита студенческих проектов осуществляется в виде презентации результатов научно-исследовательской деятельности и ее практической реализации, полученных командой разработчиков.

Аттестация уровня развития каждого студенческого проекта осуществляется в конце каждого семестра (с 11 декабря и с 5 мая соответственно) и определяется:

- результатами тестирования текущего состояния прототипа разрабатываемой системы и объемом выполненных работ за отчетный семестр;
- своевременностью предоставления результатов работы;
- качеством оформления документации проекта (в базе знаний разрабатываемой системы);
- убедительностью представленной презентации проекта и четкостью ответов на вопросы (для лучших проектов);
- приобретенным опытом участия в различных конкурсах.

Состав экспертной комиссии для аттестации студенческих проектов и индивидуальной аттестации студентов, участвующих в коллективных студенческих проектах, утверждается кафедрой за месяц до конца каждого семестра.

В состав указанной комиссии входят все магистранты, аспиранты и преподаватели кафедры, а также представители организаций, сотрудничающих с кафедрой.

Для выполнения этой работы чаще всего привлекаются магистранты кафедры ИИТ, которые знакомы с аттестуемыми проектами, но не являются членами соответствующих команд.

Начиная с 5 декабря и с 30 апреля разработчики каждого студенческого проекта представляют аттестационной комиссии:

- текущую версию разрабатываемой системы или ее прототип (желательно как Интернет-ресурс);
- полную документацию текущей версии разрабатываемой системы, которая оформляется как часть базы знаний этой системы и содержит четкое описание отличий текущей версии системы от предыдущей, являющейся результатом прошлого семестра;
- план развития текущей версии системы в следующем семестре.

Защита проекта происходит в форме презентации, отображающей содержание проекта, а именно:

- актуальность: для кого и для чего предназначена система. Степень востребованности;

- аналоги;

- достоинства и преимущества разрабатываемой системы по сравнению с аналогичными системами, при этом необходимо продемонстрировать те достоинства, которые были реализованы за отчетный период;

- основные результаты, полученные за отчетный семестр;

- планы на следующий семестр;

- планируемый состав разработчиков на следующий семестр;

- структуру планируемого распределения обязанностей в команде.

В результате защиты проекта определяется оценка общего уровня развития проекта за отчетный семестр.

Комиссия, созданная решением кафедры, по результатам тестирования текущей версии разрабатываемой системы и анализа документации проекта выставляет экспертную оценку команде, выполняющей проект в отчетном семестре, в соответствии с критериями, выделенными в бюллетени.

Экспертная оценка студенческого проекта включает в себя:

- оценку качества презентации проекта (доклад, иллюстративный материал, ответы на вопросы);

- оценку качества текущей версии интеллектуальной системы в сравнении с предыдущей версией;

- оценку качества документации;

- оценку активности по продвижению проекта (в том числе участие в конкурсах проектов);

- оценку вклада в развитие технологии;

- оценку качества команды и организации ее деятельности.

Критерии оценки уровня развития студенческого проекта следующие:

- наличие и качество действующего доступного прототипа или версии разрабатываемой системы;

- качество и полнота документации по текущей версии разрабатываемой системы;

- объем выполненных работ за семестр;

- качество демонстрационного диалога (убедительность показа достоинств);

– качество отражения в базе знаний вклада каждого разработчика в развитие проекта;

– качество кадрового состава и его сбалансированность (в том числе наличие студентов со всех потоков);

– вклад в *Технологию OSTIS*;

– уровень экспертов предметной области;

– участие в конкурсах;

– взаимодействие с потенциальными заказчиками, инвесторами и потребителями.

Качество команды студенческого проекта определяется:

– уровнем мотивации и отсутствием немотивированных участников;

– уровнем и активностью экспертов предметной области;

– отсутствием большого разброса в уровне активности и вклада в развитие проекта;

– атмосферой;

– сбалансированностью представительства всех потоков студентов и магистрантов.

Указанная комиссия на основе индивидуальных отчетов студентов выставляет также индивидуальные оценки по курсовому проектированию за отчетный семестр. Эта оценка учитывает результативность деятельности студента за этот семестр и может отличаться от оценки деятельности команды, в которую он официально входит (но, как правило, не может превышать оценку команды более чем на три балла).

Индивидуальная экспертная оценка студента по курсовому проектированию определяется следующими факторами:

– уровнем ответственности отношения к тому проекту, к которому студент официально приписан (основному для себя проекту);

– объемом и качеством всей выполненной работы (как в рамках своего основного проекта, так и в рамках других проектов);

– оценкой, которую получила вся команда того проекта, к которому студент приписан;

– четкостью индивидуального отчета;

– своевременностью представления индивидуального отчета;

– эффективностью взаимодействия с другими участниками коллективного проекта (однокурсниками, студентами младших и старших курсов, экспертами, заказчиками, магистрантами, аспирантами, преподавателями);

– индивидуальным вкладом в *Метапроект IMS*, который направлен на разработку интеллектуальной метасистемы поддержки компонентного проектирования интеллектуальных систем.

Если команда, выполняющая проект, работала вяло и нерезультативно в отчетном семестре, то решением указанной комиссии она может быть расформирована.

Если студент за отчетный период в рамках своего основного проекта работал вяло, безответственно и не соблюдал график учебного процесса (имел пропуски и задолженности), то решением экспертной комиссии этот студент может быть выведен из состава команды указанного студенческого проекта. В таком случае студент не имеет права перехода в другие коллективные студенческие проекты, при этом имеет право выбора темы индивидуального задания для курсового проектирования.

Оценка студенческого проекта снижается (по сравнению с оценкой, полученной при защите) за несвоевременное представление результатов работы во время семестра. Индивидуальная оценка по курсовому проектированию снижается за несвоевременное предоставление индивидуального отчета.

Если студент участвовал в отчетном семестре в коллективном студенческом проекте, то его итоговая индивидуальная оценка по курсовому проектированию представляет собой среднее арифметическое:

- усредненной экспертной оценки индивидуальной деятельности студента, участвующего в коллективном студенческом проекте;
- усредненной экспертной оценки уровня самого указанного коллективного проекта.

Если студент в отчетном семестре выполнял индивидуальный студенческий проект, то его итоговая индивидуальная оценка по курсовому проектированию совпадает с усредненной экспертной оценкой уровня развития указанного индивидуального студенческого проекта.

Итоговая индивидуальная оценка студента по курсовому проектированию утверждается на заседании кафедры.

9 ПЛАНИРОВАНИЕ КУРСОВОГО ПРОЕКТИРОВАНИЯ НА СЛЕДУЮЩИЙ СЕМЕСТР

Планирование курсового проектирования на следующий семестр осуществляется в ходе аттестации студенческих проектов текущего семестра и включает в себя следующие этапы:

1. Утверждается и публикуется на сайте кафедры предположительный список проектов на следующий семестр с указанием их приоритета.

Проект, выполнявшийся в текущем семестре, может быть:

- ✓ индивидуальным;
- ✓ скорректированным по тематике или введенным в состав другого проекта (чаще всего нового);
- ✓ преобразованным по статусу (из индивидуального в коллективный).

Кроме того, в список проектов могут быть добавлены новые проекты.

2. Утверждается список студентов, которые в следующем семестре должны выполнять индивидуальные проекты (в том числе те, которые они выполняли).

3. Утверждается список студентов, которые должны выйти из состава коллективных проектов, но могут войти в состав других коллективных проектов.

4. Утверждается список остальных студентов, которые не имеют никаких ограничений в выборе тематики курсового проектирования на следующий семестр.

5. Каждый студент пишет заявление о выборе курсового проекта на следующий семестр. Заявление согласуется с соответствующими командами (в форме собеседования) и утверждается кафедрой.

6. Утверждается и публикуется окончательный список студенческих (курсовых) проектов на следующий семестр и распределение всех студентов по этим проектам.

7. Только после этого в ведомости и зачетки выставляются оценки по курсовому проектированию за текущий семестр.

ПРИЛОЖЕНИЕ А (обязательное)

СТРУКТУРА БАЗЫ ЗНАНИЙ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

База знаний <разрабатываемой системы>

⇐декомпозиция раздела*:

{

- Документация. <Предметная часть базы знаний>

/* Например, для «Интеллектуальная система поддержки деятельности кафедры ИИТ» предметная часть базы знаний будет «Кафедра ИИТ». */

- Контекст <Предметной части базы знаний> в рамках Глобальной базы знаний

- Документация <разрабатываемая система>

⇐декомпозиция раздела*:

{

- Обоснование разработки <разрабатываемая система >

⇐декомпозиция раздела*:

{

- Раздел. Тип, назначение и состав <разрабатываемой системы >

- Раздел. Актуальность <разрабатываемой системы >

/*почему данный проект на данный момент является важным и насущным для жизни и развития представителей выбранной целевой аудитории, востребованность проекта*/

- Раздел. Аналоги <разрабатываемой системы> и их недостатки

- Раздел. Попытки устранения недостатков аналогов <разрабатываемой системы >

- Раздел. Что препятствует устранению недостатков аналогов <разрабатываемой системы >

- Раздел. Принципы, лежащие в основе <разрабатываемой системы >

- Раздел. Предпосылки создания <разрабатываемой системы >

- Раздел. Характеристики, используемые для сравнительного анализа <разрабатываемой системы >, и набор тестовых задач

- Раздел. Конкурентные преимущества и достоинства <разрабатываемой системы>

- /* конкурентные преимущества разрабатываемой БЗ; конкурентные преимущества разрабатываемой МОЗ (в том числе возможности решателя задач); конкурентные преимущества пользовательского интерфейса разрабатываемой системы (мощность языка вопросов, отображение информации, управление интерфейсом); конкурентные преимущества платформы, обеспечивающей интерпретацию логико-семантической модели разрабатываемой интеллектуальной системы */

- Раздел. Пример диалога <разрабатываемой системы> с ее пользователями, демонстрирующий возможности и достоинства <разрабатываемой системы>

- /*диалог структурировать на основе логико-семантической типологии задаваемых вопросов и задач*/

- Раздел. Новизна решений, реализуемых в <разрабатываемой системе>

- /*основные идеи и принципы, лежащие в основе разрабатываемой системы, в том числе методика компонентного проектирования */

}

- Обоснование выбора. Платформа реализации <разрабатываемой системы>

- Структура. База знаний <разрабатываемой системы>

- Документация. Машина обработки знаний <разрабатываемой системы>

- /*что система должна уметь, сборник задач, которые должна уметь решать разрабатываемая система, спецификация разработанных агентов*/

- Документация. Пользовательский интерфейс <разрабатываемой системы>

- /*как система взаимодействует с конечным пользователем – внешние языки, управление интерфейсом, типология команд; характеристика главного окна*/

- Документация. Подсистема консультационного обслуживания и обучения конечных пользователей <разрабатываемой системы>

- Документация. Подсистема автоматизации проектирования <разрабатываемой системы>

- Документация. Подсистема управления проектированием <разрабатываемой системы >
 - Документация. Подсистема управления информационной безопасностью <разрабатываемой системы >
 - }
 - История и текущие процессы эксплуатации <разрабатываемой системы >
 - Раздел. Проект <разрабатываемой системы >. История текущих процессов и план развития <разрабатываемой системы >
- ⇐декомпозиция раздела*:
- {
 - История развития <разрабатываемой системы >
 - Текущие процессы развития <разрабатываемой системы >
 - План развития <разрабатываемой системы >
- /*последующие версии и модификации создаваемого продукта, этапы (план) эволюции продукта (по различным направлениям); конкретный перечень проектных задач на следующий отчетный период; желательные сроки выполнения и исполнители каждой проектной задачи*/
- }
 - }

ПРИЛОЖЕНИЕ Б

(обязательное)

СПИСОК ВОЗМОЖНЫХ ВАРИАНТОВ ТЕМ КУРСОВОГО ПРОЕКТА

Для 3-го семестра:

1. Фрагмент базы знаний интеллектуальной справочной системы по геометрии.
2. Фрагмент базы знаний интеллектуальной справочной системы по физике.
3. Фрагмент базы знаний интеллектуальной справочной системы по астрономии.
4. Фрагмент базы знаний интеллектуальной справочной системы по теории графов.
5. Фрагмент базы знаний интеллектуальной справочной системы по лингвистике.
6. Фрагмент базы знаний интеллектуальной справочной системы по теории множеств.
7. Фрагмент базы знаний интеллектуальной справочной системы по географии.
8. Фрагмент базы знаний интеллектуальной справочной системы по числовым моделям.
9. Фрагмент базы знаний интеллектуальной справочной системы по фармакологии.
10. Фрагмент базы знаний интеллектуальной справочной системы по музыке.
11. Фрагмент базы знаний интеллектуальной справочной системы по химии.
12. Фрагмент базы знаний интеллектуальной справочной системы по истории.
13. Фрагмент базы знаний интеллектуальной справочной системы по изобразительному искусству.
14. Фрагмент базы знаний интеллектуальной справочной системы по автодиагностике.
15. Фрагмент базы знаний интеллектуальной справочной системы по кулинарии.
16. Фрагмент базы знаний интеллектуальной справочной системы по правилам дорожного движения.

Для 4-го семестра:

1. Набор операций информационного поиска интеллектуальной справочной системы по геометрии.
2. Набор операций информационного поиска интеллектуальной справочной системы по физике.
3. Набор операций информационного поиска интеллектуальной справочной системы по астрономии.
4. Набор операций информационного поиска интеллектуальной справочной системы по теории графов.
5. Набор операций информационного поиска интеллектуальной справочной системы по лингвистике.
6. Набор операций информационного поиска интеллектуальной справочной системы по теории множеств.
7. Набор операций информационного поиска интеллектуальной справочной системы по географии.
8. Набор операций информационного поиска интеллектуальной справочной системы по числовым моделям.
9. Набор операций информационного поиска интеллектуальной справочной системы по фармакологии.
10. Набор операций информационного поиска интеллектуальной справочной системы по музыке.
11. Набор операций информационного поиска интеллектуальной справочной системы по химии.
12. Набор операций информационного поиска интеллектуальной справочной системы по истории.
13. Набор операций информационного поиска интеллектуальной справочной системы по изобразительному искусству.
14. Набор операций информационного поиска интеллектуальной справочной системы по автодиагностике.
15. Набор операций информационного поиска интеллектуальной справочной системы по кулинарии.
16. Набор операций информационного поиска интеллектуальной справочной системы по правилам дорожного движения.

Для 5-го семестра:

1. Набор операций логического вывода интеллектуальной справочной системы по геометрии.

2. Набор операций логического вывода интеллектуальной справочной системы по физике.
3. Набор операций логического вывода интеллектуальной справочной системы по астрономии.
4. Набор операций логического вывода интеллектуальной справочной системы по теории графов.
5. Набор операций логического вывода интеллектуальной справочной системы по лингвистике.
6. Набор операций логического вывода интеллектуальной справочной системы по теории множеств.
7. Набор операций логического вывода интеллектуальной справочной системы по географии.
8. Набор операций логического вывода интеллектуальной справочной системы по числовым моделям.
9. Набор операций логического вывода интеллектуальной справочной системы по фармакологии.
10. Набор операций логического вывода интеллектуальной справочной системы по музыке.
11. Набор операций логического вывода интеллектуальной справочной системы по химии.
12. Набор операций логического вывода интеллектуальной справочной системы по истории.
13. Набор операций логического вывода интеллектуальной справочной системы по изобразительному искусству.
14. Набор операций логического вывода интеллектуальной справочной системы по автодиагностике.
15. Набор операций логического вывода интеллектуальной справочной системы по кулинарии.
16. Набор операций логического вывода интеллектуальной справочной системы по правилам дорожного движения.

Для 6-го семестра:

1. Компонент пользовательского интерфейса интеллектуальной справочной системы по геометрии.
2. Компонент пользовательского интерфейса интеллектуальной справочной системы по физике.
3. Компонент пользовательского интерфейса интеллектуальной справочной системы по астрономии.

4. Компонент пользовательского интерфейса интеллектуальной справочной системы по теории графов.
5. Компонент пользовательского интерфейса интеллектуальной справочной системы по лингвистике.
6. Компонент пользовательского интерфейса интеллектуальной справочной системы по теории множеств.
7. Компонент пользовательского интерфейса интеллектуальной справочной системы по географии.
8. Компонент пользовательского интерфейса интеллектуальной справочной системы по числовым моделям.
9. Компонент пользовательского интерфейса интеллектуальной справочной системы по фармакологии.
10. Компонент пользовательского интерфейса интеллектуальной справочной системы по музыке.
11. Компонент пользовательского интерфейса интеллектуальной справочной системы по химии.
12. Компонент пользовательского интерфейса интеллектуальной справочной системы по истории.
13. Компонент пользовательского интерфейса интеллектуальной справочной системы по изобразительному искусству.
14. Компонент пользовательского интерфейса интеллектуальной справочной системы по автодиагностике.
15. Компонент пользовательского интерфейса интеллектуальной справочной системы по кулинарии.
16. Компонент пользовательского интерфейса интеллектуальной справочной системы по правилам дорожного движения.

Для 7-го семестра:

1. Компонент библиотеки унифицированных логико-семантических моделей баз знаний.
2. Компонент библиотеки операций информационного поиска.
3. Компонент библиотеки операций логического вывода.
4. Компонент библиотеки операций поиска пути решения явно сформулированных задач.
5. Компонент библиотеки операций повышения качества баз знаний.
6. Компонент библиотеки унифицированных логико-семантических моделей пользовательских интерфейсов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Голенков, В. В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования / В. В. Голенков, Н. А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2012) : материалы II Междунар. науч.-техн. конф., Минск, 16–18 февраля 2012 г. / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2012. – С. 23–52.

2 Шункевич, Д. В. Средства поддержки компонентного проектирования систем, управляемых знаниями / Д. В. Шункевич [и др.]. // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2015) : материалы V Междунар. науч.-техн. конф., Минск, 19–21 февраля 2015 г. / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2015. – С. 79–88.

3 Давыденко, И. Т. Технология компонентного проектирования баз знаний на основе унифицированных семантических сетей / И. Т. Давыденко // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013) : материалы III Междунар. науч.-техн. конф., Минск, 21–23 февраля 2013 г. / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2013. – С. 185–190.

4 Шункевич, Д. В. Модели и средства компонентного проектирования машин обработки знаний на основе семантических сетей / Д. В. Шункевич // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013) : материалы III Междунар. науч.-техн. конф., Минск, 21–23 февраля 2013 г. / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2013. – С. 269–280.

5 Заливако, С. С. Семантическая технология компонентного проектирования интеллектуальных решателей задач / С. С. Заливако, Д. В. Шункевич // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2012) : материалы II Междунар. науч.-техн. конф., Минск, 16–18 февраля 2012 г. / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2012. – С. 297–315.

6 Корончик, Д. Н. Реализация хранилища унифицированных семантических сетей / Д. Н. Корончик // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013) : материалы III

Международ. науч.-техн. конф., Минск, 21–23 февраля 2013 г. / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2013 – С. 125–129.

7 SC-WEB repository [Электронный ресурс]. – 2017. – Режим доступа : <https://github.com/Ivan-Zhukau/sc-web>.

8 Интеллектуальная Метасистема IMS [Электронный ресурс]. –2016. – Режим доступа : <http://ims.ostis.net>.

9 sc-machine [Электронный ресурс]. – 2016. – Режим доступа : <https://github.com/deniskoronchik/sc-machine>.

10 Структура sc-машины [Электронный ресурс]. – 2016. – Режим доступа: <https://github.com/deniskoronchik/sc-machine/wiki>.

Библиотека БГУИР

Учебное издание

Голенков Владимир Васильевич
Гулякина Наталья Анатольевна
Гракова Наталья Викторовна и др.

**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ.
КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

ПОСОБИЕ

Редактор *М. А. Зайцева*

Компьютерная правка, оригинал-макет *В. М. Задоля*

Подписано в печать 14.07.2017. Формат 60x84 1/16. Бумага офсетная. Гарнитура «Таймс».

Отпечатано на ризографе. Усл. печ. л. 3,84. Уч.-изд. л. 4,0. Тираж 80 экз. Заказ. 416.

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 07.04.2014.

ЛП №02330/264 от 14.04.2014.

220013, Минск, П. Бровки, 6