

УДК 004.89
DOI: 10.15827/0236-235X.124.667-672

Дата подачи статьи: 03.07.18
2018. Т. 31. № 4. С. 667–672

Средства поддержки моделирования логических правил в нотации RVML

Н.О. Дородных^{1, 2}, к.т.н., младший научный сотрудник, tualatin32@mail.ru

А.Ю. Юрин^{1, 2}, к.т.н., доцент, зав. лабораторией, iskander@icc.ru

С.А. Коршунов³, программист, info@centrasib.ru

¹ Институт динамики систем и теории управления им. В.М. Матросова СО РАН,
г. Иркутск, 664033, Россия

² Иркутский национальный исследовательский технический университет,
г. Иркутск, 664074, Россия

³ ЦентраСиб, г. Иркутск, 664003, Россия

При наличии широкого спектра различных средств и методов представления и обработки знаний продукционная модель до сих пор является одним из наиболее распространенных и востребованных формализмов представления знаний при разработке экспертных систем различного назначения. В свою очередь, эффективное применение языков разработки продукционных баз знаний требует использования и создания специализированного инструментария, обеспечивающего не только разработку баз знаний, но и визуализацию данного процесса. Для решения задачи визуализации перспективным является использование специализированных нотаций, расширяющих существующие, в частности UML. Примером подобной нотации является Rule visual modeling language (RVML).

В работе рассмотрены основные элементы RVML и программных средств, обеспечивающих его поддержку, в частности, веб-ориентированный редактор Web RVML Editor, обеспечивающий не только поддержку визуального моделирования баз знаний продукционного типа в нотации RVML с ориентацией на непрограммирующего специалиста, но и интеграцию с внешними программными системами: Knowledge Base Development System для доступа к модулям-конверторам и Personal Knowledge Base Designer для тестирования создаваемых баз знаний. Приведены описание архитектуры редактора и его функций, а также примеры пользовательского интерфейса.

Другим программным средством, рассмотренным в статье, является Tiny RVML Editor, представляющий собой настольное приложение, функционально дублирующее веб-ориентированный редактор и являвшееся для него прототипом.

Описанные средства в совокупности обеспечивают многоплатформенную поддержку процесса визуального моделирования продукций в RVML, по возможности используя единый подход и принципы организации пользовательского интерфейса.

Ключевые слова: визуальное моделирование, база знаний, правила, представление знаний, RVML, веб-ориентированный редактор, настольное приложение.

В рамках исследований по искусственному интеллекту накоплены большой опыт и широкий спектр различных средств и методов представления и обработки знаний. Несмотря на растущую в последнее время популярность семантических технологий, в частности, онтологий как одного из основных формализмов представления знаний, до сих пор наиболее распространенными и востребованными при разработке экспертных систем (ЭС) остаются языки представления знаний (ЯПЗ), основанные на правилах. Привлекательность данной модели представления знаний обусловлена ее простотой и наглядностью для предметников, высокой модульностью, легкостью внесения дополнений и изменений и прозрачностью механизма логического вывода.

При наличии множества ЯПЗ продукционного типа (CLIPS, JESS, Drools, RuleML, SWRL и др.) и стандартов, например RIF (Rule Interchange Format), остается актуальной задача автоматизации создания баз знаний (БЗ) при помощи специализированного ПО, автоматизирующего этапы получения, концептуализации и формализации предметных знаний. На основе классификации [1] и ее

последующей модификации можно выделить основные группы подходов, реализуемых в ПО, решающем данную задачу.

- **Текстовый**, обеспечивающий прямое манипулирование конструкциями ЯПЗ. Подход ориентирован на программистов, реализуется в форме специализированных редакторов, для некоторых из которых существуют расширения, например, VISUAL JESS [2] для JESS (Java Expert Systems Shell) обеспечивает цветовое выделение синтаксических конструкций и интерактивное определение элементов БЗ.

- **Табличный**, основанный на построении таблиц решений и их трансляции в программные коды на ЯПЗ. При этом используются как стандартный формализм таблиц решений [3], так и его специализации, например ХТТ2 [4]. В частности, ХТТ2 обеспечивает переопределение моделей UML (State Machine Diagrams и Activity Diagrams) для проектирования БЗ.

- **Графический**, обеспечивающий создание визуальных абстракций, соответствующих элементам логических правил, с последующей их трансляцией в программные коды на ЯПЗ. Данный под-

ход является наиболее перспективным, так как позволяет значительно расширить круг разработчиков БЗ ЭС за счет непрограммирующих пользователей, владеющих навыками визуального моделирования.

В свою очередь, в рамках графического подхода можно выделить направления, основанные на использовании семантических графовых структур и специализированных нотаций.

Семантические графовые структуры [5, 6]. Данное направление хорошо представлено системами для построения концепт-карт и онтологий. Однако отсутствие общепринятой содержательной трактовки отношений между концептами при трансляции моделей в логические правила затрудняет широкое использование данного подхода при создании продукционных БЗ и ЭС. Например, в инструментальном средстве VisiRule [7] используется и цветовая дифференциация узлов графических блок-схем (flowchart), в [8] – цветовая дифференциация дуг, а в [9] вводятся ограничения на именованности концептов и отношений.

Специализированные нотации. Обеспечивают моделирование логических и причинно-следственных отношений. При этом некоторые из них, например VIPR [10], содержат нестандартные условные обозначения (артефакты), что не всегда интуитивно понятно разработчику. В связи с этим перспективны нотации, являющиеся расширениями или профилями известных языков, например UML, в частности URML [11] или RVML [12], однако их применение ограничивается отсутствием или значительными функциональными ограничениями программных средств.

Целью данной работы является разработка программного средства для создания продукционных БЗ на основе использования графической нотации RVML с возможностью интеграции с различными системами концептуального моделирования в части импорта и анализа визуальных информационных моделей, а также кодогенерации на ЯПЗ, в частности CLIPS.

Язык моделирования логических правил на основе UML

RVML – язык визуального моделирования правил, предназначенный для моделирования и описания продукций. Он основан на UML и может быть рассмотрен как его профиль расширения, использующий терминологию диаграмм классов [12]. В качестве основных элементов используются понятия «класс» и «ассоциация». Данный язык позволяет абстрагироваться от различных продукционных ЯПЗ при представлении логических правил. При этом уточнение отдельных элементов языка (например, приоритет правила (B), значение *коэффициента уверенности* (KY), значение слота «по умолчанию» и др.) позволяет учесть особенности некоторых продукционных ЯПЗ, например CLIPS или FuzzyCLIPS.

Описание основных элементов RVML может быть представлено в расширенной нотации Бэкуса–Наура:

```

<RVML> = {<Fact>}, {<Template>}, {<Rule>},
{<Relationship>}.
<Rule> = (<Condition>{<Condition>}),
<Core>, (<Action>{<Action>}).
<Condition> = <C_Element>.
<Action> = <A_Element> {<A_Element>}.
<C_Element> = (<C_Operator>,
<Fact>{<Fact>})
|
(<C_Operator>, <C_Element>).
<A_Element> = (<A_Operator>,
(<Fact>{<Fact>})).
<A_Operator> = Add | Delete | Modify | Stop.
<C_Operator> = AND | OR | NOT.
<Fact> = <Name>, <CF>, <F_Slot> {<F_Slot>}.
<F_Slot> = <Name>, <Constraint>, <Value>.
<Core> = <Name>, <CF>, <P>.
<Name> = <String>.
<Value> = <Number> | (<Symbol> {<Symbol>}) |
<Set>.
<Constraint> = > | < | = | >= | <= | <.
<CF> = [0,1]. (the certainty factor)
<P> = [1,100]. (the priority of the rule)
<Template> = <Name>, <T_Slot> {<T_Slot>}.
<T_Slot> = <Name>, <DataType>,
[<Constraint>, <Default_Value>].
<DataType> = String | Symbol | Number.
<Default_Value> = <Value>.
<String> = <Symbol>{<Symbol>}.
<Relationship> = <Element>, <Connection>,
<Element>.
<Connection> = <Kind>, <Name>.
<Kind> = <Association>|<Realization>
<Element> = <Condition> | <Action> |
<Template> | <C_Operator>

```

Таким образом, RVML содержит следующие графические примитивы:

- шаблон факта, используемый для создания фактов и частей правила, таких как условие и действие (рис. 1);
- факт как элемент условия логического правила, используемый для представления фактов, над которыми производятся манипуляции в рабочей памяти; данный примитив позволяет представить не только описание факта (слоты и их значение), но и КУ (рис. 2);
- факт как действие в правиле (рис. 3);
- логическое правило, представленное в виде композиции экземпляров трех примитивов: фактов как условий, фактов как действий и ядра как узлового элемента для связи условий и действий; узловой элемент позволяет представить наименование правила, КУ и приоритет (B) (рис. 4);
- соединение между элементами (рис. 5), представленное как ассоциация, содержащая узел для отображения операции с фактами в рабочей памяти: «+» – добавление, «-» – удаление, «~» – изменение, «!» – остановка логического вывода.

На рисунке 6 представлен пример конструкций RVML: шаблон правила (обобщенное правило).

RVML позволяет использовать отдельные графические примитивы для отображения всех эле-

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">Имя шаблона</th> <th style="text-align: center;">Материал</th> </tr> <tr> <td style="font-size: small;"><имя слота> : <тип> <знак> <значение по умолчанию></td> <td style="font-size: small;">вид : String = "сталь"</td> </tr> </table> <p style="text-align: center;">а) б)</p> <p style="text-align: center;">Рис. 1. Шаблон факта в RVML: а) общее представление; б) пример</p> <p style="text-align: center;">Fig. 1. A RVML fact template: а) a common view, б) an example</p>	Имя шаблона	Материал	<имя слота> : <тип> <знак> <значение по умолчанию>	вид : String = "сталь"							
Имя шаблона	Материал										
<имя слота> : <тип> <знак> <значение по умолчанию>	вид : String = "сталь"										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">Имя факта</th> <th style="text-align: center;">КУ</th> <th style="text-align: center;">Напряжение</th> <th style="text-align: center;">1</th> </tr> <tr> <td style="font-size: small;"><имя слота> <знак> <значение></td> <td></td> <td style="font-size: small;">вид = "растягивающие"</td> <td></td> </tr> </table> <p style="text-align: center;">а) б)</p> <p style="text-align: center;">Рис. 2. Факт в RVML (как условие правила): а) общее представление; б) пример</p> <p style="text-align: center;">Fig. 2. A RVML fact (as an antecedent): а) a common view, б) an example</p>	Имя факта	КУ	Напряжение	1	<имя слота> <знак> <значение>		вид = "растягивающие"				
Имя факта	КУ	Напряжение	1								
<имя слота> <знак> <значение>		вид = "растягивающие"									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">Имя факта</th> <th style="text-align: center;">КУ</th> <th style="text-align: center;">Напряжение</th> <th style="text-align: center;">1</th> </tr> <tr> <td style="font-size: small;"><имя слота> <знак> <значение></td> <td></td> <td style="font-size: small;">вид = "растягивающие"</td> <td></td> </tr> </table> <p style="text-align: center;">а) б)</p> <p style="text-align: center;">Рис. 3. Факт в RVML (как действие правила): а) общее представление; б) пример</p> <p style="text-align: center;">Fig. 3. A RVML fact (as a consequent): а) a common view, б) an example</p>	Имя факта	КУ	Напряжение	1	<имя слота> <знак> <значение>		вид = "растягивающие"				
Имя факта	КУ	Напряжение	1								
<имя слота> <знак> <значение>		вид = "растягивающие"									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center;">Имя правила</th> <th style="text-align: center;">КУ</th> <th style="text-align: center;">Механизм повреждения->Кинетика повреждения</th> <th style="text-align: center;">1</th> <th style="text-align: center;">10</th> </tr> <tr> <td></td> <td style="font-size: small;">В</td> <td></td> <td></td> <td></td> </tr> </table> <p style="text-align: center;">а) б)</p> <p style="text-align: center;">Рис. 4. Узловой элемент правила в RVML: а) общее представление; б) пример</p> <p style="text-align: center;">Fig. 4. A RVML logical rule core: а) a common view, б) an example</p>	Имя правила	КУ	Механизм повреждения->Кинетика повреждения	1	10		В				
Имя правила	КУ	Механизм повреждения->Кинетика повреждения	1	10							
	В										

Программный инструментарий

С целью обеспечения программной поддержки RVML предлагается набор инструментов для различных платформ: веб-ориентированный редактор Web RVML Editor и настольное приложение Tiny RVML Editor.

Web RVML Editor. На сегодняшний день веб-технологии предоставляют разработчикам практически неограниченные возможности для реализации своих идей, а веб-приложения получают все большую популярность. При этом они имеют ряд существенных достоинств по сравнению с настольными (desktop) приложениями, а именно: кросс-платформенность, легкий способ распространения, отсутствие установки и базовой настройки, снижение нагрузки на аппаратную часть компьютера пользователя и т.д. Также в контексте популярности облачных вычислений перспективным является разработка веб-приложения, поддерживающего визуальную разработку БЗ и ЭС. При этом в качестве будущего развития данного веб-средства возможна его реализация как облачного сервиса в соответствии с SaaS-моделью.

Основным назначением разработанного программного средства в форме веб-редактора RVML является поддержка визуального моделирования БЗ производственного типа в нотации RVML с ориентацией на непрограммирующего специалиста (например, эксперта предметной области, аналитика и т.д.). Рассмотрим его основные функции и архитектуру.

Функции. В контексте создания БЗ разработанный веб-редактор предоставляет следующий набор основных функций:

- создание, открытие и сохранение БЗ;
- просмотр элементов БЗ с использованием RVML;

ментов продукции, а не стереотипы или типизированные классы, как в UML, присваивать отдельным фактам субъективные вероятности в виде коэффициентов уверенности, а также наглядно отображать тип выполняемых действий, таких как добавление, изменение, удаление.

Текущая спецификация RVML 1.1 позволяет описывать только простые правила, однако возможно расширение ее выразительных способностей с помощью языка объектных ограничений OCL.



Рис. 5. Соединение между элементами в RVML
Fig. 5. A connection in RVML

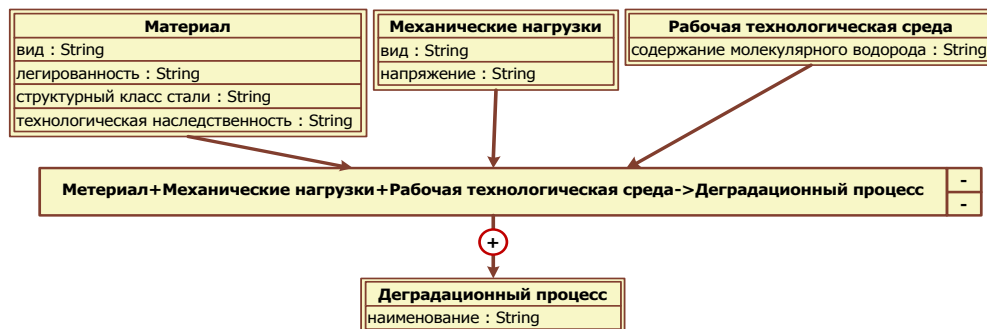


Рис. 6. Примеры конструкций RVML (шаблон правила)
Fig. 6. Construction examples of a logical rule in RVML

- создание, изменение и удаление отдельных элементов БЗ (шаблонов фактов и правил, начальных фактов, конкретных правил);
- генерация кода БЗ на целевом ЯПЗ, в частности CLIPS, включая описание слотов, шаблонов, фактов, правил.

Дополнительной возможностью веб-редактора является интеграция с внешними программными системами, в частности с KBDS и PKBD.

KBDS (Knowledge Base Development System) – веб-ориентированная программная система автоматизированной разработки программных компонентов (конверторов), обеспечивающих генерацию кода БЗ на основе трансформации различных информационных (концептуальных) моделей предметных областей [13]. Благодаря данной системе пользователь получает возможность сформировать прототип (каркас) БЗ на основе концептуальной модели с последующим его уточнением в редакторе RVML.

PKBD (Personal Knowledge Base Designer) – настольный редактор для прототипирования БЗ [14]. Данная система позволяет произвести проверку работоспособности (тестирование) созданной БЗ путем:

- обеспечения логического вывода при помощи подключаемых библиотек машин вывода с целью проверки адекватности созданных БЗ;
- описания (протоколирования) процесса поиска решения (цепочки активированных правил) с целью объяснения полученных результатов;
- формирования отчетов с описанием БЗ (правил, шаблонов и фактов) и результатов логического вывода.

Архитектура. Для реализации данных функций разработана клиент-серверная архитектура веб-редактора RVML, включающая следующие основные модули:

- подсистему визуализации БЗ, обеспечивающую манипулирование графическими примитивами RVML;

- БД для хранения моделей; в структуре БД содержатся только общие конструкции, разделяемые большинством производственных ЯПЗ, что позволяет представить знание в собственном, независимом от конкретного ЯПЗ (CLIPS, JESS, Drools и др.) формате;

- модуль интеграции с внешними приложениями, обеспечивающий поддержку API для взаимодействия с KBDS и PKBD;

- модуль синтеза программных кодов, позволяющий сгенерировать код БЗ в формате CLIPS на основе сформированных RVML-диаграмм;

- модуль генерации отчетов, обеспечивающий распечатку (вывод на принтер) описания БЗ: правил, шаблонов и фактов.

Модель БД. Для поддержки возможности работы со знаниями в виде правил разработана специальная модель продукций, которая содержит только общие конструкции, разделяемые большинством производственных ЯПЗ. Таким образом, разработанная модель продукций позволяет представить знание в собственном, независимом от конкретного ЯПЗ формате, тем самым обеспечивая унифицированное представление и хранение продукционных БЗ. В качестве СУБД для хранения модели продукций использован MySQL, однако, применяя механизм миграций, существует возможность использовать другие СУБД, например PostgreSQL.

Метамодель, определяющая основные концепты, из которых состоит данная модель продукций, представлена на рисунке 7.

Интерфейс. Графический интерфейс веб-редактора RVML (см. <http://www.swsys.ru/uploaded/image/2018-4/2018-4-dop/5.jpg>) можно разделить на три основные области.

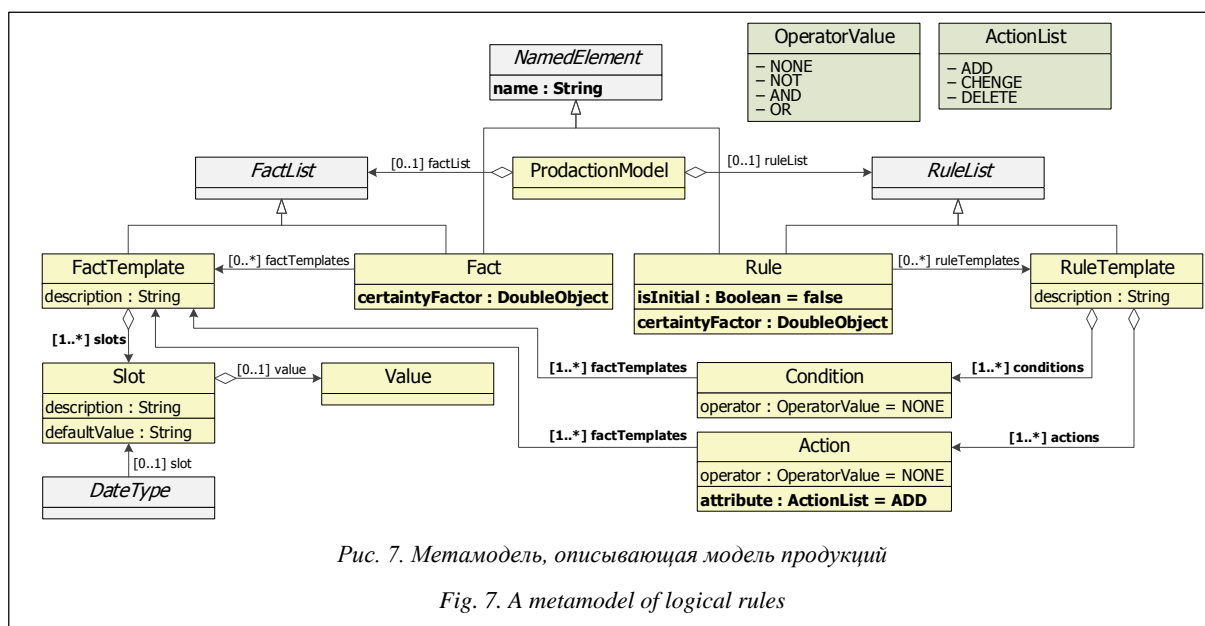


Рис. 7. Метамодель, описывающая модель продукций

Fig. 7. A metamodel of logical rules

Главное меню расположено в верхней части экрана. Оно обеспечивает доступ к проектам БЗ пользователя (создание, открытие, сохранение), к функциям по добавлению новых элементов БЗ (шаблонов фактов и правил, начальных фактов и конкретных правил) и к генерации кода БЗ в формате CLIPS.

Навигационная панель расположена слева и обеспечивает управление элементами БЗ (просмотр, изменение, удаление), поиск и фильтрацию элементов БЗ.

Рабочая область – центральный и основной элемент интерфейса редактора, отображающий выбранные элементы БЗ в нотации RVML. Отображаемые элементы БЗ являются интерактивными (перемещение элементов, поддержка горячих клавиш и т.д.).

Процесс ввода и редактирования элементов БЗ осуществляется при помощи последовательности экранных форм (модальных окон).

Программное средство реализовано с использованием PHP, Yii2 Framework, MySQL, а также библиотек jQuery и jsPlumb.

Tiny RVML Editor представляет собой настольное приложение, которое хронологически разработано раньше веб-редактора RVML и может рассматриваться как подсистема или модуль редактора продукционных БЗ PKBD [14].

PKBD, в свою очередь, обеспечивает возможность функционирования в нескольких режимах: универсального редактора, проблемно-ориентированного редактора [15] и подсистемы визуализации логических правил. При этом по возможности используются единая модель внутреннего представления знаний (рис. 7), а также набор программ-мастеров (см. <http://www.swsys.ru/uploaded/image/2018-4/2018-4-dop/6.jpg>), с помощью которых и производятся ввод и изменение элементов БЗ. Как и в веб-приложении, используется модульный подход при отображении правил, то есть одновременно в рабочей области отображается только одно правило, элементы которого можно детально рассмотреть и изменить (см. <http://www.swsys.ru/uploaded/image/2018-4/2018-4-dop/7.jpg>). Базовая функциональность совпадает с функциональностью веб-редактора RVML.

Благодаря интеграции с PKBD обеспечивается возможность генерации кодов редактируемых элементов на CLIPS, а также проверки работоспособности (тестирования) созданной БЗ.

Заключение

Языки представления знаний, основанные на правилах, по-прежнему остаются востребованными при разработке ЭС, несмотря на популярность семантических технологий, в частности онтологий. Эффективное применение данных языков требует использования и разработки специализированного ПО, позволяющего не только синтезиро-

вать языковые конструкции и создавать программы, но и визуализировать данный процесс.

Для решения задачи визуализации процесса разработки продукционных БЗ существует ряд подходов, основанных на использовании семантических графовых структур и специализированных нотаций. При этом перспективным является использование специализированных нотаций, расширяющих существующие языки, например UML. Одной из подобных нотаций является RVML, обеспечивающий не только моделирование логических правил на основе узнаваемых графических примитивов, но и учет особенностей отдельных элементов ЯПЗ (например, приоритет правила и др.).

В данной работе рассмотрены основные элементы RVML и описаны программные средства, обеспечивающие его поддержку для различных платформ: веб-ориентированного редактора Web RVML Editor и настольного приложения Tiny RVML Editor.

Работа выполнена при частичной поддержке гранта РФФИ, проект № 18-37-00006.

Литература

1. Гаврилова Т.А., Гулякина Н.А. Визуальные методы работы со знаниями: попытка обзора // Искусственный интеллект и принятие решений. 2008. № 1. С. 15–21.
2. Grissa-Touzi A., Ounally H., Boulila A. VISUAL JESS: An expandable visual generator of oriented object expert systems. Intern. J. of Comp. and Inform. Eng., 2007, no. 1, pp. 1668–1671.
3. Башлыков А.А., Еремеев А.П. Основы конструирования интеллектуальных систем поддержки принятия решений в атомной энергетике. М.: Инфра-М, 2018. 351 с. DOI: 10.12737/textbook_590b1950f1cab3.34304392.
4. Nalepa G.J., Kluza K. UML representation for rule-based application models with XTT2-based business rules. Intern. J. of Soft. Eng. and Knowledge Eng., 2012, vol. 4, no. 22, pp. 485–524. DOI: 10.1142/S021819401250012X.
5. IBM WebSphere ILOG JRules. URL: https://www.ibm.com/support/knowledgecenter/ru/SSZJPZ_11.7.0/com.ibm.swg.im.is.com.jrules.use.doc/topics/ilog_jrules_container.html (дата обращения: 17.06.2018).
6. Visual Rules BRM. URL: <https://www.bosch-si.com/bpm-and-brm/visual-rules/business-rules-management.html> (дата обращения: 17.06.2018).
7. VisiRule. Logic Programming Associates. URL: http://www.lpa.co.uk/ind_hom.htm (дата обращения: 17.06.2018).
8. Павлов А.И., Столбов А.Б. Прототип системы поддержки проектирования агентов для имитационных моделей сложных систем // Программные продукты и системы. 2016. № 3. С. 79–84. DOI: 10.15827/0236-235X.115.079-084.
9. Дородных Н.О., Юрин А.Ю. Использование концепт-карт для автоматизированного создания продукционных баз знаний // Программные продукты и системы. 2017. № 4. С. 658–662. DOI: 10.15827/0236-235X.030.4.658-662.
10. Кознов Д.В. Основы визуального моделирования. М.: БИНОМ, 2012. 246 с.
11. Lukichev S., Giurca A., Wagner G., Gasevic D., Ribaric M. Using UML-based rules for web services modeling. Proc. 2nd Intern. Workshop on Services Eng., 2007, pp. 290–297.
12. Юрин А.Ю. Нотация для проектирования баз знаний продукционных экспертных систем // Объектные системы. 2016. № 12. С. 48–54.
13. Юрин А.Ю., Дородных Н.О. Web-сервис для автоматизированного формирования продукционных баз знаний на основе концептуальных моделей // Программные продукты и си-

стемы. 2014. № 4. С. 103–107. DOI: 10.15827/0236-235X.108.103-107.

14. Дородных Н.О., Грищенко М.А., Юрин А.Ю. Система программирования продукционных баз знаний: Personal Knowledge Base Designer // Открытые семантические технологии про-

ектирования интеллектуальных систем. 2016. № 6. С. 209–212.

15. Берман А.Ф., Грищенко М.А., Николайчук О.А., Юрин А.Ю. Проблемно-ориентированный редактор продукционных баз знаний // Программные продукты и системы. 2015. № 2. С. 13–19. DOI: 10.15827/0236-235X.110.013-019.

Software & Systems

DOI: 10.15827/0236-235X.124.667-672

Received 03.07.18

2018, vol. 31, no. 4, pp. 667–672

Support tools for modeling logical rules in the RVML notation

N.O. Dorodnykh^{1,2}, Ph.D. (Engineering), Junior Researcher, tualatin32@mail.ru

A.Yu. Yurin^{1,2}, Ph.D. (Engineering), Associate Professor, Head of Laboratory, iskander@icc.ru

S.A. Korshunov³, Programmer, info@centrasib.ru

¹Matrosov Institute of System Dynamics and Control Theory SB RAS, Irkutsk, 664033, Russian Federation

²Irkutsk National Research State Technical University, Irkutsk, 664074, Russian Federation

³CentraSib LLC., 11, Karl Marks Str., Irkutsk, Russia

Abstract. Nowadays there is a wide range of different means and methods of knowledge representation and processing. At the same time, a rule-based model is still one of the most common and popular formalisms of knowledge representation for the development of expert systems for various purposes. In turn, the effective application of languages for rule-based knowledge base engineering requires the use and creation of specialized toolkits that provide the visualization of this process. The use of specialized notations, which extend existing ones (UML in particular) is promising in the field of visualization of rule base engineering. The Rule Visual Modeling Language (RVML) is an example of such UML extension.

The paper describes the main elements of RVML and software that supports it. In particular, it considers the webcentric Web RVML Editor. This editor supports RVML and is designed for end-users who are non-programmers. It also provides integration with external specialized software that is: the Knowledge Base Development System to access converting modules and Personal Knowledge Base Designer to test knowledge bases. The paper describes the Web RVML Editor architecture, its main functions and GUI.

The paper also presents the Tiny RVML Editor that is desktop software. It is a prototype of the Web RVML Editor and it duplicates it functionally. The described software provide multi-platform support for visual engineering of logical rules in RVML notation and use the common approach and principles of GUI organization.

Keywords: visual modeling, knowledge base, rules, knowledge representation, RVML, web-based software, desktop software.

Acknowledgements. The study has been partially supported by RFBR, research project no. 18-37-00006.

References

- Gavrilova T.A., Gulyakina N.A. Visual knowledge processing techniques: a brief review. *Artificial Intelligence and Decision Making*. 2008, no. 1, pp. 15–21 (in Russ.).
- Grissa-Touzi A., Ounally H., Boulila A. VISUAL JESS: An expandable visual generator of oriented object expert systems. *Intern. J. of Computer and Information Engineering*. 2007, vol. 1, no. 11, pp. 1668–1671.
- Bashlykov A.A., Ereemeev A.P. Fundamentals of designing intelligent decision support systems in nuclear power engineering. Moscow, INFRA-M Publ., 2018, 351 p. DOI: 10.12737/textbook_590b1950f1cab3.34304392.
- Nalepa G.J., Kluza K. UML representation for rule-based application models with XTT2-based business rules. *Intern. J. of Software Engineering and Knowledge Engineering*. 2012, vol. 22, no. 4, pp. 485–524. DOI: 10.1142/S021819401250012X.
- IBM WebSphere ILOG JRules. Available at: https://www.ibm.com/support/knowledgecenter/ru/SSZJPZ_11.7.0/com.ibm.swg.im.iis.conn.jrules.use.doc/topics/ilog_jrules_container.html (accessed June 17, 2018).
- Visual Rules BRM. Available at: <https://www.bosch-si.com/bpm-and-brm/visual-rules/business-rules-management.html> (accessed June 17, 2018).
- VisiRule. Logic Programming Associates. Available at: http://www.lpa.co.uk/ind_hom.htm (accessed June 17, 2018).
- Pavlov A.I., Stolbov A.B. A prototype of an agents design support system for complex system simulation models. *Software & Systems*. 2016, no. 3, pp. 79–84 (in Russ.). DOI: 10.15827/0236-235X.115.079-084.
- Dorodnykh N.O., Yurin A.Yu. Using concept maps for rule-based knowledge bases engineering. *Software & Systems*. 2017, no. 4, pp. 658–662 (in Russ.). DOI: 10.15827/0236-235X.030.4.658-662.
- Koznov D.V. *Fundamentals of Visual Modeling*. Moscow, BINOM Publ., 2012, 246 p.
- Lukichev S., Giurca A., Wagner G., Gasevic D., Ribaric M. Using UML-based rules for web services modeling. *Proc. 2nd Intern. Workshop on Services Engineering*. 2007, pp. 290–297.
- Yurin A.Yu. Notation for design of knowledge bases of rule-based expert systems. *Object Systems*. 2016, no. 12, pp. 48–54 (in Russ.).
- Yurin A.Yu., Dorodnykh N.O. A Web-service for knowledge base generation based on conceptual models. *Software & Systems*. 2014, no. 4, pp. 103–107 (in Russ.). DOI: 10.15827/0236-235X.108.103-107.
- Dorodnykh N.O., Grischenko M.A., Yurin A.Yu. Software for rule knowledge bases design: Personal Knowledge Base Designer. *Open Semantic Technologies for Intelligent System Design*. 2016, no. 6, pp. 209–212.
- Berman A.F., Grishchenko M.A., Nikolaychuk O.A., Yurin A.Yu. A problem-oriented editor to design of rule-based knowledge bases. *Software & Systems*. 2015, no. 2, pp. 13–19 (in Russ.). DOI: 10.15827/0236-235X.110.013-019.