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.

Библиотека БГУИР

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, обусловливает 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 отчуждаемого 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 EKB – 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.

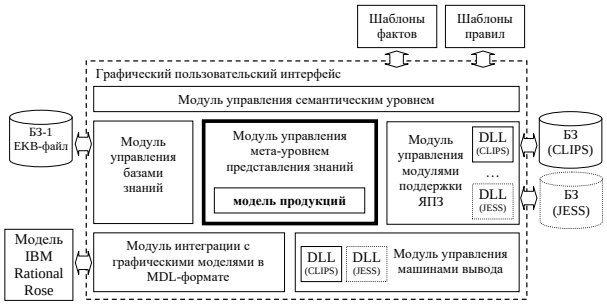

Рисунок 1 – Архитектура системы программирования продукционных БЗ

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 хранить знания в собственном независимом формате. Программная реализация структуры на уровне хранения знаний выполнена при помощи XML.

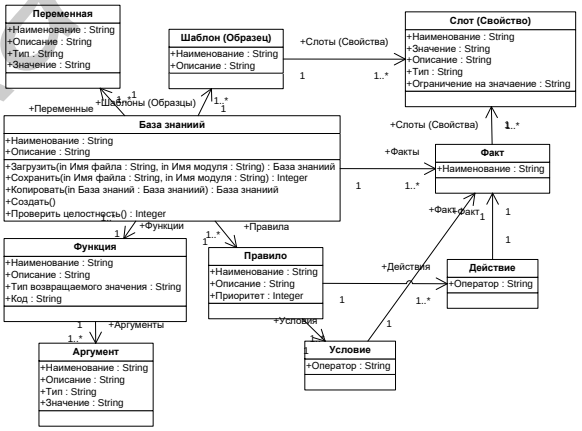

Рисунок 2 – Обобщенная модель продукций

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];

Библиотека БГУИР

- «Справочная информация» - содержит окна справочных сведений и истории действий пользователя (с возможностью их отмены).

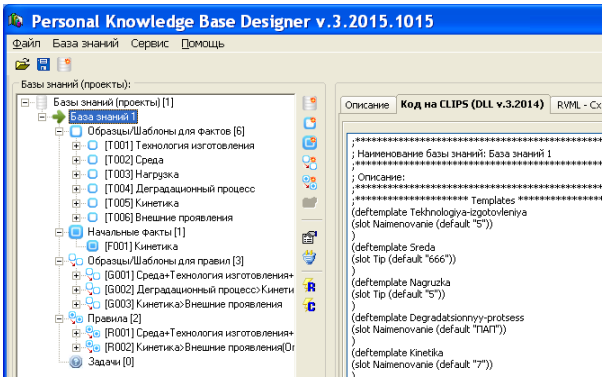

Рисунок 3 – Основное рабочее пространство пользователя

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).

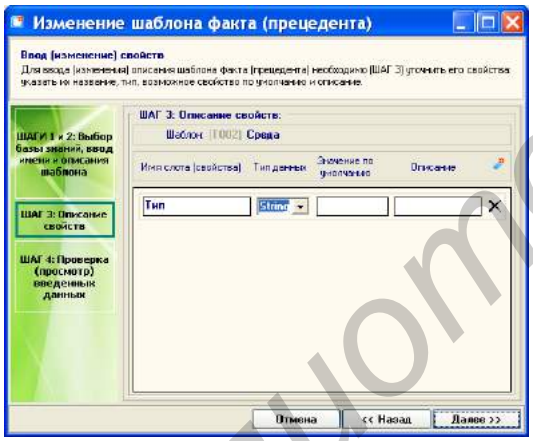

Рисунок 4 – Изменение/Ввод шаблона факта

Подобные мастера применяются при вводе и редактировании фактов и правил (рисунок 5) и при тестировании баз знаний (рисунок 6).

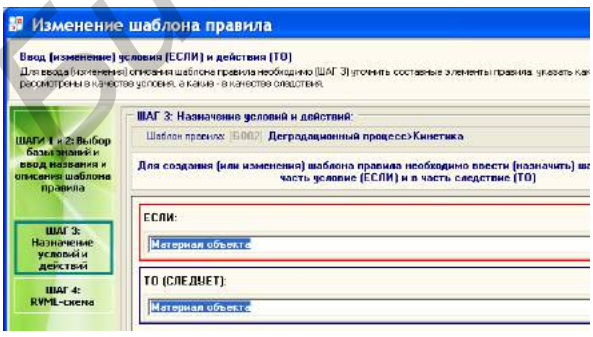

Рисунок 5 – Изменение/Ввод шаблона правила

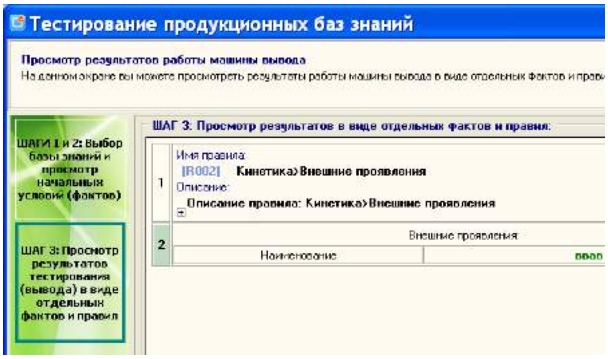

Рисунок 6 – Результаты проверки работоспособности (тестирования)

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

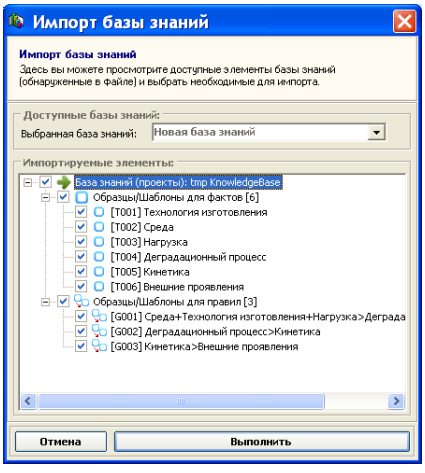

Рисунок 8 – Просмотр импортируемых из mdl-файла (IBM Rational Rose) элементов

Библиотека БГУИР

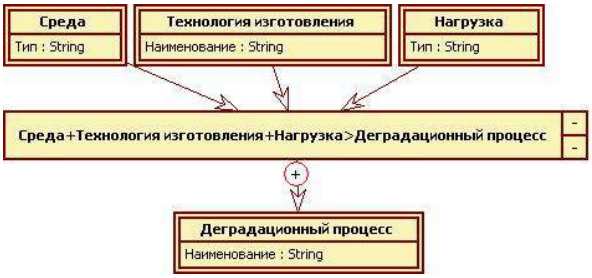

Рисунок 9 – Пример RVML-схемы

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 продукций*, 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 продукций. 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 (ИрНИТУ) 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.

Библиотека БГУИР