

## КОНЦЕПЦИЯ ПРОГРАММНОЙ СИСТЕМЫ СОЗДАНИЯ ВЕБ-СЕРВИСОВ СИНТЕЗА БАЗ ЗНАНИЙ НА ОСНОВЕ КОНЦЕПТУАЛЬНЫХ МОДЕЛЕЙ<sup>1</sup>

*Н.О. Дородных, А.Ю. Юрин, С.А. Коришонов*

*Институт динамики систем и теории управления имени В.М. Матросова СО РАН,  
Иркутск*

*e-mail: [tualatin32@mail.ru](mailto:tualatin32@mail.ru), [iskander@icc.ru](mailto:iskander@icc.ru), [grey.for@gmail.com](mailto:grey.for@gmail.com)*

*Предложена концепция сервис-ориентированной системы для создания программных модулей (веб-сервисов) синтеза баз знаний на основе автоматизированного анализа концептуальных моделей. Рассмотрена концептуальная трехуровневая архитектура данной системы. Описан общий алгоритм метода трансформации концептуальных моделей, представленных в формате XML, в код баз знаний, на основе визуального построения соответствий (правил преобразования) исходных и целевых элементов метамodelей. Особенностью подхода является автоматическое выделение элементов метамodelей для визуализации построения правил трансформации, а также использование специальной нотации RVML для уточнения полученных продукций.*

*Ключевые слова: технология создания программных систем, сервис, трансформация моделей, база знаний, приобретение знаний, концептуальная модель, онтология*

### **Введение**

В настоящее время разработка новых методов и подходов к созданию интеллектуальных систем (ИС) основанных на знаниях остается перспективной областью научных исследований. Основную сложность при разработке подобных систем представляет этап формирования баз знаний (БЗ), который включает задачи по получению, структурированию и формализации (представлению) знаний [1, 2, 3].

Эффективность данного процесса может быть повышена путем автоматизации процессов повторного использования (преобразования) концептуальных моделей предметных областей, построенных с использованием различных программных средств (например, CASE-средств, систем концептуального, когнитивного или онтологического моделирования). Однако, данные программные решения имеют ряд существенных недостатков, в частности, они практически не поддерживают возможность генерации кода на различных языках программирования баз знаний (ЯПБЗ) и не обеспечивают совместную, распределенную работу пользователей в сети Интернет.

Таким образом, с целью преодоления выделенных недостатков, в данной работе предлагается повысить эффективность процесса разработки БЗ, путем создания сервис-ориентированной системы (платформы) разработки веб-сервисов (программных модулей) для синтеза БЗ на основе автоматизированного анализа концептуальных моделей, выполненных в различных программных средствах.

Для достижения поставленной цели осуществлена разработка: общей концепции сервис-ориентированной системы и ее основных компонентов; метода трансформации исходных концептуальных моделей в код на целевом ЯПБЗ, основанного на процедуре

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проект №15-37-20655, №15-07-03088)

установления соответствий элементов метамоделей. Разработанные при помощи данной платформы веб-сервисы должны обеспечивать преобразование различных концептуальных моделей в расширенную онтологию (внутреннее представление знаний в системе), с последующим моделированием (уточнением и дополнением) полученных знаний с использованием нотации RVML (Rule Visual Modeling Language) [4] и их отображение (трансляцию) на целевой ЯПБЗ. Применение веб-технологий, при создании такой платформы, обеспечивает возможность совместной и распределенной работы пользователей (экспертов, инженеров по знаниям и системных аналитиков) над процессом формирования БЗ.

Данная работа является логическим продолжением предыдущих исследований авторов в области автоматизации разработки БЗ на основе концептуальных моделей [5].

## **1. Концепция программной системы**

Назначением разрабатываемой сервис-ориентированной системы (платформы) является создание программных модулей (веб-сервисов) для поддержки процесса автоматизированного формирования БЗ на определенном ЯПБЗ, путем автоматической трансформации различных концептуальных моделей.

Основные принципы разрабатываемой программной системы:

1) *Обеспечение удаленного доступа* через Интернет к функциям сервис-ориентированной системы без необходимости размещения дополнительного программного обеспечения на компьютерах пользователей, т.е. предоставление пользователям веб-сервисов вместо предоставления им непосредственно некоторых версий программных средств.

2) *Организация единого пространства* (среды) для разработки и функционирования различных веб-сервисов синтеза БЗ, при этом предполагается наличие веб-сервисов двух типов, взаимодействующих через унифицированные интерфейсы:

- *стандартные системные сервисы* – предлагаются разработчиками платформы «по умолчанию»;
- *пользовательские прикладные сервисы* – создаются пользователями при помощи инструментального сервиса разработки.

3) *Поддержка прав доступа* к функциональным возможностям (ресурсам) сервисов платформы, при помощи единого сервиса системного администрирования, который, прежде всего, подразумевает предотвращение несанкционированного доступа к веб-сервисам синтеза БЗ.

4) *Обеспечение совместной и распределенной деятельности пользователей* в процессе формирования БЗ.

5) *Обеспечение накопления и развития интеллектуальных сервисов* формирования БЗ на основе различных концептуальных моделей.

### **1.1 Концептуальная архитектура системы**

На рисунке 1 представлена концептуальная архитектура сервис-ориентированной системы, которая состоит из трех основных компонентов:

- *Информационные ресурсы* – компонент предназначен для хранения множества различной информации (данных и знаний), которая используется системными и

прикладными сервисами платформы, как для обеспечения собственного функционирования, так и для решения определенных пользовательских задач;

- *Системные сервисы* – компонент представляет собой набор всех предлагаемых пользователям сервисов платформы, обеспечивающих базовое взаимодействие пользователей с системой и предоставляющие инструментарий для создания прикладных веб-сервисов;

- *Прикладные (пользовательские) сервисы* – компонент представляет собой набор всех разработанных веб-сервисов пользователями, обеспечивающих возможность автоматического формирования БЗ на различных ЯПБЗ, путем трансформации концептуальных моделей.

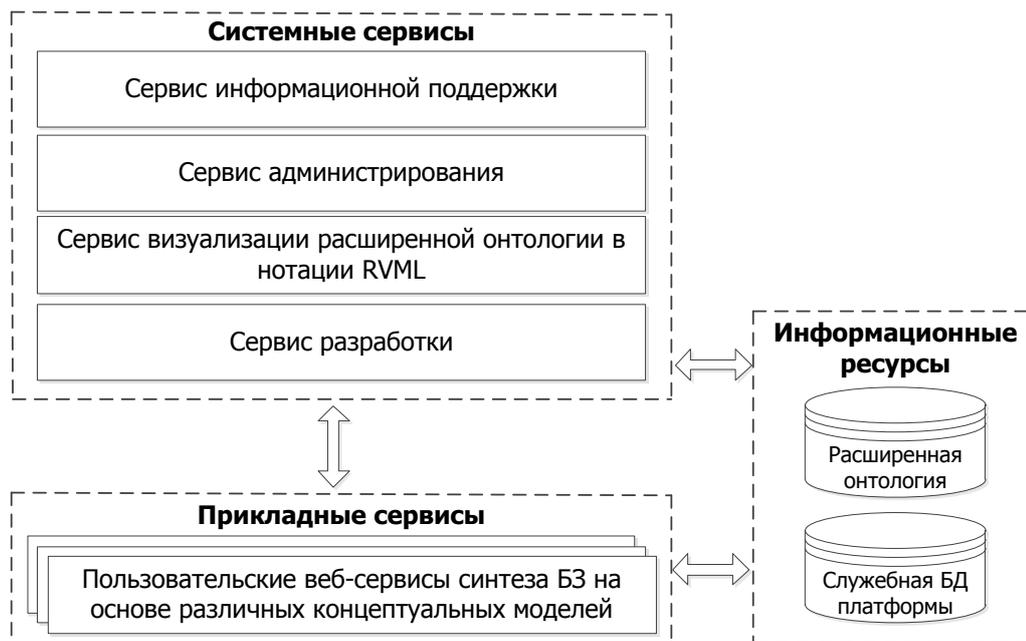


Рис. 1. Концептуальная архитектура сервис-ориентированной системы

Рассмотрим подробнее системные сервисы:

Сервис информационной поддержки – обеспечивает взаимодействие всех пользователей с платформой и поддерживает совместную работу пользователей или групп пользователей с проектами БЗ, в рамках которых ведется разработка БЗ.

Сервис администрирования – предоставляет возможность администраторам платформы управлять всеми системными сервисами и всеми зарегистрированными пользователями.

Сервис визуализации расширенной онтологии в нотации RVML – представляет собой графический редактор обеспечивающий возможность визуального отображения и изменения понятий и причинно-следственных отношений расширенной онтологической модели, при помощи специальной нотации Rule Visual Modeling Language (RVML) [4].

Сервис разработки – обеспечивает разработку прикладных пользовательских сервисов для автоматизированного формирования БЗ на основе трансформации различных концептуальных моделей. Данный сервис включает следующие модули:

- Модуль поддержки импорта концептуальных моделей, их метамodelей, а также метамodelей БЗ;
- Модуль визуального построения правил соответствий (трансформации) элементов;

- Транслятор (модуль экспорта) концептуальных моделей в код на ЯПБЗ, на основе построенных правил преобразования.

## **2. Метод трансформации концептуальных моделей в код на ЯПБЗ**

На сегодняшний день разработано множество подходов (языков) трансформации моделей. При этом можно выделить три основных направления трансформации моделей:

- трансформация с использованием графовых грамматик (Graph Rewriting) [6];
- трансформация моделей на основе написания правил соответствий элементов метамodelей;
- трансформация с использованием различных языков преобразований XML-документов.

Трансформация моделей является центральной частью модельно-ориентированного подхода к разработке программного обеспечения (Model-Driven Engineering, MDE). В рамках подхода MDE существуют и развиваются различные языки трансформации моделей, которые воспринимают входные модели, соответствующие данной метамodelи и представляют их как выходные модели в соответствии с другой метамodelью.

Наиболее распространенными языками трансформации моделей являются: QVT (Query/View/Transformation) [7], ATL (ATLAS Transformation Language) [8], VIATRA (Visual Automated model TRAnsformations) [9], GReAT (Graph REwriting And Transformation) [10], XSLT (eXtensible Stylesheet Language Transformations) [11] и др.

Однако у всех существующих языков трансформации моделей есть ряд существенных недостатков, одним из основных является то, что на пользователя накладываются очень высокие требования к разработке трансформаций. Пользователю необходимо знать не только синтаксис языков трансформации моделей (умение писать правила преобразований на данном языке), который достаточно сложен, но также и языки метамodelирования. Следует также отметить, что программные средства, реализующие данные языки, зачастую не предоставляют возможность визуализации процесса трансформации моделей и правила таких преобразований приходится писать вручную. Все это затрудняет использование данных языков и средств трансформации моделей.

Таким образом, с целью преодоления отмеченных недостатков был разработан метод трансформации концептуальных (информационных) моделей. Основной задачей разработанного метода трансформации является преобразование исходных концептуальных моделей, построенных с использованием разных программных средств, в код целевой БЗ, представленной при помощи различных ЯПБЗ.

Данный метод представляет собой горизонтальную, экзогенную трансформацию, т.е. преобразование, исходной и целевой модели одного уровня иерархии, но описанные на различных языках моделирования (разные концептуальные языки моделирования и ЯПБЗ).

Основные принципы разработанного метода трансформации:

- 1) визуальное построение правил трансформации (соответствий) элементов метамodelи концептуальной модели в элементы метамodelи БЗ, без использования специальных языков трансформации моделей;
- 2) автоматическое выделение элементов метамodelей для процесса визуализации построения правил трансформации;
- 3) использование XML для представления описания исходных концептуальных моделей (XSD-метамodelь);

4) использование метамодели продукций (расширенной онтологии), обеспечивающей унифицированное описание правил.

Формализовать постановку задачи можно следующим образом. Необходимо определить оператор трансформации концептуальной модели  $T$ :

$$T : CM \rightarrow Code_{KRL}, \quad (1)$$

где  $CM$  – исходная концептуальная модель;  $Code_{KRL}$  – целевая модель представления знаний (код на ЯПБЗ).

Предполагается реализовать данный метод трансформации в стандартном системном сервисе разработки программных модулей синтеза БЗ, в составе сервис-ориентированной системы.

## 2.1. Общий алгоритм формирования БЗ на основе различных концептуальных моделей

В общем виде алгоритм преобразования может быть представлен в виде последовательности действий (Рис. 2).

На этапах 1 и 2 преобразования средствами внешних программ пользователь строит концептуальную модель, которая представляется в формате XML. Этот формат является универсальным и наиболее распространенным способом интеграции программных систем и обеспечения обмена информацией между приложениями.

Например, при представлении UML-моделей в виде XML-структур используется стандарт XMI (XML Metadata Interchange), предназначенный главным образом для хранения UML-моделей и обмена ими между различными инструментальными средствами разработки.

На этапе 3 пользователь, средствами внешних программ, формирует XSD-метамодель, которая соответствует данной концептуальной модели, а также метамодель, с использованием стандарта MOF, представляющая структуру целевой модели представления знаний.

Этапы 4 и 5 подразумевают автоматизированное построение пользователем таблицы соответствий элементов (рис. 3), состоящей из правил трансформации (перевода) моделей, при этом:

*Таблица соответствий* – это два столбца, левая колонка содержит все элементы исходной метамодели, а правая все элементы целевой метамодели. Каждая строка (кортеж) таблицы представляет собой элементарное, жесткое (бинарное), однонаправленное правило трансформации (перевода).

*Правило трансформации (перевода)* – это соответствие одного элемента исходной метамодели с одним элементом целевой метамодели.

Формализуем процесс трансформации моделей в БЗ с использованием таблицы соответствий  $Tb$ :

$$Tb = (r_1, r_2 \dots r_n),$$

где  $r_i$  – правило трансформации, при этом  $r_i = \langle L_i, R_i, P_i \rangle$ , где  $L_i$  – левая часть правила (элемент исходной метамодели);  $R_i$  – правая часть правила (элемент целевой метамодели);  $P_i$  – приоритет выполнения правила, определяющий последовательность выполнения правил в таблице,  $i = \overline{1, n}$ .

Таким образом, на этапе 4 данная таблица соответствий строится автоматически на основе анализа исходной XSD-метамодел и целевой MOF-метамодел и не содержит правил трансформации (*первоначальная таблица без правил  $Tb_{BEGIN}$* ).

После чего пользователь, при помощи специального модуля визуального формирования соответствий, задает правила преобразования элементов  $r_i$  (этап 5), т.е. соединяет левые поля таблицы с необходимыми правыми, тем самым конкретизируя первоначальную таблицу соответствий  $Tb_{BEGIN}$  (*конечная таблица с правилами  $Tb_{END}$* ).

На основе построенной таблицы соответствий  $Tb_{END}$  выделяются понятия, предметной области, их свойства, отношения и формируется расширенная онтология (этап 6), как универсальное представление знаний, независящее от используемого ЯПБЗ или исходной концептуальной модели. При помощи специальной графической нотации RVML предоставляется возможность визуализации, модификации и проверки полученных знаний (продукций) (этап 7).

На этапе 8 происходит генерация кода БЗ в целевом формате на основе расширенной онтологии или непосредственно из исходной концептуальной модели.

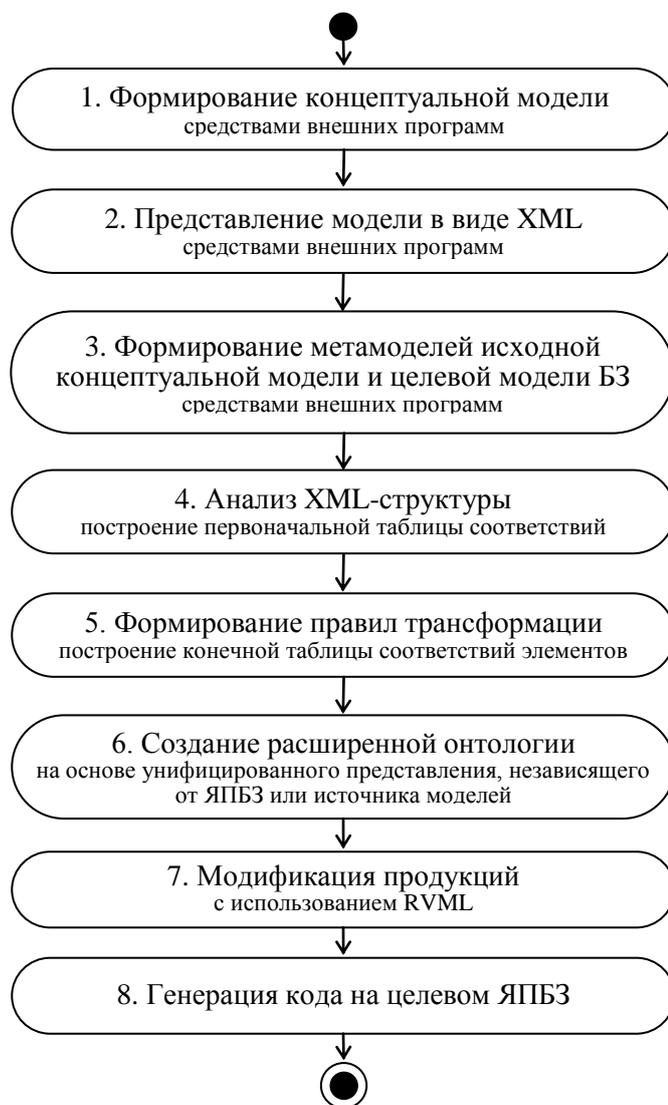


Рис. 2. Обобщенный алгоритм формирования кода целевой БЗ на основе исходных концептуальных моделей

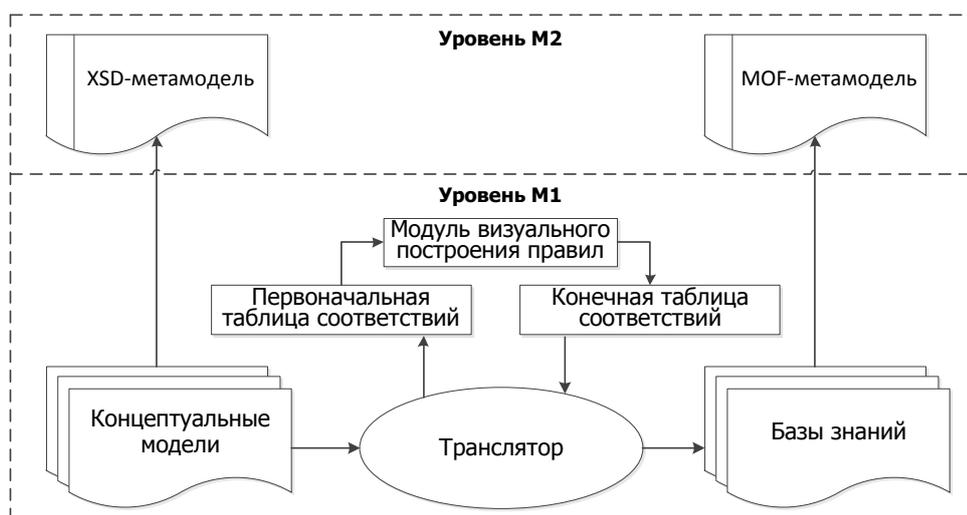


Рис. 3. Трансформация исходных концептуальных моделей в целевые БЗ, представленные на различных ЯПБЗ

Следует отметить, что данный метод трансформации учитывает три варианта преобразования:

- 1) Перевод исходной концептуальной модели в расширенную онтологию;
- 2) Отображение расширенной онтологии в код на целевом ЯПБЗ;
- 3) Трансформация исходной концептуальной модели непосредственно в код на целевом ЯПБЗ, без использования расширенной онтологии.

Во всех трех случаях строится таблица соответствий элементов и задаются правила трансформации:

- 1) Таблица соответствий элементов исходной метамодели с элементами расширенной онтологии;
- 2) Таблица соответствий элементов расширенной онтологии с элементами целевой метамодели модели представления знаний (код на ЯПБЗ);
- 3) Таблица соответствий элементов исходной метамодели непосредственно с элементами целевой метамодели модели представления знаний (код на ЯПБЗ).

Таким образом, уточним оператор преобразования концептуальной модели из (1):

$$T = \langle T_{CM-ONT}, T_{ONT-Code}, T_{CM-Code} \rangle,$$

$$T_{CM-ONT} : M_{CM} \rightarrow M_{ONT}, T_{ONT-Code} : M_{ONT} \rightarrow Code_{EKL}, T_{CM-Code} : M_{CM} \rightarrow Code_{KRL}, \quad (3)$$

где  $T_{CM-ONT}$  – оператор преобразования концептуальной модели в расширенную онтологию;  $T_{ONT-Code}$  – оператор преобразования расширенной онтологии в код на ЯПБЗ;  $T_{CM-Code}$  – оператор преобразования концептуальной модели в код на ЯПБЗ;  $M_{CM}$  – представление концептуальной модели в формате XML;  $M_{ONT}$  – представление концептуальной модели в расширенной онтологии;  $Code_{KRL}$  – код БЗ на целевом ЯПБЗ.

## 2.2 Расширенная онтология

Для унифицированного хранения и представления знаний разработана специальная модель расширенной онтологии. Разработанная модель позволяет абстрагироваться от особенностей описания знаний в различных ЯПБЗ, используемых при реализации БЗ

(например, CLIPS, Jess, OWL/SWRL и др.), и позволяет хранить знания в собственном независимом формате.

Формализуем описание модели расширенной онтологии:

$$M_{ONT} = \langle DT_{ONT}, CN_{ONT}, PN_{ONT}, Obj_{ONT}, R_{ONT} \rangle,$$

где  $DT_{ONT}$  – перечень базовых типов данных, при этом  $DT_{ONT} = \{\text{литерал, объект, коллекция}\}$ ,  $CN_{ONT}$  – имена классов,  $PN_{ONT}$  – имена свойств классов,  $Obj_{ONT}$  – понятия (константы, объекты) предметной области,  $R_{ONT}$  – конечное множество отношений между концептами заданной предметной области.

Уточним введенные понятия онтологии и расширим определение из [12], путем введения причинно-следственного отношения:

$R_{ONT} = \{R_{ONT}^{IS-A}, R_{ONT}^P, R_{ONT}^C\}$ , где  $R_{ONT}^{IS-A}$  – отношение наследования между классами  $CN_{ONT}$ ,  $R_{ONT}^P$  – отношение между классами и свойствами,  $pn R_{ONT}^P \langle cn, cn\_bt \rangle$ , где  $pn \in PN_{ONT}$ ,  $cn \in CN_{ONT}$ ,  $cn\_bt \in CN_{ONT} \cup DT_{ONT}$ , последнее означает, что свойство с именем  $pn_i$  характеризует класс  $cn_j$ , а значениями свойства могут быть элементы типа другого класса из  $CN_{ONT}$  или основного множества типов  $DT_{ONT}$ ,  $R_{ONT}^C$  – причинно-следственные отношения между концептами. В свою очередь  $R_{ONT}^C = \{r_{ONT_1}^C \dots r_{ONT_g}^C\}$ ,  $r_{ONT_j}^C = \langle name_j, cn_{LHS_j}, cn_{RHS_j}, op_j \rangle$ ,  $j \in \overline{1, g}$ ; где  $name_j$  – имя отношения,  $cn_{LHS_j}$  – левый класс отношения (ссылка на класс),  $cn_{RHS_j}$  – правый класс отношения (ссылка на класс),  $op_j$  – оператор отношения,  $op_j \in \{and, or, not\}$ .

### Заключение

Решение задач связанных с получением, структурированием и формализацией (представлением) знаний позволяет повысить эффективность процесса разработки БЗ ИС. В настоящей работе предлагается повысить эффективность процесса разработки БЗ, путем разработки программной системы для создания веб-сервисов синтеза БЗ, на основе автоматизированного анализа концептуальных моделей, выполненных в различных программных средствах.

При решении задач исследования был проведен анализ области получения, концептуализации и формализации знаний, а также программных средств концептуального, когнитивного и онтологического моделирования. Проанализированы подходы, языки и средства трансформации моделей. На основании проведенного анализа были выявлены недостатки существующих подходов, средств, языков и принято решение о разработке сервис-ориентированной системы создания программных модулей синтеза БЗ. Описана общая концепция сервис-ориентированной системы и ее основных элементов. Разработан метод трансформации концептуальных моделей в код, на различных ЯПБЗ, основанный на визуальном построении правил соответствий элементов исходной и целевой метамодели. Разработанный метод предполагается реализовать в составе исследовательского прототипа сервис-ориентированной системы, в виде системного сервиса разработки программных модулей синтеза БЗ.

Безусловно, данный подход не позволяет исключить ошибки обусловленные неточностью или неполнотой анализируемых концептуальных моделей, однако,

автоматическая генерация программного кода на основе моделей позволяет использовать принцип быстрого прототипирования при реализации БЗ, с последующей их проверкой и тестированием в сторонних средствах. По результатам тестирования возможно внесение необходимых изменений в модели (при помощи RVML) и повторная генерация программного кода БЗ.

### Литература

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000. – 384 с.
2. Giarratano, J.C., Riley, G. Expert Systems: Principles and Programming, 4th Edition, Thomson Course Technology, 2005. – 288 p.
3. Частиков А.П., Гаврилова Т.А., Белов Д.Л. Разработка экспертных систем. Среда CLIPS. СПб.: БХВ-Петербург, 2003. – 608 с.
4. Грищенко М.А., Юрин А.Ю., Павлов А.И. Разработка экспертных систем на основе трансформации информационных моделей предметной области // Программные продукты и системы. – 2013. – №3. – С. 143-147.
5. Дородных Н.О., Юрин А.Ю. Использование диаграмм классов UML для формирования продукционных баз знаний // Программная инженерия. – 2015. – №4. – С. 3-9.
6. Rozenberg G. Handbook of Graph Grammars and Computing by Graph Transformations, 1997. – 553 p.
7. Документация стандарта Query/View/Transformation (QVT), V1.2. 2015. URL: <http://www.omg.org/spec/QVT/1.2> (дата обращения: 15.09.2015)
8. Документация языка ATL Transformation Language (ATL). 2015. URL: [https://wiki.eclipse.org/MMT/ATL\\_Transformation\\_Language\\_\(ATL\)](https://wiki.eclipse.org/MMT/ATL_Transformation_Language_(ATL)) (дата обращения: 15.09.2015)
9. Varro D., Balogh A. The model transformation language of the VIATRA2 framework // Science of Computer Programming. – 2007. – Vol. 63. – No. 3. – P. 214-234.
10. Balasubramanian D., Narayanan A., Buskirk C., Karsai G. The graph rewriting and transformation language: GReAT // Electronic Communications of the EASST. – 2007. – Vol. 1. – P. 1-8.
11. Документация стандарта XSL Transformations (XSLT) Version 2.0. 2015. URL: <http://www.w3.org/TR/xslt20> (дата обращения: 15.09.2015)
12. Николайчук О.А. Методы, модели и инструментальное средство для исследования надежности и безопасности сложных технических систем // Автореферат на соискание степени доктора технических наук по специальности 05.13.01. ИСА РАН: Москва, 2011. – 37 с.

UDC 004.89

SOFTWARE CONCEPTION FOR DESIGN OF WEB-SERVICES FOR KNOWLEDGE  
BASES SYNTHESIS BASED ON CONCEPTUAL MODELS

*N.O. Dorodnyhk, A.Yu. Yurin, S.A. Korshunov*

*Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian  
Academy of Sciences, Irkutsk*

*e-mail: [tualatin32@mail.ru](mailto:tualatin32@mail.ru), [iskander@icc.ru](mailto:iskander@icc.ru), [grey.for@gmail.com](mailto:grey.for@gmail.com)*

*The paper describes the conception of service-oriented software for design of software modules (web-services) for knowledge bases synthesis based on automated analysis of conceptual models. The conceptual architecture of this system is considered. The general algorithm of conceptual models transformation represented in XML to the code of knowledge bases is described. The technique based on the visual design of the corresponding (transformation rules) between the source and target elements of metamodels. Feature of this approach is automatic extraction of metamodel elements for visualization of transformation rules design as well as the use of special notation of RVML for specification of the obtained rules.*

*Keywords: technology of software design, service, model transformation, knowledge base, knowledge acquisition, conceptual model, ontology*