

Germany, March 26 –April 3, 2011 // Lecture Notes in Computer Science. Springer. V2011. V. 6601. 330 p.

5. *Городня Л.В.* О проблеме автоматизации параллельного программирования // В сборнике Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: многообразие суперкомпьютерных миров». URL: <http://agora.gugu.ru/abrau2014>.

6. *Городня Л.В.* Парадигма программирования : курс лекций / Л.В. Городня ; Новосиб. гос. Ун-т. – Новосибирск: РИЦ НГУ, 2015. 206 с.

7. *Городня Л.В.* Парадигмы параллельного программирования в университетских образовательных программах и специализации // Всероссийская научная конференция «Научный сервис в сети Интернет: решение больших задач». – Новороссийск-Москва, 2008. с. 180-184.

8. *Хорстман К.* Scala для нетерпеливых. – ДМК пресс, 2013. 408 с.

9. *Peter Van Roy.* The principal programming paradigms (2008). – Available at: <https://www.info.ucl.ac.be/~pvr/paradigmsDIAGRAMeng108.pdf>.

10. Бек К. Экстремальное программирование – Питер, 2003. 224 с.

11. *Палмер С.Р., Фелсинг Дж.М.* Практическое руководство по функционально ориентированной разработке ПО. – М.: Вильямс, 2002. 299 с.

Classification of programming and parallel programming paradigms

Lidia Vasiljevna Gorodnyaya, Ph.D., Docent, Senior Researcher

The article concerns the actual problem of study and development of methods of analysis, comparison and formal definition of the programming paradigms. Importance of this topic comes from the steep increase in the number of new-generation programming languages oriented at application and development of modern information technologies.

Keywords – programming languages and systems; programming paradigms; computer languages definition methods; parallel programming; educational programming.

УДК 004.4'22:004.891

ИНСТРУМЕНТАЛЬНОЕ СРЕДСТВО СОЗДАНИЯ ПРОДУКЦИОННЫХ ЭКСПЕРТНЫХ СИСТЕМ НА ОСНОВЕ MDA

Максим Андреевич Грищенко, программист

Тел.: 8 395 245 31 57, e-mail: maksmg@icc.ru

Институт динамики систем и теории управления им. А.М. Матросова СО РАН

<http://www.icc.irk.ru>

Ольга Анатольевна Николайчук, д-р техн. наук, с.н.с.

Тел.: 8 395 245 31 57, e-mail: nikoly@icc.ru

Институт динамики систем и теории управления им. А.М. Матросова СО РАН

<http://www.icc.irk.ru>

Александр Иннокентьевич Павлов, канд. техн. наук, с.н.с.

Тел.: 8 395 245 30 19, e-mail: asd@icc.ru

Институт динамики систем и теории управления им. А.М. Матросова СО РАН

<http://www.icc.irk.ru>

Александр Юрьевич Юрин, канд. техн. наук, зав.лаб.

Тел.: 8 395 245 30 19, e-mail: iskander@icc.ru

Институт динамики систем и теории управления им. А.М. Матросова СО РАН

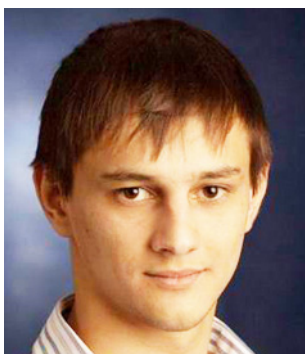
<http://www.icc.irk.ru>

В работе описываются архитектура и функции инструментального средства создания экспертных систем на основе модельно-управляемого подхода, в частности методологии MDA (Model Driven Architecture). Рассмотрены основные компоненты и этапы технологии, приведены примеры результатов моделирования.

Ключевые слова: Экспертные системы, Model Driven Architecture, онтология, инструментальное средство, автоматизация научных исследований

Работа выполнена при частичной поддержке грантов РФФИ (проекты № 15-07-03088, 15-07-05641, 16-37-00041).

Информатизация общества создает предпосылки для обеспечения возможности использования большого объема разнородной по степени формализации информации для решения



М.А. Грищенко

любых предметных задач. С течением времени задачи меняются, имеющуюся информацию необходимо анализировать с разных точек зрения, желательно в реальном масштабе времени, комбинируя средства анализа. Проблемы возникают из-за отсутствия у предметных специалистов достаточной подготовки в области информационных технологий, в частности, технологий интеллектуальных информационных систем. Привлечение дополнительных специалистов в подобной ситуации удлинняет



О.А. Николайчук

процесс решения задачи за счет необходимости изучения ими задачи и соответствующей ей информации, а также наличия эффекта «испорченного телефона» при интерпретации знаний эксперта [1] и т.д. Решением проблемы является создание инструментальных средств с интуитивно понятным интерфейсом, позволяющих специалистам-предметникам (экспертам) самостоятельно обрабатывать информацию. Например, к таким средствам можно отнести OLAP-технологии и Data Mining. Если говорить об экспертных системах (ЭС), то необходимо отметить их особенность, что, с одной стороны,



А.И. Павлов

ЭС интуитивно понятны специалистам-предметникам по способу представления информации (знаний), с другой стороны, инструменты для их создания, присутствующие на рынке и являющиеся доступными, не обладают «дружественным» интерфейсом, не способны «легко» интегрироваться с СУБД, математическими пакетами, системами имитационного моделирования и т.д., что затрудняет их широкое использование (ClipsWin, ExSys, Drools Expert).



А.Ю. Юрин

Ранее авторами предложена концепция системы автоматизации исследований на основе компонентного подхода [2], которая затем была дополнена возможностью использования сервисно-ориентированной технологии разработки программного обеспечения в процессе реализации отдельных компонентов системы. Основным ее преимуществом является обеспечение единого информационного пространства исследования на основе онтологического моделирования и возможности расширения функциональности системы на основе вновь подключаемых/интегрируемых компонентов/сервисов. Другое декларируемое свойство системы – ориентированность на непрограммирующего пользователя, т.е. специалиста-предметника, что обусловило использование технологий визуального моделирования предметной области.

Также для решения указанных проблем при создании экспертных систем авторы предложили в рамках системы автоматизации исследований применить MDA-подход¹

¹ OMG Model Driven Architecture. URL: <http://www.omg.org/mda/>

[3] и специфицировать его, т.е. определить состав моделей и правил трансформации:

$$MDA^{ES} = \left\langle MOF, UML^{ES}, CWM, CIM^{ES}, PIM^{ES}, PSM^{ES}, PDM^{ES}, F_{CIM-to-PIM}^{ES}, F_{PIM-to-PSM}^{ES}, F_{PSM-to-CODE}^{ES} \right\rangle$$

где MDA^{ES} – система для создания баз знаний и экспертных систем, основанная на методологии MDA ; MOF (Meta Object Facility) – абстрактный язык для описания информации о моделях (язык описания метамodelей, основан на UML); UML^{ES} (Unified Modelling Language) – унифицированный язык моделирования (используется для построения визуальных моделей согласно MDA [OMG]), ориентированный на моделирование знаний; CWM (Common Warehouse Metamodel) – метамodelь хранения данных; CIM^{ES} (Computation Independent Model) – вычислительно-независимая модель ЭС – модель, скрывающая любые детали реализации и процессов системы, которая описывает только требования к системе и ее окружению, реализована в виде онтологии, описывающей основные понятия и отношения предметной области, а также архитектурные элементы ЭС: формы ввода/вывода, машина вывода; PIM^{ES} (Platform Independent Model) – платформу-независимая модель ЭС – модель, скрывающая детали реализации системы, зависимые от платформы, и содержащая элементы, не изменяющиеся при взаимодействии системы с любой платформой, представлена описанием продукций; PSM^{ES} (Platform Specific Model) – платформу-зависимая модель ЭС – модель системы с учетом деталей реализации и процессов, зависимых от конкретной платформы, представлена в виде продукций на языке представления знаний и элементов пользовательского интерфейса на языках программирования HTML, JavaScript, PHP; PDM^{ES} (Platform Description Model) – модель платформы ЭС – набор технических характеристик и описаний технологий и интерфейсов, составляющих программную (технологическую) платформу;

$$F_{CIM-to-PIM}^{ES} : CIM^{ES} \rightarrow PIM^{ES}, F_{PIM-to-PSM}^{ES} : PIM^{ES} \rightarrow PSM^{ES},$$

$$F_{PSM-to-CODE}^{ES} : PSM^{ES} \rightarrow CODE \text{ – правила трансформации моделей.}$$

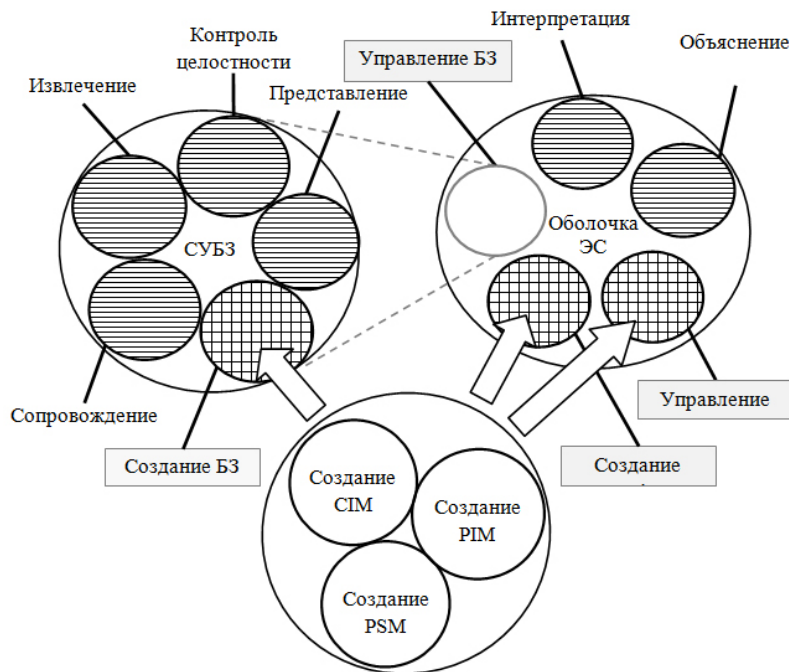


Рис.1. Функции оболочки ЭС и СУБЗ, как область применения предлагаемого метода

Применение MDA позволяет решить проблемы, связанные с постоянно увеличивающимся количеством технологических платформ, а также может ускорить разработку и интеграцию систем [4; 5]. В процессе разработки ЭС, технологическими платформами могут выступать CLIPS и JESS, ориентированные на desktop и web разработку соответственно, при этом ускорение разработки обусловлено автоматизацией этапа создания программного кода базы знаний ЭС и возможностью повторного использования моделей предметной области, что также обеспечивает возможность интеграции с системами, ориентированными на обработку аналитических моделей.

Для уточнения области применения предлагаемого подхода в классической технологии создания ЭС [1] опишем функции оболочки экспертной системы и входящую в нее систему управления базами знаний (СУБЗ) в виде блочно-функциональной структуры (рис. 1). Предлагаемые специфицированные этапы MDA охватывают самые сложные этапы – это этапы создания баз знаний (БЗ) и ЭС, частично решают задачи управления ими.

Целью данной работы является описание архитектуры и функциональности инструментального средства, реализующего предложенный подход.

Архитектура и функции инструментального средства

С учетом требований к функциональности инструментального средства была разработана его web-ориентированная архитектура. Опишем основные модули и их назначение (рис. 2).

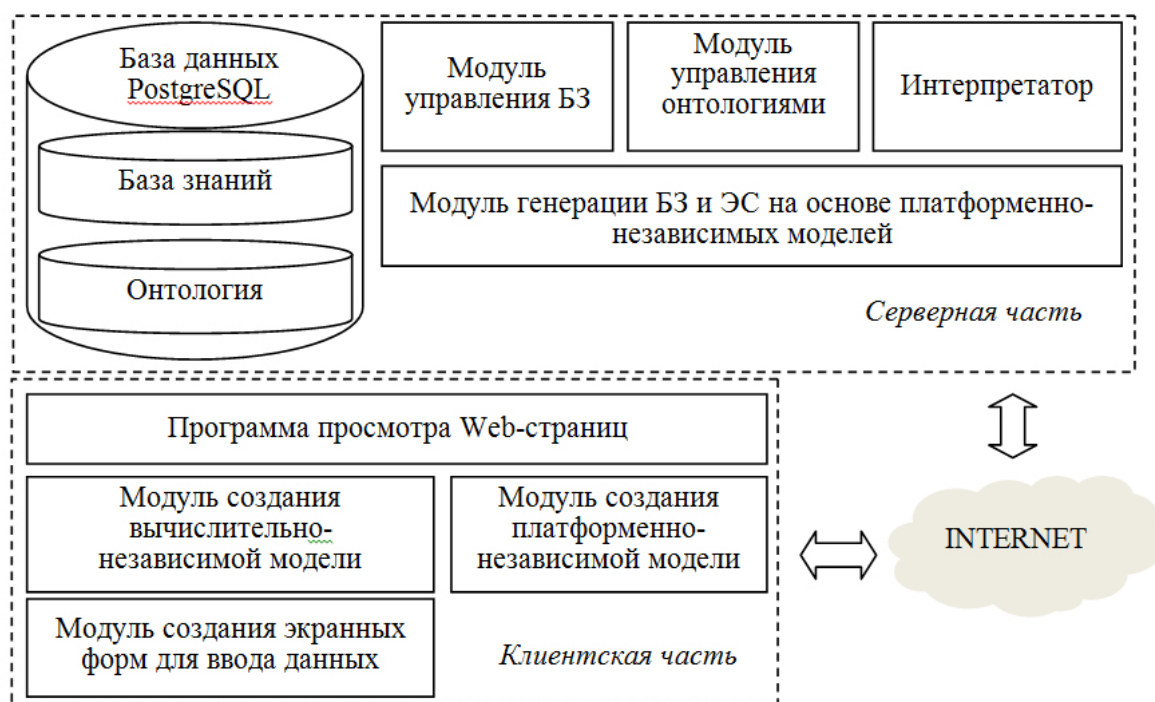


Рис. 2. Клиент-серверная архитектура программного средства

Серверная часть:

- Модуль управления базами знаний, реализующий загрузку и сохранение баз знаний; слияние и разделение существующих баз знаний.
- Модуль управления базами данных, обеспечивает взаимодействие системы с базой данных, загрузку и сохранение баз данных.
- Модуль управления онтологиями (вычислительно-независимыми моделями), позволяющий интегрировать понятия предметной области в базу знаний, на их основе создавая продукции.
- Модуль генерации БЗ и БД на основе платформо-зависимых моделей позволяет преобразовать продукции базы знаний в выбранный язык представления знаний

(например, CLIPS, JESS) и преобразовать логические модели БД в базу данных PostgreSQL. Генерация кода может быть осуществлена как для базы знаний в целом, так и для отдельных ее элементов (шаблон факта, правило, описание начальных условий). Для тестирования кода базы знаний используется специальный терминал, обеспечивающий возможность редактирования сгенерированного кода и его выполнения с помощью интерпретатора JESS. Терминал обеспечивает отображение результатов логического вывода: состояние рабочей памяти, список сработавших правил, сообщения об ошибках.

Клиентская часть:

Модуль создания вычислительно-независимых моделей, позволяющий непрограммирующим пользователям создавать и редактировать модель предметной области (онтологию) в графическом виде. Графическое отображение онтологии в виде семантической сети обеспечивает визуальное представление знаний в форме, наиболее соответствующей современным представлениям об организации долговременной памяти человека. Модуль обладает следующей функциональностью: создание моделей исследуемых областей, создание перечня возможных свойств понятий, формирование иерархической структуры понятий от абстрактных к более конкретным, определения свойств понятий и отношений между ними (рис. 3).

+	Деталь	
-	Дефект	Состояние объекта обусловленное изготовлением, монтажом или ремонтом.
Свойства понятия		
+	Вид дефекта	Возможные виды дефектов: закаты; неметаллические включения; поры; расслоение; трещина (ы) – дефекты; флокены.
+	Местоположение	Возможные значения: зона термального влияния, на поверхности, на внешней поверхности, на внутренней поверхности, близлежащие зоны термического влияния, на поверхности, в пределах одной зоны термического влияния.
+	Причины возникновения	
Подпонятия		
+	Включения неметаллические	Инородные неметаллические вещества в основном материале. Могут быть допустимыми по величине. Причина: несовершенство и (или) нарушение технологии плавления и (или) разливки металла.
+	Волосовины	Нарушение сплошности металла в виде вытянутых вдоль направления деформации закатов с прослойкой включений. Причина: проявление невыявленных закатов и включений.

Рис. 3. Фрагмент онтологии в виде иерархии понятий

Результат применения модуля для описания модели состояний деградационного процесса представлен рис. 4. Системы с подобной функциональностью (т.е. с возможностью создания концептуальных моделей) существуют, например, редактор Protégé, но они не ориентированы на коллективную работу исследователей, которая может быть эффективно реализована в случае использования СУБД для хранения информации, а не XML-подобных файлов.

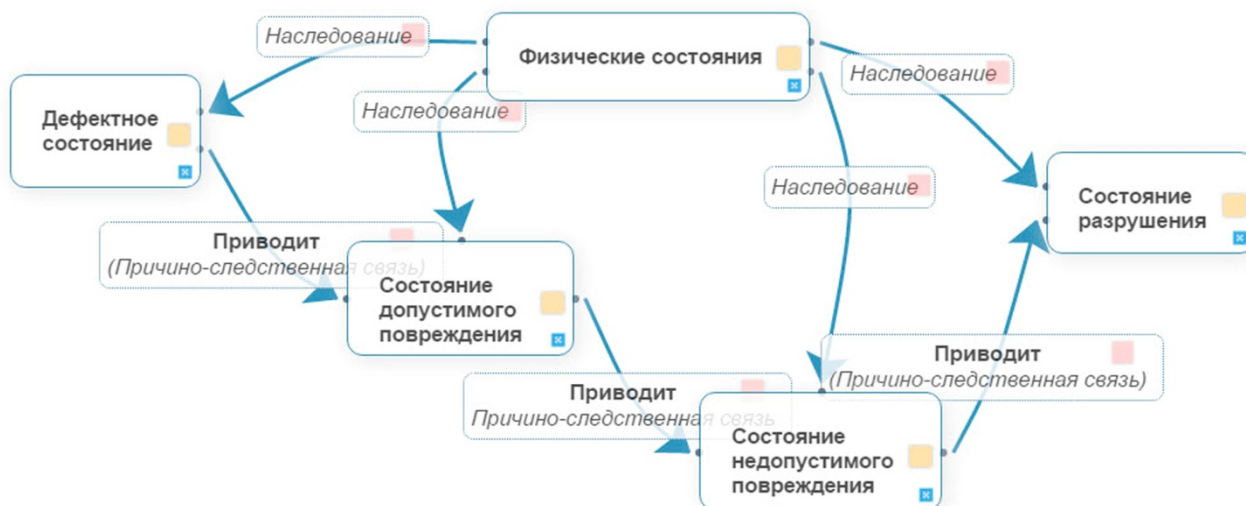


Рис. 4. Пример онтологии в виде семантической сети «Развитие деградационного процесса»

Модуль создания платформо-независимых моделей, позволяющий создавать непрограммирующему пользователю платформо-независимые модели (продукционные базы знаний) в графическом виде, а также редактировать базы знаний продукционного типа. Предлагаются две авторские нотации графического языка: UML-ориентированное (RVML – Rule Visual Modeling Language) и «интуитивно-понятное» сетевое представление (рис.5) правил.

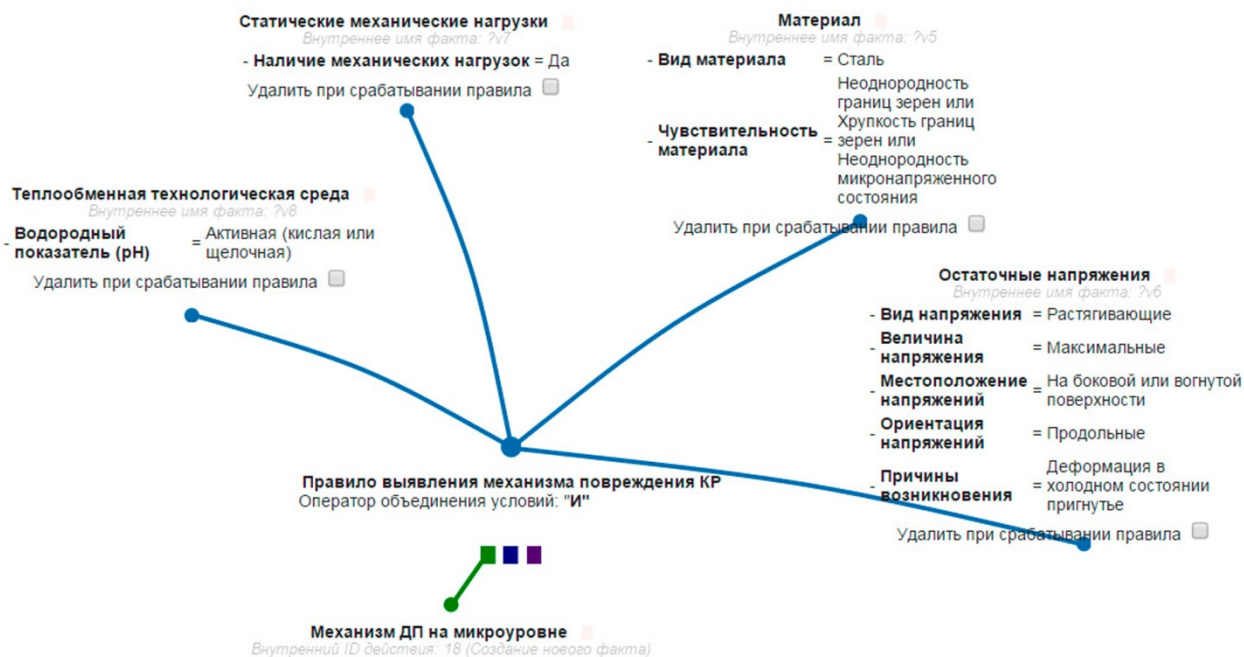


Рис. 5. Пример конструирования правила

Модуль обладает следующей функциональностью: создание баз знаний для выбранной концептуальной модели, формирование перечня шаблонов фактов на основе существующих понятий концептуальной модели, визуальное конструирование правил вида «ЕСЛИ условия ТО действия»; ввод начальных условий для запуска процедуры логического вывода; генерацию и тестирование кода базы знаний на языках JESS, CLIPS, COOL. В настоящее время при конструировании правил можно использовать условия следующего вида: наличие факта указанного шаблона, ограничение на значение слота факта (текст, число), сравнение значения слота с существующим фактом, сравнение со значением слота другого факта. Для формирования правой части правила используются следующие виды действий: создание нового факта,

модификация существующего факта, удаление факта, вызов внешней процедуры.

База данных реализует универсальную модель для хранения понятий предметной области и их свойств, эти понятия могут также служить шаблонами для продукций. Предложенная модель включает понятия: база знаний, правила, действия правил, постулаты правил, условия правил, предусловия правил, набор условий, шаблоны фактов, новые факты, новые факты на основании шаблонов, параметры новых фактов, атрибуты фактов, типы атрибутов фактов, возможные значения атрибутов, определения атрибутов фактов, условия на значения фактов.

Технологический процесс, реализующий предложенный подход представлен на рис. 6.

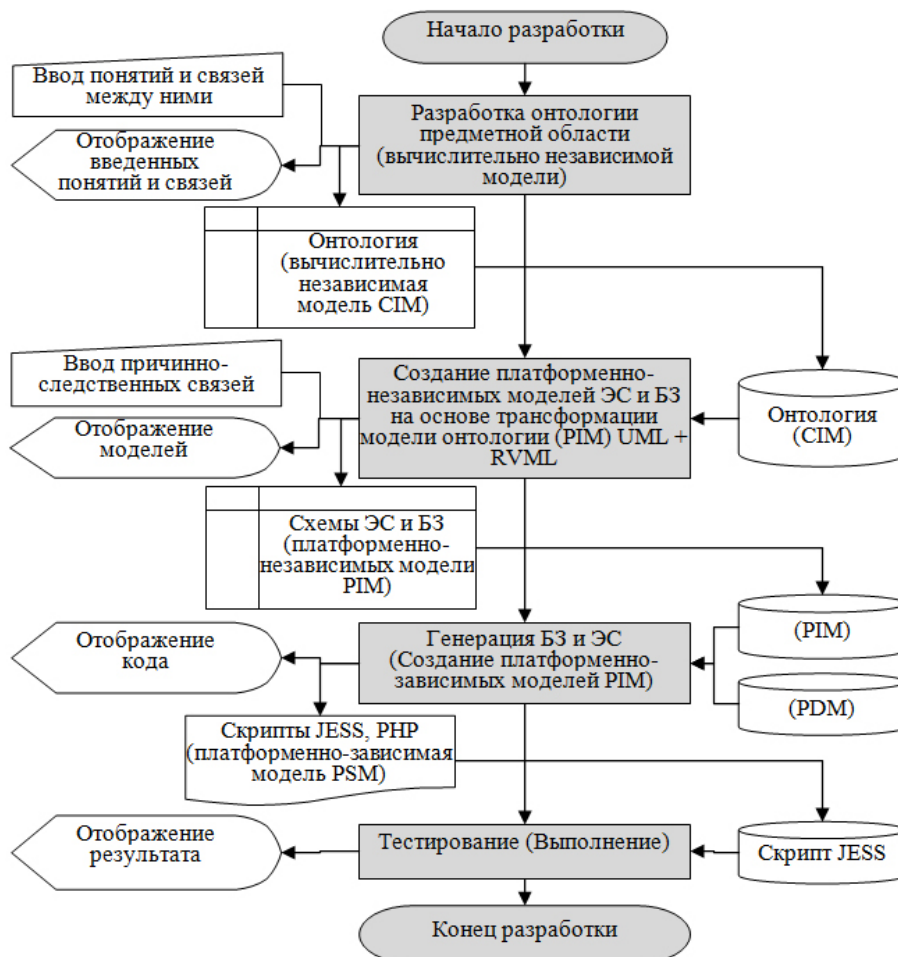


Рис. 6. Технологическая схема создания экспертной системы на основе MDA

Заключение

Авторы считают, что в данной работе новыми являются следующие результаты: предложена реализация метода создания продукционных экспертных систем и баз знаний, заключающийся в перенесении основных принципов MDA на область инженерии знаний и отличающийся от стандартного MDA использованием онтологии для представления вычислительно-независимой модели и новых графических нотаций для построения моделей продукционных баз знаний. Предложенный подход апробирован при создании фрагментов экспертных систем для моделирования деградационных процессов и поддержки принятия решений по ликвидации разливов нефти.

Литература

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний и интеллектуальные системы. – СПб.: Питер, 2000.
2. Николайчук О.А., Павлов А.И. Применение компонентного подхода для создания системы автоматизации исследований // Вестник компьютерных и информационных технологий.

2010. №4. С.23-32.

3. Грищенко М.А., Николайчук О.А., Юрин А.Ю. Метод создания продукционных экспертных систем на основе модельно-управляемого подхода // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2014), 24-27 сентября 2014 г., г. Казань, Россия: Труды конференции. Т.3. – Казань: РИЦ «Школа», 2014. С.100-108.

4. Кузнецов М.Б. Трансформация UML-моделей и ее использование в технологии MDA // Программирование. 2007. Вып. 33. С. 44-53.

5. De Miguel M., Jourdan J., Salicki S. Practical experiences in the application of MDA // LNCS. 2002. Vol. 2460. P. 128-139.

Software tool for development of rule-based expert systems on the basis of MDA

Maksim Andreevich Grishenko, programmer

Olga Anatolievna Nikolaychuk, Dr.Sc. Senior Reseacher

Aleksandr Innokentievich Pavlov, Ph.D. Senior Reseacher

Aleksandr Yurievich Yurin, Ph.D. Head of Laboratory

The paper describes the architecture and functions of the software tool for development of rule-based expert systems on the basis of a model-driven approach, in particular the MDA (Model Driven Architecture) methodology. The main components and stages of the technology are considered. The results of modeling and model transformations are presented.

Keywords – Rule-based expert systems, Model Driven Architecture, ontology, software tool, automation of scientific researches.

УДК 630.43

ЗАДАЧИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ПРОЦЕССОМ ЛИКВИДАЦИИ ПРИРОДНОГО ПОЖАРА

Георгий Алексеевич Доррер, заведующий кафедрой системотехники

Тел.: 89135341066, e-mail: g_a_dorrer@mail.ru

Сибирский государственный технологический университет

<http://www.sibgtu.ru>

Предложена простая модель динамики природного пожара, позволяющая решать ряд задач синтеза систем управления этим процессом.

Ключевые слова: Природный пожар, управление, оптимальный состав противопожарных сил и средств.

Введение

Задача оптимального использования сил и средств для ликвидации природных пожаров рассматривалась в работах G.M. Parks [1] и M. Parlar, R.G. Vicson [2]. На основе принципа максимума Понтрягина были получены выражения для оптимального закона управления противопожарными силами при постоянной скорости фронта пожара. При этом управление осуществляется в разомкнутом режиме, без учета фактического состояния системы. Не учитываются также возможные помехи и неполнота информации об объекте управления.

Таким образом, существует проблема разработки методов оптимального управления процессом борьбы с природными пожарами на основе со-



Г.А. Доррер