

УДК 004.891.3

## РАЗРАБОТКА ДИАГНОСТИЧЕСКИХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ОСНОВЕ ОНТОЛОГИЙ

М.А. Грищенко<sup>1,a</sup>, Н.О. Дородных<sup>2,b</sup>, С.А. Коршунов<sup>1,c</sup>, А.Ю. Юрин<sup>2,d</sup>

<sup>1</sup> ООО «ЦентраСиб», Иркутск, Россия

<sup>2</sup> Институт динамики систем и теории управления им. В.М. Матросова СО РАН, Иркутск, Россия

<sup>a</sup> makcmg@icc.ru, <sup>b</sup> tualatin32@mail.ru, <sup>c</sup> grey.for@gmail.com, <sup>d</sup> iskander@icc.ru

### Аннотация

Создание интеллектуальных проблемно-ориентированных систем различного назначения, направленных на решение слабоструктурированных задач, остаётся актуальной проблемой, эффективность решения которой может быть повышена путём совершенствования подходов (методологий) или создания специализированного программного обеспечения. В работе описывается подход к разработке интеллектуальных диагностических систем, включая базы знаний, на основе онтологических моделей. В качестве методологической основы используется модельно-управляемый подход, в рамках которого произведено переопределение основных этапов, уточнены модели, определены правила трансформации и средства реализации. Реализация предлагаемого подхода осуществлена на примере создания диагностических систем. Под диагностированием понимается определение состояния объекта, в том числе: контроль состояния, поиск места и определение причин нарушения функционирования (отказ, неисправность), прогнозирование состояния. Объекты, подвергаемые диагностированию, описываются в виде последовательности состояний, где состояние – это множество параметров объекта и их значений в заданный момент времени. Особенностью реализации подхода является использование: онтологии в качестве вычислительно-независимой модели, описывающей особенности решаемой задачи; оригинального визуального языка моделирования правил (Rule Visual Modeling Language) для описания платформо-независимых и платформо-зависимых моделей и языка программирования C Language Integrated Production System. Описан пример применения подхода с использованием прототипов программных средств: Knowledge Base Development System и Personal Knowledge Base Designer.

**Ключевые слова:** модельно-ориентированный подход, интеллектуальная система, база знаний, онтология, правила, концептуальная модель.

**Цитирование:** Грищенко М.А. Разработка диагностических интеллектуальных систем на основе онтологий / М.А. Грищенко, Н.О. Дородных, С.А. Коршунов, А.Ю. Юрин // Онтология проектирования. – 2018. – Т. 8, №2(28). – С.265-284. – DOI: 10.18287/2223-9537-2018-8-2-265-284.

### Введение

В настоящее время разработка проблемно-ориентированных интеллектуальных систем (ИС), направленных на решение сложных, специфических задач (например, в области диагностирования сложных технических систем, прогнозирования чрезвычайных ситуаций, энергетической безопасности, в медицине и т.д.) остаётся актуальной задачей [1-5]. Повышение эффективности создания программного обеспечения (ПО) подобного вида может быть достигнуто путём совершенствования подходов (методологий) к разработке специализированного инструментария. Одной из тенденций в этой области является использование визуальных информационных (концептуальных) моделей, в том числе онтологий [6-8] и специальных программных инструментов в виде редакторов для концептуального (например, ИМС SmartTools и др.) или онтологического моделирования (например, Protégé, OntoStudio, OntoEdit, FluentEditor, WebOnto, OntoMap и др.). Данные системы позволяют разрабатывать модели, соответствующие знаниям эксперта какой-либо предметной области (ПрО). Однако

большинство из них не охватывает все этапы создания ИС и не обеспечивает полноты процесса разработки: от моделей ПрО до программных кодов, в некоторых случаях позволяя получать только графические артефакты (изображения) структур баз знаний (БЗ). Исключением является система Protégé, которая позволяет не только моделировать онтологии разной сложности, но и обеспечивает интеграцию разработанных онтологий с правилами в форматах Jess [9] (с использованием специального плагина JessTab [10]) или SWRL [11] (при помощи плагина SWRLTab [12]). При этом необходимо отметить, что данная система ориентирована на специалистов в области инженерии знаний, знакомых с существующими моделями и языками представления знаний (ЯПЗ). Таким образом, создание специализированного алгоритмического и программного обеспечения, поддерживающего не только моделирование знаний (в том числе в форме онтологий), но и генерацию программных кодов и приложений на основе построенных моделей, является актуальным.

## **1 Разработка ИС на основе онтологий и модельно-управляемого подхода**

### **1.1 Краткий обзор работ**

В работе Загорулько Ю.А. [13] предлагается технология построения ИС, ориентированная на экспертов и базирующаяся на специализированных оболочках, настроенных на определённые области знаний и классы задач. В качестве базиса представления знаний предлагается интегрированная модель с ведущей ролью онтологий. В [14] обсуждается проект компонентного проектирования ИС (OSTIS) на основе открытых семантических технологий (моделей и языков). В [15] описывается подход к разработке семантических информационных систем, в архитектуру которых предложено ввести «вариативную часть» в виде онтологии, определяющей специфику ПрО, методов и задач. В [16] предлагается подход к проектированию систем управления на основе трансформаций онтологий (онтологических описаний управляющего приложения на основе функциональных блоков IEC 61499). В [17] описывается проект «умного города» – «DIMMER», основанный на использовании онтологий и направленный на предоставление гибкой интернет-платформы для разработки приложений, сбора и интеграции данных, используя информацию о зданиях и сооружениях, энергетических сетях и т.д. В [18] представлена концепция, на основе спецификации MDA (Model Driven Architecture), для реализации проекта совместимых производственных систем знаний (IMKS). Центральным элементом в данной концепции являются онтологии, в частности производственная онтология (Manufacturing Core Ontology), которая используется как платформо-независимая модель в цепочке преобразований моделей. В [19] предлагается фреймворк «Smart Modeller» для онтологического проектирования ПО (средство входит в проект – SOFIA). Фреймворк ориентирован на программистов и включает в себя более гибкие и производительные средства для онтологического моделирования на основе гибридизации дескрипционной логики с шаблонами SPARQL. В [20] предложен инструмент (среда) для разработки ПО в области здравоохранения, и в частности – системы администрирования пациентов, с использованием онтологий. Архитектура инструмента состоит из трёх слоев: слой онтологий, агентов/ЯПЗ и приложений – RDQL, генераторы кода, редакторы онтологий. В [21] предлагается система поддержки проектирования (анализа и визуализации) ирригационного трубопровода на основе онтологии. В [22] описывается интеллектуальная система медицинских рекомендаций по клинической практике для лечения сахарного диабета, основанная на онтологии ПрО. Анализ этих и подобных исследований [23-30] показал широкое использование онтологий, а также принципов концептуального, визуального моделирования и

проектирования, в частности, модельно-ориентированного подхода (Model Driven Engineering - MDE) [31] и его реализаций.

MDE (или Model Driven Development - MDD) – это подход к разработке ПО, использующий информационные модели в качестве основных артефактов, которые, в свою очередь, могут быть использованы для получения других моделей (например, с более высокой детализацией) и генерации программных кодов [32, 33]. Одной из самых известных инициатив MDE является подход MDA [34], спецификации которого стандартизованы консорциумом OMG (Object Management Group). Разработка теории и инструментов для создания ПО на основе отображения онтологий осуществляется в рамках таких направлений MDE/MDD как: Ontology Driven Architecture (ODA) [35] и Ontology Driven Software Engineering (ODSE) [36]. Данные направления основаны на исследованиях взаимодействия методов и средств программной инженерии с семантическими технологиями.

В данной работе предлагается повысить эффективность разработки проблемно-ориентированных ИС путём автоматического анализа и трансформации онтологических моделей ПрО с использованием принципов MDA/MDD-подхода. В качестве ПрО выбраны диагностические ИС для определения причин повреждения и разрушения конструкционных материалов в нефтехимии. Под диагностированием понимается определение состояния объекта, в том числе: контроль состояния, поиск места и определение причин нарушения функционирования (отказ, неисправность), прогнозирование состояния. Объекты, подвергаемые диагностированию, описываются в виде последовательности состояний, где состояние – это множество параметров объекта и их значений в заданный момент времени. Параметры объектов могут быть как числовыми, так и качественными, последний тип преобладает в связи со слабой структурированностью ПрО. Динамика состояний может быть структурирована в виде классов состояний (событий), обусловленных этапами/стадиями жизненного цикла объекта, этапами развития физических (физиологических) процессов и т.п. и представлена в виде причинно-следственных отношений, формирующих цепочки [4].

## 1.2 Постановка задачи

В данной работе онтологические модели использованы в качестве исходных артефактов в процессе создания диагностических ИС. С формальной точки зрения, необходимо обеспечить преобразование онтологических моделей в программные коды и спецификации, то есть определить оператор преобразования:

$$(1) \quad T : Ont \rightarrow Code ,$$

где *Ont* – онтологическая модель; *Code* – программный код и спецификации БЗ и ИС, т.е.:  $Code = \langle KB, IS \rangle$ .

В последнее время активно развиваются автоматизированные методы формирования онтологий на основе различных информационных источников: баз данных, электронных таблиц и текстовых документов, концептуальных моделей (схем, диаграмм, концепт-карт) и т.д. Подобные методы позволяют повторно использовать большие объёмы накопленной разнородной информации. Принимая во внимание данный аспект, предлагается дополнить разрабатываемый подход этапом автоматического формирования онтологий на основе анализа и преобразования концептуальных моделей ПрО.

Обобщённая концептуальная схема предлагаемого подхода к созданию проблемно-ориентированных ИС на основе онтологий показана на рисунке 1.

В качестве целевого языка программирования выбран CLIPS (C Language Integrated Production System) [37] - широко используемый язык и инструментальная среда для разработки продукционных экспертных систем.

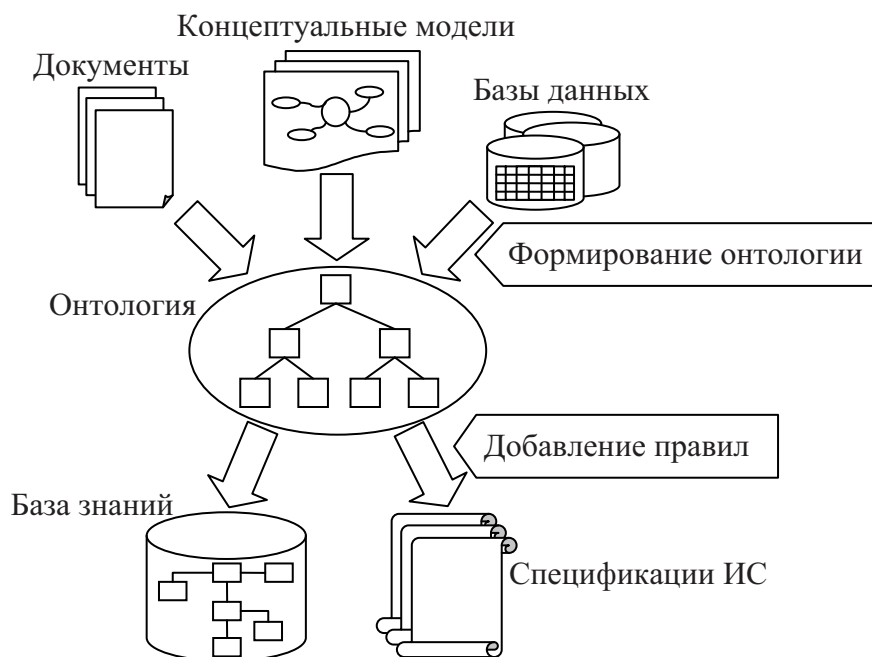


Рисунок 1 - Концептуальная схема предлагаемого подхода

Реализацию подхода предлагается осуществить на основе авторского ПО:

- веб-ориентированной программной системы – Knowledge Base Development System (KBDS), основной задачей которой является генерация онтологий на основе трансформации различных концептуальных моделей [38];
- специализированного настольного редактора – Personal Knowledge Base Designer (PKBD) [39, 40], основной задачей которого является поддержка MDA/MDD-подхода и автоматическая генерация CRUD-интерфейса для БЗ.

Тогда уточним (1):  $KB = CLIPS$ ,  $IS = PKBD$ , где *CLIPS* – программные коды на CLIPS, *PKBD* – спецификация модели ИС для интерпретатора PKBD.

Предлагаемый подход включает модели, методы и программные средства, в том числе, правила и алгоритмы для преобразования элементов онтологии в программные коды.

### 1.3 Основные этапы предлагаемого подхода

**Этап 1:** Формирование онтологии ПрО.

Основным результатом данного этапа являются онтологические модели. С точки зрения MDA-подхода этот этап соответствует стадии создания вычислительно-независимой модели (Computation Independent Model - CIM). CIM описывает то, что должна делать система, но скрывает все описания, связанные с реализующими технологиями, чтобы обеспечить максимальную независимость модели от способов реализации. В сущности, CIM представляет собой часть технического задания, описывающую модель ПрО и элементы программной системы. В предлагаемом подходе CIM реализована в форме онтологии. Так как класс создаваемых систем известен априори, то пользователю необходимо построить только онтологию ПрО, которая содержит основные понятия, их свойства и отношения. С точки зрения стандартной методологии разработки систем, основанных на знаниях [41], этот этап соответствует этапам идентификации, получения и концептуализации знаний.

Для решения задач данного этапа могут быть использованы инструменты онтологического моделирования (например, Protégé и др.). В статье предлагается автоматизировать дан-

ный этап при помощи средств KBDS, который позволяет создавать программные компоненты (модули-конверторы) ИС, обеспечивающие автоматическое формирование онтологий на основе преобразования различных концептуальных моделей, представленных в XML-подобных форматах. В частности, это могут быть диаграммы классов UML (Unified Modeling Language) в формате XMI или концепт-карты IHMC StarTools в формате XTM. Таким образом, расширим (1):

$$(2) \quad T : CM \rightarrow Ont \rightarrow Code ,$$

где  $CM$  – исходная концептуальная модель.

Формализованное описание структуры онтологической модели представлено в [42]. Данная модель позволяет абстрагироваться от особенностей описания знаний в различных ЯПЗ, используемых при реализации онтологий, и хранить знания в собственном независимом формате. При этом KBDS предоставляет дополнительную возможность выгрузки (экспорта) сформированной онтологической модели в формат OWL [45]. Таким образом, разработанные на основе концептуальных моделей онтологии могут быть использованы в других приложениях.

**Этап 2:** Разработка онтологий и правил.

Основными результатами данного этапа являются формализованные описания элементов производственной модели БЗ и ИС. С точки зрения MDA-подхода этот этап соответствует стадии создания платформо-независимой модели (Platform Independent Model - PIM). В предлагаемом подходе PIM описывает логические правила (продукции), которые получены в результате автоматизированного преобразования CIM. Таким образом, таксономия онтологических понятий преобразуется в набор фактов и элементов правил (таких как условия и действия), а причинно-следственные отношения преобразуются в правила. Фактически, осуществляется комбинация онтологических знаний с правилами. PIM обеспечивает достаточную степень независимости от средств реализации, чтобы в дальнейшем её отобразить на одну или несколько программных платформ.

Визуальное моделирование является одним из основных аспектов методологии MDA/MDD. При этом MDA традиционно использует язык UML [44]. Однако UML не предназначен для наглядного и однозначного представления причинно-следственных связей в виде правил. Для устранения данного недостатка предлагается использовать оригинальную авторскую графическую нотацию – Rule Visual Modeling Language (RVML) [45] для визуального представления логических правил (продукций). Эта нотация позволяет описывать причинно-следственные связи, абстрагируясь от особенностей конкретных языков программирования БЗ производственного типа.

**Этап 3:** Платформо-ориентированная модификация производственной модели БЗ.

Основными результатами данного этапа является платформо-ориентированное (ориентированное на определённый ЯПЗ) описание элементов производственной модели БЗ. С точки зрения MDA-подхода этот этап соответствует стадии создания платформо-зависимой модели (Platform Specific Model, PSM). PSM объединяет спецификации в PIM с деталями, необходимыми для определения того, как система использует определённый тип платформы. PSM позволяет учитывать особенности определённого ЯПЗ (в нашем случае - это CLIPS), такие как приоритеты правил, значения «по умолчанию» для слотов и т.д. Для построения PSM также используется RVML. С точки зрения стандартной методологии разработки систем, основанных на знаниях [41], этапы 2 и 3 соответствуют формализации знаний.

**Этап 4:** Генерация программных кодов и спецификаций.

Основными результатами этого этапа являются код БЗ в формате CLIPS и спецификация модели ИС для интерпретатора PKBD. При этом существует возможность получения спецификаций в двух формах:

- в виде типизированных текстовых файлов (см. таблицы 1, 2) - используется для совместимости с более ранними разработками авторов, в частности, с программной системой для поддержки проведения экспертизы промышленной безопасности [46];
- в виде XML-файлов (таблица 3) формата PKBD.

Таблица 1 - Структура типизированного файла для описания шаблона факта

<i>Элемент файла</i>	<i>Описание</i>	<i>Пример</i>
<b>[Metadata]</b>	Заголовок файла	<b>[Metadata]</b>
;description	Описание шаблона	;Наблюдаемые повреждения
tempale_name= <template name>	Имя шаблона, поле используется для синтеза графического пользовательского интерфейса и программных кодов	tempale_name=Damage
edited_by_user= <No Yes>	Инструкция для генератора пользовательского интерфейса, определяющая возможность создания форм редактирования для данного шаблона	edited_by_user=No
<b>[Fields]</b>	Заголовок раздела описания слотов шаблона	<b>[Fields]</b>
<slot_name>= <datatype>	Описание слота, где <slot_name> – внутреннее имя, используемое для генерации графического пользовательского интерфейса и кодов на CLIPS; <datatype> = string   integer   float   <имя_переменной_из_списка_значений>, где <имя_переменной_из_списка_значений> = <значение>; ...; <значение> определяется в разделе «Values»	dam-type=string dam-orientation=val3:1
<b>[Captions]</b>	Заголовок раздела для наименований слотов шаблона, которые отображаются на пользовательских формах и используются для мультиязыковой поддержки	<b>[Captions]</b>
<slot_name> = <name>	Описание слота	form=Exist damage dam-type=Type of the damage
<b>[Values]</b>	Заголовок раздела описания возможных значений слотов	<b>[Values]</b>
<value_name> = <value_1>, ..., <value_N>	Описание возможных значений слотов	Val3:1=LONGITUDINAL, CROSS-SECTION

Таблица 2 - Структура типизированного файла для описания шаблона правила

<i>Элемент файла</i>	<i>Описание</i>	<i>Пример</i>
<b>[Generalized rules]</b>	Заголовок файла	<b>[Generalized rules]</b>
#<Name of the process>	Наименование деградационного процесса, который описывается правилом. Данное имя отображается на формах пользователя.	#Corrosion cracking
##<Name of the process modification>	Наименование модификации процесса или подпроцесса. Данное имя отображается на формах пользователя.	##1
<rule_name> = <title> : <template names from the rule conditions> : <template names from the rule actions>	<rule_name> – имя правила, используемое для синтеза элементов пользовательского интерфейса и кодов CLIPS, <title> – имя правила, отображаемое на формах пользователя, <template names from the rule conditions> = <template_name>, ..., <template_name>.	fail-mechanism-fail-ky = Rule_for defenition_of_the failure_mechanism: exist-meh-des, exist-des: exist-meh-fail

Таблица 3 - Структура XML-файла формата PKBD

Элемент файла	Описание
<KnowledgeBase/>	Описание БЗ, содержит основные вложенные элементы: <Templates/> – шаблоны фактов; <Facts/> – факты; <Rules/> – правила; <GRules/> – шаблоны правил.
<Template/>	Описание шаблона, содержит вложенные элементы: <ID/> – идентификатор; <Name/> – имя для форм пользователя; <ShortName/> – имя для генератора форм пользователя и кода CLIPS; <Description/> – описание; <Slots/> – слоты.
<Slot/>	Описание слота, содержит вложенные элементы: <Name/> – имя для форм пользователя; <ShortName/> – имя для генератора форм пользователя и кода CLIPS; <Description/> – описание; <Value/> – значение, в зависимости от принадлежности слота шаблону или факту, это может быть либо значение «по умолчанию», либо фактическое значение; <DataType/> – тип данных; <Constraint/> – ограничение на значение.
<Fact/>	Описание факта аналогично описанию шаблона.
<GRule/>	Описание шаблона правила, содержит вложенные элементы: <ID/> – идентификатор; <Name/> – имя для форм пользователя; <ShortName/> – имя для генератора форм пользователя и кода CLIPS; <Description/> – описание; <Salience/> – важность правила или его приоритет; <Conditions/> – описание условий; <Actions/> – описание действий.
<Rule/>	Описание правила аналогично описанию шаблона правила.

**Эман 5:** Тестирование БЗ и ИС.

На данном этапе полученные программные коды и спецификации тестируются в PKBD с использованием встроенной машины вывода CLIPS. Пользователь (эксперт Про) разрабатывает только CIM и PIM. При разработке PSM требуется участие специалиста в области инженерии знаний. Описанная последовательность этапов концептуально совпадает со «стандартным» подходом MDA, но содержимое этапов переопределяется в соответствии со спецификой создания БЗ и ИС.

**1.4 Преобразования моделей**

Процесс разработки ПО на основе MDA-подхода представляет собой последовательность модельных преобразований от более абстрактных моделей к менее абстрактным:  $CIM \rightarrow PIM \rightarrow PSM \rightarrow Code$ . В соответствии с предлагаемой последовательностью этапов создания БЗ и ИС, модельные преобразования выглядят следующим образом:  $CM \rightarrow Ont \rightarrow RVML \rightarrow Code$ .

Уточним операторы преобразования из (1) и (2):

$$(3) \quad T = \langle T_{CM-ONT}, T_{ONT-RVML}, T_{RVML-Code} \rangle, \quad T_{CM-ONT} : M^{XML} \rightarrow M^{ONT}, \\ T_{ONT-RVML} : M^{ONT} \rightarrow M^{RVML}, \quad T_{RVML-Code} : M^{RVML} \rightarrow Code,$$

где  $T_{CM-ONT}$  – это оператор преобразования исходной концептуальной модели в онтологию;  $T_{ONT-RVML}$  – оператор преобразования онтологии в RVML-модель;  $T_{RVML-Code}$  – оператор преобразования RVML-модели в целевой код;  $M^{XML}$  – представление исходной концептуальной модели в виде XML-документа;  $M^{ONT}$  – онтологическая модель;  $M^{RVML}$  – RVML-модель для представления логических правил (продукций);  $Code$  – программный код БЗ и спецификации ИС.

Преобразование моделей представляет собой автоматическую генерацию целевой модели из исходной модели в соответствии с набором правил преобразования. Правило

преобразования – это описание того, как одна или несколько конструкций на исходном языке могут быть преобразованы в одну или несколько конструкций на целевом языке [34]. Можно выделить ряд основных направлений в области модельных трансформаций [47]:

- использование языков трансформации моделей общего назначения, например, Query/View/Transformation (QVT), ATLAS Transformation language (ATL), Epsilon и др.;
- использование языков, основанных на графовых грамматиках, например, Visual Automated model TRAnsformations (VIATRA2), Graph REwriting And Transformation (GReAT), Henshin и др.;
- использование языков для преобразования XML-документов, например, eXtensible Stylesheet Language Transformations (XSLT), XQuery и др.;
- использование декларативных и процедурных языков программирования.

В предлагаемом подходе оператор  $T_{CM-ONT}$  реализуется с помощью декларативного предметно-ориентированного языка – Transformation Model Representation Language (TMRL) [48]. Данный язык используется в KBDS и описывает модель (сценарий) трансформации, представляющую собой совокупность правил преобразования (установленных соответствий между элементами исходной и целевой метамодели) с определённым приоритетом (очередностью) выполнения каждого правила.

Формирование онтологических моделей на основе преобразования концептуальных предметных моделей и последующим экспортом в OWL-формат можно представить в виде следующей последовательности действий (см. рисунок 2).



Рисунок 2 - Процесс формирования OWL-онтологий



- 1) пользователь (эксперт) создаёт концептуальную модель с использованием «предметных» нотаций, представляющих собой удобный и понятный для специалиста-предметника способ представления знаний (например, концептуальные карты, интеллект-карты или карты знаний, диаграммы Венна и Исикавы, деревья событий, отказов и др.).
- 2) происходит автоматическое представление и сохранение построенных концептуальных моделей в формате XML. Для этой цели могут использоваться различные программные средства (редакторы), в частности CASE-средства, которые позволяют сохранять эти концептуальные модели в XML-подобных форматах.
- 3) построенная модель импортируется (загружается) в KBDS, где при помощи специального программного компонента (модуля-конвертора) осуществляется анализ XML-структуры концептуальной модели, в результате которого автоматически извлекаются понятия ПрО, их свойства и отношения.
- 4) автоматически генерируется онтологическая модель на основе извлечённых понятий и отношений.
- 5) полученную онтологическую модель можно модифицировать и визуализировать, используя специальный графический редактор, входящий в состав KBDS.
- 6) генерируется код онтологии в формате OWL (в данной работе рассматривается генерация OWL-кода, соответствующего синтаксису RDF/XML).

Преобразования  $T_{ONT-RVML}$  и  $T_{RVML-Code}$  реализованы с помощью процедурных языков программирования и скрыты от пользователя.

Предлагаемый подход реализуется в виде прототипов программных средств: KBDS [38] и РКВД [39, 40], которые взаимодействуют через API. При этом KBDS обеспечивает полную поддержку этапов 1 и 2 предлагаемого подхода и генерацию OWL-онтологий. РКВД обеспечивает частичную поддержку этапов 2-5, а также генерацию и интерпретацию спецификаций ИС и кодов БЗ.

## 2 Применение подхода к разработке диагностической ИС в нефтехимии

Основной причиной повреждения и разрушения конструкционных материалов в нефтехимическом оборудовании являются деградационные процессы (ДП), представляющие собой физико-химические процессы, обусловленные как различными технологическими процессами, так и структурными, производственными и эксплуатационными нарушениями, которые вызывают отказы механических и нефтехимических систем или аппаратов [49]. Каждый ДП характеризуется механизмом и кинетикой. Механизм ДП представляет собой набор свойств технологического объекта и влияющих факторов. Описание кинетики включает в себя: события, параметры событий, функциональные отношения для определения параметров события в определённый момент времени.

**Пример этапа 1.** Формирование онтологии ПрО осуществляется с использованием концептуальной модели динамики технических состояний [50]. В качестве источника концептуальных моделей в виде концептуальных карт использовалось ИМС SmartTools. Построенные концепт-карты описывают различные виды ДП на стадиях повреждения, разрушения и отказа. Фрагмент концепт-карты, описывающий таксономию понятия технического объекта, представлен на рисунке 3.

Фрагменты анализируемых концепт-карт, описывающих механизм ДП «водородное охрупчивание» и его кинетику на стадии повреждения, представлены на рисунках 4 и 5. При этом иерархия понятий описывается путём установления явного отношения «имеет подкласс», а отношения типа «имеет свойство» и «имеет» обозначают наличие явных связей

между концептом и его свойством. Все остальные связи могут быть интерпретированы как причинно-следственные отношения.

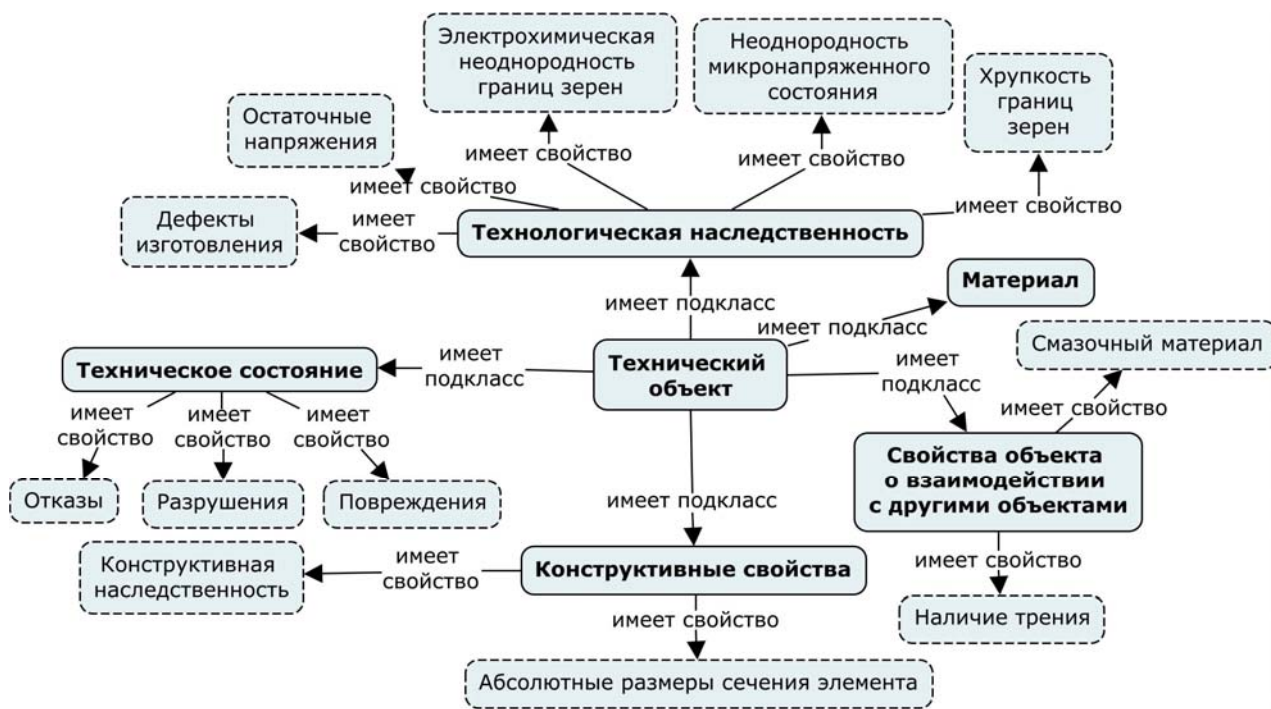


Рисунок 3 - Фрагмент концепт-карты описания технического объекта

Сформированные концепт-карты сохранялись в виде XML-документов. Понятия, их свойства и отношения, извлечённые из данных XML-документов, отображались в онтологическую модель. Полученная онтология представлена и модифицирована с использованием визуального редактора онтологических моделей KBDS (см. рисунок 6).

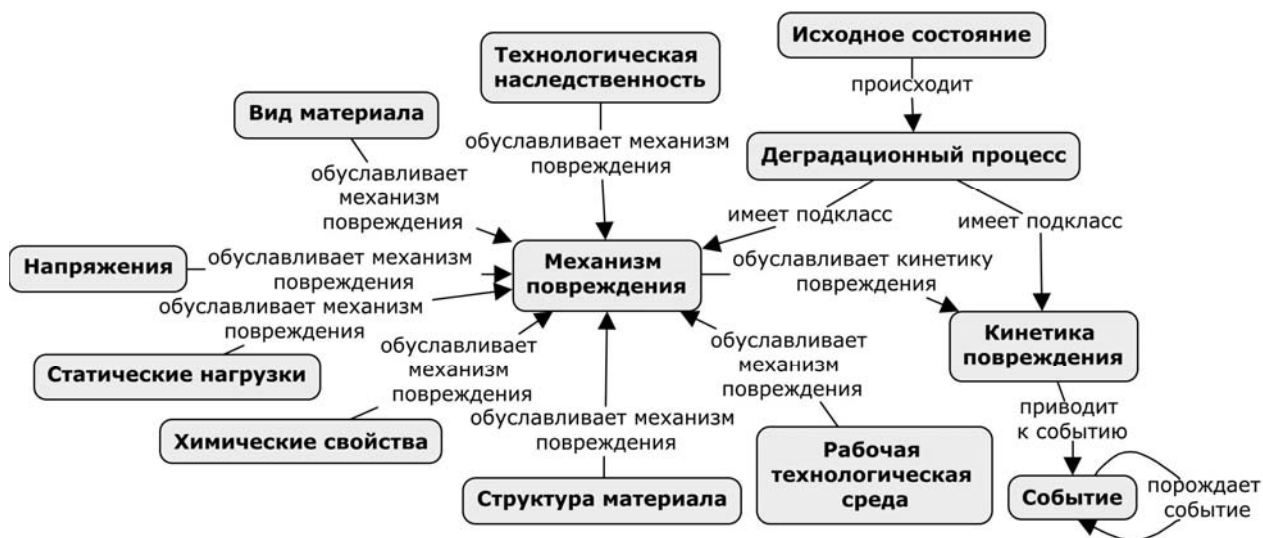


Рисунок 4 - Фрагмент концепт-карты структуры ДП

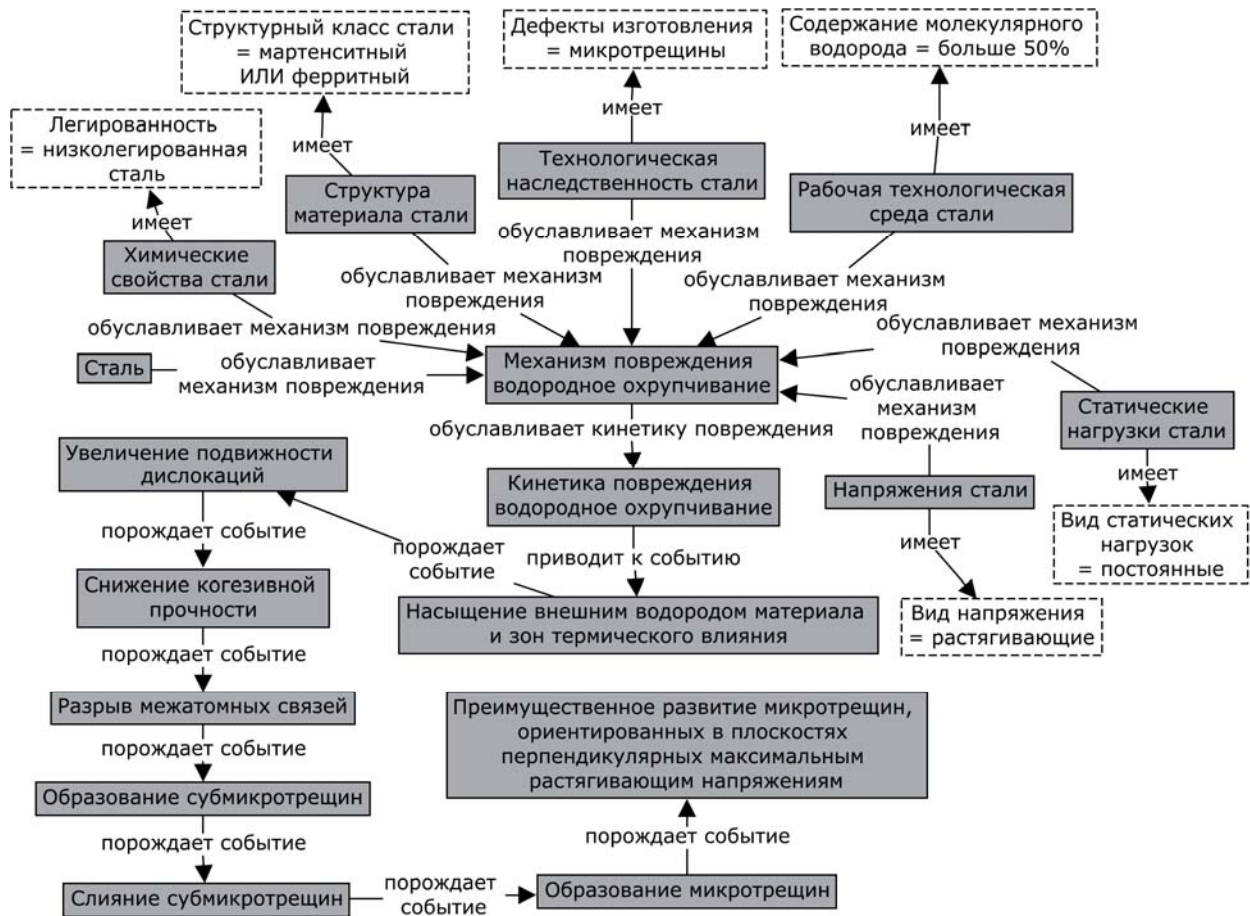


Рисунок 5 - Фрагмент концепт-карты, конкретизирующей ДП «водородное охрупчивание»

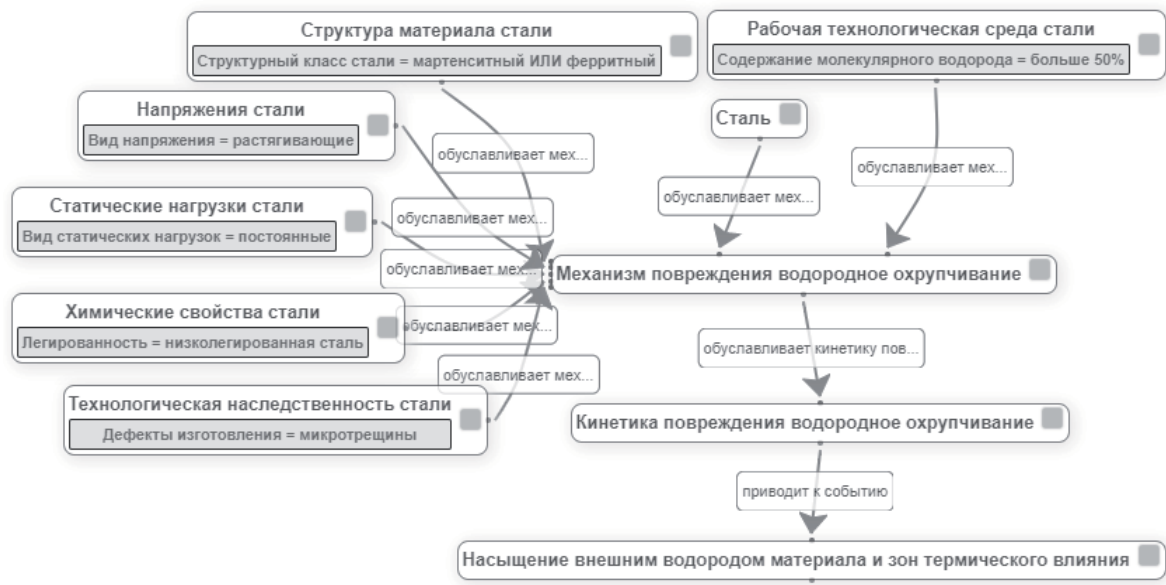


Рисунок 6 - Фрагмент полученной онтологической модели (визуальный редактор онтологий KBDS)

Средствами KBDS сформированная онтологическая модель может быть экспортирована в формат OWL. Фрагмент сгенерированного OWL-кода (описание классов, объектных свойств и свойств значений), соответствующего рисункам 3-5, представлен ниже.

```

<owl:Class rdf:about="ВоздействующийФактор" />
<owl:Class rdf:about="МеханическиеНагрузки">
  <rdfs:subClassOf rdf:resource="ВоздействующийФактор" />
</owl:Class>
<owl:Class rdf:about="ДеградационныйПроцесс" />
<owl:Class rdf:about="МеханизмПовреждения">
  <rdfs:subClassOf rdf:resource="ДеградационныйПроцесс" />
</owl:Class>
...
<owl:ObjectProperty rdf:about="Происходит">
  <rdfs:domain rdf:resource="#ИсходноеСостояние" />
  <rdfs:range rdf:resource="#ДеградационныйПроцесс" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="ОбуславливаетМеханизмПовреждения">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="ВидМатериала" />
        <rdf:Description rdf:about="ХимическиеСвойства" />
        <rdf:Description rdf:about="СтруктураМатериала" />
        <rdf:Description rdf:about="ТехнологическаяНаследственность" />
        <rdf:Description rdf:about="РабочаяТехнологическаяСреда" />
        <rdf:Description rdf:about="СтатическиеНагрузки" />
        <rdf:Description rdf:about="Напряжения" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#МеханизмПовреждения" />
</owl:ObjectProperty>
...
<owl:DatatypeProperty rdf:about="СодержаниеМолекулярногоВодорода">
  <rdfs:domain rdf:resource="#РабочаяТехнологическаяСреда" />
  <rdfs:range rdf:resource="xsd:string" />
</owl:DatatypeProperty>
...
<МеханизмПовреждения rdf:about="МеханизмПоврежденияВодородноеОхрупчивание">
  <ОбуславливаетКинетикуПовреждения
rdf:resource="#КинетикаПоврежденияВодородноеОхрупчивание" />
</МеханизмПовреждения>
<КинетикаПовреждения rdf:about="КинетикаПоврежденияВодородноеОхрупчивание">
  <ПриводитКСобытию
rdf:resource="#НасыщениеВнешнимВодородомМатериалаИЗонТермическогоВлияния" />
</КинетикаПовреждения>

```

**Пример этапов 2-3.** Анализ и интеграция онтологической модели с правилами, визуализация и модификация полученных продукций. В зависимости от типа элемента CIM понятия преобразовываются в факты и правила для дальнейших модификаций пользователем в RVML-редакторе (см. рисунки 7 и 8).

**Пример этапа 4.** Формирование PSM и генерация кода с использованием PKBD включает, с одной стороны, адаптацию диаграмм RVML к CLIPS и генерацию программного кода CLIPS, с другой стороны – синтез спецификаций ИС (см. рисунок 9) для интерпретатора, который обеспечивает генерацию пользовательского интерфейса для создания, чтения, обновления и удаления (CRUD) элементов производственной БЗ (см. рисунок 10) и взаимодействия программных компонентов.

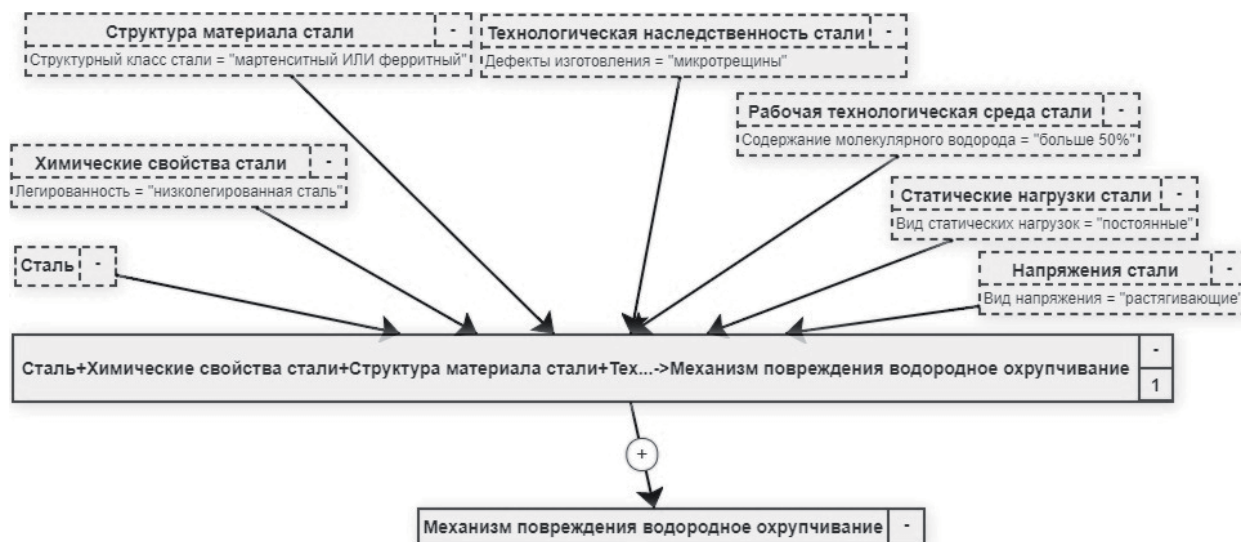


Рисунок 7 - Пример RVML-диаграммы (факты)



Рисунок 8 - Пример RVML-диаграммы (правило)

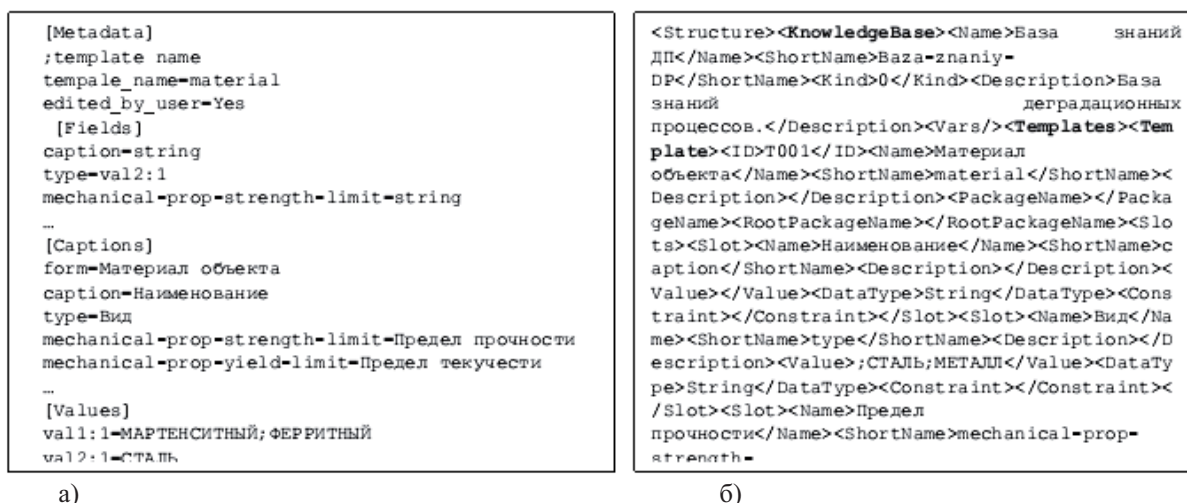


Рисунок 9 - Пример сгенерированных спецификаций: а) типизированный текстовый файл, б) XML-файл

**Пример этапа 5.** Проверка разработанных БЗ и ИС выполняется экспертом с помощью логических правил вывода (см. рисунок 11). Возможен возврат к одному из предыдущих этапов в соответствии с результатами тестирования.

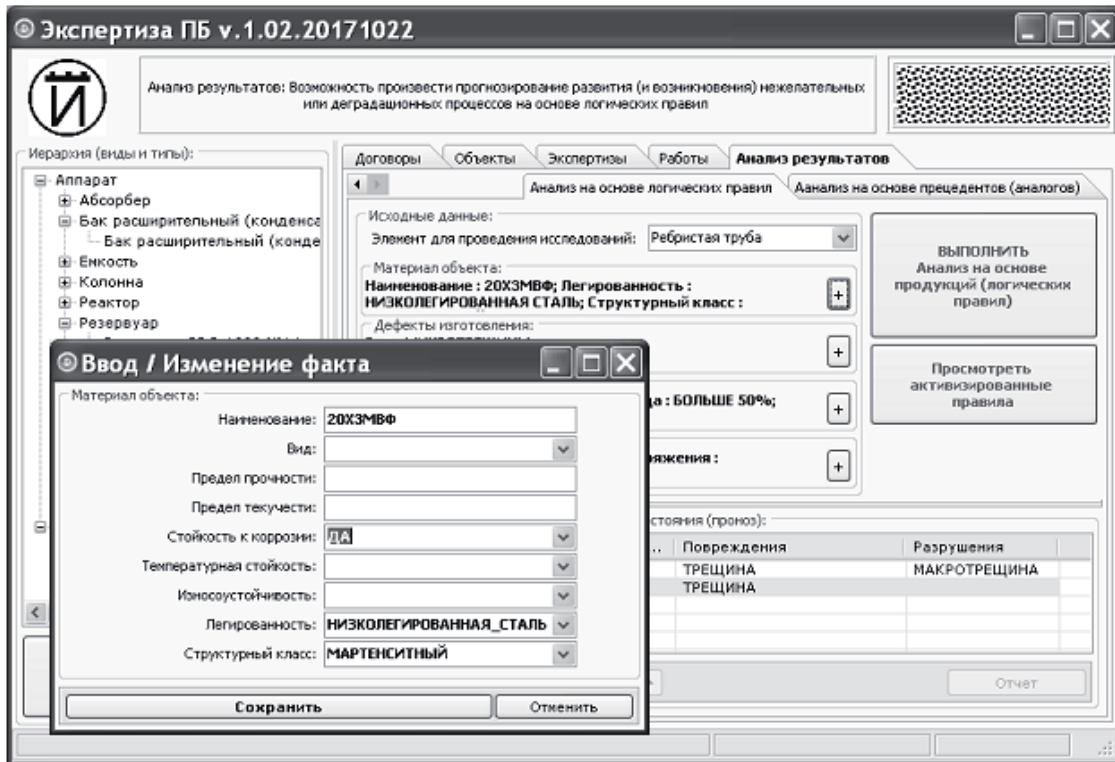


Рисунок 10 - Пример формы, динамически создаваемой на основе спецификаций в программной системе «Экспертиза»

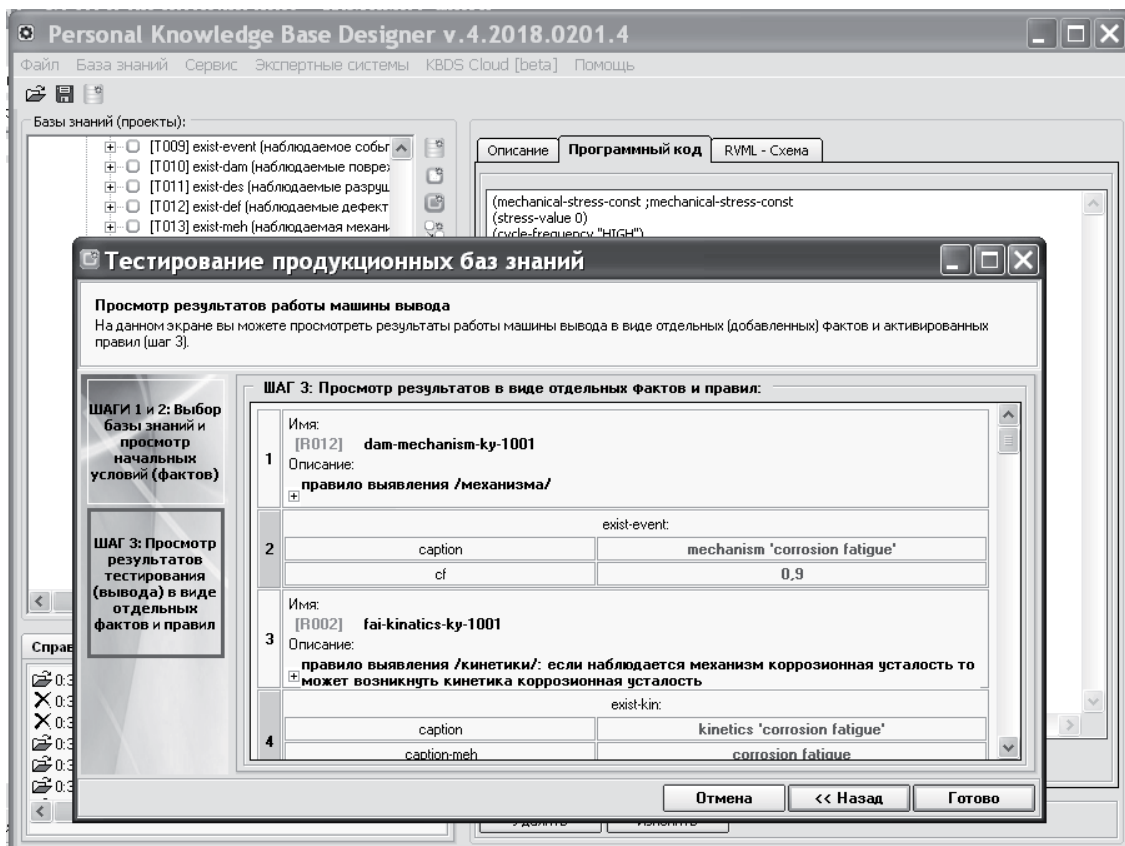


Рисунок 11 - Пример пользовательского интерфейса РКВД: просмотр активированных правил

## Заключение

В статье описан подход к разработке проблемно-ориентированных ИС диагностического типа с использованием онтологий. Предлагаемый подход основан на принципах MDE/MDD/MDA и их модификации в контексте инженерии знаний, на использовании онтологий в качестве вычислительно-независимой модели. Процесс создания онтологии автоматизируется путём анализа и преобразования концептуальных моделей ПрО в виде концепт-карт, представленных в формате XML, с последующей возможностью загрузки сформированных онтологий в формат OWL. В процессе разработки моделей используется RVML, а в качестве целевых платформ – CLIPS и PKBD. Подход позволяет охватить все этапы создания ИС и обеспечивает полноту процесса разработки: от моделей ПрО до программных кодов.

Благодаря использованию онтологий достигается значительное сокращение затрат времени на этапах реализации и устранения ошибок программирования (вследствие автоматической генерации кода), концептуализации и формализации предметных знаний. В качестве средств реализации использованы прототипы программных средств: веб-ориентированной программной системы – KBDS и редактора – PKBD, взаимодействующие через API.

Предложенный подход и ПО были применены при разработке экспериментальной версии проблемно-ориентированной диагностической ИС для определения причин повреждения и разрушения конструкционных материалов в нефтехимии.

## Благодарности

Работа выполнена при частичной поддержке грантов РФФИ 18-37-00006, 18-29-03039.

## Список источников

- [1] *Ноженкова, Л.Ф.* Создание комплексной системы безопасности региона на основе системной интеграции технологий / Л.Ф. Ноженкова, В.В. Ничепорчук, А.И. Ноженков // Информатизация и связь. – 2013. – №2. – С.122-124.
- [2] *Массель, Л.В.* Технологии и инструментальные средства интеллектуальной поддержки принятия решений в экстремальных ситуациях в энергетике / Л.В. Массель, А.Г. Массель // Вычислительные технологии. – 2013. – Т.18. Специальный выпуск. – С.37-44.
- [3] *Балута, В.И.* Среда моделирования, прогнозирования и экспертиз как интеллектуальное ядро поддержки управления сложными системами / В.И. Балута, В.П. Осипов, О.Ю. Яковенко // Препринты ИПМ им. М.В.Келдыша. – 2015. – № 82. – 16 с.
- [4] *Berman, A.F.* A methodology for the investigation of the reliability and safety of unique technical systems / A.F. Berman, O.A. Nikolaychuk, A.Yu. Yurin, A.I. Pavlov // Part O: Journal of Risk and Reliability. – 2014. – Vol. 228, No. 1. – P.29-38.
- [5] *Ebrahim, A.S.* A structured approach for the diagnosis of formation damage caused by organic scale deposits and surface active agents, Part II: Expert system development / A.S. Ebrahim, A.A. Garrouch, H.M.S. Lababidi // Journal of Petroleum Science and Engineering. – 2016. – Vol. 138. – P.245-252.
- [6] *Starr, R.R.* Concept maps as the first step in an ontology construction method / R.R. Starr, J.M. Parente de Oliveira // Information Systems. – 2013. – Vol. 38. – P.771-783.
- [7] *Gavrilova, T.A.* Ontology design and individual cognitive peculiarities: A pilot study / T.A. Gavrilova, I.A. Leshcheva // Expert Systems with Applications. – 2015. – Vol. 42. – P.3883-3892.
- [8] *Herrero-Zazo, M.* Conceptual models of drug-drug interactions: A summary of recent efforts / M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez // Knowledge-Based Systems. – 2016. – Vol. 114. – P.99-107.
- [9] *Friedman-Hill, E.* Jess in Action: Rule-based Systems in Java / Friedman-Hill E // Manning. – 2003. – 443 p.
- [10] *Eriksson, H.* The JessTab Approach to Protégé and Jess Integration / H. Eriksson // In Proceedings of the IFIP 17th World Computer Congress – TC12 Stream on Intelligent Information Processing, Kluwer, B.V., Dordrecht. – 2002. – P.237-248.

- [11] SWRL: A Semantic Web Rule Language Combining OWL and RuleML. – <https://www.w3.org/Submission/SWRL/>.
- [12] SWRLtab plugin. – <https://protegewiki.stanford.edu/wiki/SWRLTab>.
- [13] **Загорюлько, Ю.А.** Семантическая технология разработки интеллектуальных систем, ориентированная на экспертов предметной области / А.Ю. Загорюлько // Онтология проектирования. – 2015. – Т.5. - №1 (15). – С.30-46.
- [14] **Голенков, В.В.** Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н.А. Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2011). – Минск: БГУИР. – 2011. – С.21-58.
- [15] **Копайгородский, А.Н.** Применение онтологий в семантических информационных работах / А.Н. Копайгородский // Онтология проектирования. – 2014. – № 4 (14). – С.78-89.
- [16] **Дубинин, В.Н.** Проектирование управляющих приложений на основе трансформации онтологий с использованием языков логического программирования / В.Н. Дубинин, В.В. Вяткин // Труды Международной научно-технической конференции «Современные информационные технологии». – 2012. – вып. 16. – С.6-25.
- [17] **Brizzi, P.** Towards an ontology driven approach for systems interoperability and energy management in the smart city / P. Brizzi, D. Bonino, A. Musetti, A. Krylovskiy, E. Patti, M. Axling // In Processing of the International Multidisciplinary Conference on Computer and Energy Science (SpliTech). – 2016. – P.1-7.
- [18] **Chungoora, N.** A model-driven ontology approach for manufacturing system interoperability and knowledge sharing / N. Chungoora, R.I. Young, G. Gunendran, C. Palmer, Z. Usman, N.A. Anjum, A.-F. Cutting-Decellec, J.A. Harding, K. Case // Computers in Industry. – 2013. – Vol. 64, No. 4. – P.392-401.
- [19] **Katasonov, A.** Ontology-driven software engineering: beyond model checking and transformations / A. Katasonov // International Journal of Semantic Computing. – 2012. – Vol. 6, No. 2. – P.205-242. – DOI: 10.1142/s1793351x12500031.
- [20] **Thaddeus, S.** A Semantic Web Tool for Knowledge-based Software Engineering / S. Thaddeus, S.V. Kasmir Raja // In Proceedings of the 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006). – 2006. – P.337-342.
- [21] **Tanaka, Y.** Developing Design Support System Based on Semantic of Design Model / Y. Tanaka, K. Tsuda // In Proceedings of the 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016. Procedia Computer Science. – 2016. – Vol. 96. – P.1231-1239. – DOI: 10.1016/j.procs.2016.08.167.
- [22] **Chen, R.-C.** A recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection / R.-C. Chen, Y.-H. Huang, C.-T. Bau, S.-M. Chen // Expert Systems with Applications. – 2012. – Vol. 39, No. 4. – P.3995-4006. – DOI: 10.1016/j.eswa.2011.09.061.
- [23] **Rajput, Q.** Ontology Based Expert-System for Suspicious Transactions Detection / Q. Rajput, N.S. Khan, A. Larik, S. Haider // Computer and Information Science. – 2014. – Vol. 7, No. 1. – P.103-114. – DOI: 10.5539/cis.v7n1p103.
- [24] **Shue, L.** The development of an ontology-based expert system for corporate financial rating / L. Shue, C. Chen, W. Shiue // Expert Systems with Applications. – 2009. – Vol. 36. – P.2130-2142. – DOI: 10.1016/j.eswa.2007.12.044.
- [25] **Nofal, M.** Developing Web-Based Semantic Expert Systems / M. Nofal, K.M. Fouad // International Journal of Computer Science. – 2014. – Vol. 11, No. 1. – P.103-110.
- [26] **Dong, M.** Ontology-based service product configuration system modeling and development / M. Dong, D. Yang, L. Su // Expert Systems with Applications. – 2011. – Vol. 38, No. 9. – P.11770-11786. – DOI: 10.1016/j.eswa.2011.03.064.
- [27] **Afacan, Y.** An ontology-based universal design knowledge support system / Y. Afacan, H. Demirkan // Knowledge-Based Systems. – 2011. – Vol. 24, No. 4. – P.530-541. – DOI: 10.1016/j.knosys.2011.01.002.
- [28] **Macek, K.** Ontology-Driven Design of an Energy Management System / K. Macek, K. MaĀík, P. Stluka // In Proceedings of the 21st European Symposium on Computer Aided Process Engineering, 1st Edition. – 2011. – Vol. 29. – P.2009-2014. – DOI: 10.1016/b978-0-444-54298-4.50180-x.
- [29] **Arsene, O.** Medicine expert system dynamic Bayesian Network and ontology based / O. Arsene, I. Dumitrache, I. MiĀu // Expert Systems with Applications. – 2011. – Vol. 38, No. 12. – P.15253-15261. – DOI: 10.1016/j.eswa.2011.05.074.
- [30] **Parreiras, F.S.** Using ontologies with UML class-based modeling: The TwoUse approach / F.S. Parreiras, S. Staab // Data & Knowledge Engineering. – 2010. – Vol. 69, No. 11. – P.1194-1207. – DOI: 10.1016/j.datak.2010.07.009.



- [31] *Cretu, L.G.* Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice / L.G. Cretu, D. Florin // Apple Academic Press. – 2014. – DOI: 10.1201/b17480.
- [32] *da Silva A.R.* Model-driven engineering: A survey supported by the unified conceptual model / A.R. da Silva // Computer Languages, Systems & Structures. – 2015. – Vol. 43. – P.139-155. – DOI: 10.1016/j.cl.2015.06.001.
- [33] *Sami, B.* Model-Driven Software Development / B. Sami, B. Book, V. Gruhn // Springer. – 2005. – DOI: 10.1007/3-540-28554-7.
- [34] *Kleppe, A.* MDA Explained: The Model Driven Architecture: Practice and Promise, 1st ed. / A. Kleppe, J. Warmer, W. Bast // Addison-Wesley. – 2003.
- [35] *Djurić, D.* Ontology Modeling and MDA / D. Djurić, D. Gašević, V. Devedžić // Journal of Object technology. – 2005. – Vol. 4, No. 1. – P.109-128. – DOI: 10.5381/jot.2005.4.1.a3.
- [36] *Staab, S.* Model Driven Engineering with Ontology Technologies / S. Staab, T. Walter, G. Gröner, F.S. Parreiras // Reasoning Web. Lecture Notes in Computer Science. Springer Berlin Heidelberg. – 2010. – Vol. 6325. – P.62-98. – DOI: 10.1007/978-3-642-15543-7\_3.
- [37] CLIPS: A Tool for Building Expert Systems. – <http://www.clipsrules.sourceforge.net>.
- [38] *Юрин, А.Ю.* Web-сервис для автоматизированного формирования продукционных баз знаний на основе концептуальных моделей / А.Ю. Юрин, Н.О. Дородных // Программные продукты и системы. – 2014. – №4. – С.103-107.
- [39] *Дородных, Н.О.* Система программирования продукционных баз знаний: Personal Knowledge Base Designer / Н.О. Дородных, М.А. Грищенко, А.Ю. Юрин // Открытые семантические технологии проектирования интеллектуальных систем. – 2016. – №6. – С.209-212.
- [40] *Юрин, А.Ю.* Редактор баз знаний в формате CLIPS / А.Ю. Юрин А.Ю., М.А. Грищенко // Программные продукты и системы. – 2012. – №4. – С.83-87.
- [41] *Джексон, П.* Введение в экспертные системы. Пер. с англ. / П. Джексон – М: Вильямс. – 2001. – 624 с.
- [42] *Николайчук, О.А.* Моделирование знаний для исследования динамики технического состояния уникальных объектов / О.А. Николайчук // Проблемы управления. – 2009. - № 4. - С.58–65.
- [43] OWL 2 Web Ontology Language, Document Overview (Second Edition). – <https://www.w3.org/TR/owl2-overview/>.
- [44] *Буч, Г.* Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. / Г. Буч, Д. Рамбо, И. Якобсон. – М.: ДМК Пресс. – 2006. – 496 с.
- [45] *Юрин, А.Ю.* Нотация для проектирования баз знаний продукционных экспертных систем / А.Ю. Юрин // Объектные системы. – 2016. – №12. – С.48-54.
- [46] *Берман, А.Ф.* Поддержка принятия решений на основе продукционного подхода при проведении экспертизы промышленной безопасности / А.Ф. Берман, О.А. Николайчук, А.Ю. Юрин, К.А. Кузнецов // Химическое и нефтегазовое машиностроение. – 2014. – №11. – С.28-35.
- [47] *Czarnecki, K.* Feature-based survey of model transformation approaches / K. Czarnecki, S. Helsen // IBM Systems Journal. – 2006. – Vol. 45, No. 3. – P.621-645. – DOI: 10.1147/sj.453.0621.
- [48] *Бычков, И.В.* Подход к разработке программных компонентов для формирования баз знаний на основе концептуальных моделей / И.В. Бычков, Н.О. Дородных, А.Ю. Юрин // Вычислительные технологии. – 2016. – Т. 21. – № 4. – С.16-36.
- [49] *Берман, А.Ф.* Деграадация механических систем / А.Ф. Берман. – Новосибирск: Наука. – 1998. – 320 с.
- [50] *Берман, А.Ф.* Интеллектуальная информационная система анализа отказов / А.Ф. Берман, О.А. Николайчук, А.Ю. Юрин // Проблемы машиностроения и надежности машин. – 2012. – № 4. – С.88-96.

## ONTOLOGY-BASED DEVELOPMENT OF DIAGNOSTIC INTELLIGENT SYSTEMS

М.А. Grischenko<sup>1,a</sup>, Н.О. Dorodnykh<sup>2,b</sup>, С.А. Korshunov<sup>1,c</sup>, А.Ю. Yurin<sup>2,d</sup>

<sup>1</sup>CentraSib LLC, Irkutsk, Russia,

<sup>2</sup>Matrosov Institute for System Dynamics and Control Theory of SB RAS, Irkutsk, Russia,

<sup>a</sup>makcmg@icc.ru, <sup>b</sup>tualatin32@mail.ru, <sup>c</sup>grey.for@gmail.com, <sup>d</sup>iskander@icc.ru

### Abstract

Development of problem-oriented intelligent systems for solving semistructured problems remains a topical problem. The solution efficiency of this problem can be increased by improving approaches (methodologies) or creating special-

ized software. The paper describes an approach for the ontology-based development of diagnostic intelligent systems (including knowledge bases). A model-driven approach is used as a methodological basis with a subsequent redefinition of the main stages, models, transformation rules and means of implementation. The proposed approach is used for development of diagnostic systems in petrochemistry. Diagnosis is considered as the definition of the object's state, including: monitoring the state, finding a location and determining the causes of a failure, forecasting the state. The objects to be diagnosed are presented as a sequence (chain) of states, where the state is a set of parameters and their values at a given time. The features of the proposed approach implementation are the following: the use of ontology as a computation independent model that describes the features of the problem; the use of the Rule Visual Modeling Language to present the platform independent and platform dependent models; the use of C Language Integrated Production System as a programming language and a target platform. An example of the approach application by means of software prototypes called Knowledge Base Development System and Personal Knowledge Base Designer is described.

**Key words:** *model-driven approach, intelligent system, knowledge base, ontology, rules, conceptual model.*

**Citation:** *Grischenko MA, Dorodnykh NO, Korshunov SA, Yurin AY. Ontology-based development of diagnostic intelligent systems [In Russian]. Ontology of designing. 2018; 8(2): 265-284. - DOI: 10.18287/2223-9537-2018-8-2-265-284.*

## Acknowledgment

The work is partially supported by RFBR grant 18-37-00006, 18-29-03039.

## References

- [1] *Nozhenkova LF, Nicheporchuk VV, Nozhenkov A.* Creating the comprehensive regional safety system based on system integration of technologies [In Russian]. *Informatization and communication.* 2013; 2: 122-124.
- [2] *Massel LV, Massel AG.* Technologies and tools for intelligent decision-making support in extreme situations in the energy sector [In Russian]. *Computational Technologies.* 2013; 13(S1): 37-44.
- [3] *Baluta VI, Osipov VP, Yakovenko OY.* The environment of modeling, forecasting and expertise as the intellectual core of support management complex systems [In Russian]. *Moscow: KIAM Preprint.* 2015; 82.
- [4] *Berman AF, Nikolaychuk OA, Yurin AY, Pavlov AI.* A methodology for the investigation of the reliability and safety of unique technical systems. Part O: *Journal of Risk and Reliability.* 2014; 228(1): 29-38.
- [5] *Ebrahim AS, Garrouch AA, Lababidi HMS.* A structured approach for the diagnosis of formation damage caused by organic scale deposits and surface active agents, Part II: Expert system development. *Journal of Petroleum Science and Engineering.* 2016; 138: 245-252.
- [6] *Starr RR, Parente de Oliveira JM.* Concept maps as the first step in an ontology construction method. *Information Systems.* 2013; 38: 771-783.
- [7] *Gavrilova TA, Leshcheva IA.* Ontology design and individual cognitive peculiarities: A pilot study. *Expert Systems with Applications.* 2015; 42: 3883-3892.
- [8] *Herrero-Zazo M, Segura-Bedmar I, Martínez P.* Conceptual models of drug-drug interactions: A summary of recent efforts. *Knowledge-Based Systems.* 2016; 114: 99-107.
- [9] *Friedman-Hill E.* *Jess in Action: Rule-based Systems in Java.* Manning; 2003.
- [10] *Eriksson H.* The JessTab Approach to Protégé and Jess Integration. In *Proceedings of the IFIP 17th World Computer Congress – TC12 Stream on Intelligent Information Processing.* Dordrecht; 2002: 237-248.
- [11] SWRL: A Semantic Web Rule Language Combining OWL and RuleML. – <https://www.w3.org/Submission/SWRL/>.
- [12] SWRLtab plugin. – <https://protegewiki.stanford.edu/wiki/SWRLTab>.
- [13] *Zagorulko YA.* Semantic technology for development of intelligent systems oriented to experts in subject domain [In Russian]. *Ontology of Designing.* 2015; 5(1): 30-46.
- [14] *Golenkov VV, Gulyakina NA.* Principles of construction of mass semantic technology component of intelligent systems [In Russian]. In *Proceedings of the International Scientific and Technical Conference on Open semantic technologies of intelligent systems (OSTIS-2011).* Minsk, BGUIR; 2011: 21-58.
- [15] *Kopyagorodsky AN.* Use of ontologies in semantic information systems [In Russian]. *Ontology of Designing.* 2014; 4(14): 78-89.
- [16] *Dubinina VN, Vyatkin VV.* Designing control applications based on the transformation of ontologies using logical programming languages. In *Proceedings of the International Scientific and Technical Conference on Contemporary information technologies.* 2012; 16: 6-25.

- [17] **Brizzi P, Bonino D, Musetti A, Krylovskiy A.** Towards an ontology driven approach for systems interoperability and energy management in the smart city. In Processing of the International Multidisciplinary Conference on Computer and Energy Science (SpliTech); 2016: 1-7.
- [18] **Chungoora N, Young RI, Gunendran G, Palmer C, Usman Z, Anjum NA, Cutting-Decellec AF, Harding JA, Case K.** A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. Computers in Industry. 2013; 64(4): 392-401. – DOI: 10.1016/j.compind.2013.01.003.
- [19] **Katasonov A.** Ontology-driven software engineering: beyond model checking and transformations. International Journal of Semantic Computing. 2012; 6(2): 205-242. – DOI: 10.1142/s1793351x12500031.
- [20] **Thaddeus S, Kasmir Raja SV.** A Semantic Web Tool for Knowledge-based Software Engineering. In Proceedings of the 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006); 2006: 337-342.
- [21] **Tanaka Y, Tsuda K.** Developing Design Support System Based on Semantic of Design Model. In Proceedings of the 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems (KES2016). Procedia Computer Science. 2016; 96: 1231-1239. – DOI: 10.1016/j.procs.2016.08.167.
- [22] **Chen RC, Huang YH, Bau CT, Chen SM.** A recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection. Expert Systems with Applications. 2012; 39(4): 3995-4006. – DOI: 10.1016/j.eswa.2011.09.061.
- [23] **Rajput Q, Khan NS, Larik A, Haider S.** Ontology Based Expert-System for Suspicious Transactions Detection. Computer and Information Science. 2014; 7(1): 103-114. – DOI: 10.5539/cis.v7n1p103.
- [24] **Shue L, Chen C.** The development of an ontology-based expert system for corporate financial rating. Expert Systems with Applications. 2009; 36: 2130-2142. – DOI: 10.1016/j.eswa.2007.12.044.
- [25] **Nofal M, Fouad KM.** Developing Web-Based Semantic Expert Systems. International Journal of Computer Science. 2014; 11(1): 103-110.
- [26] **Dong M, Yang D, Su L.** Ontology-based service product configuration system modeling and development. Expert Systems with Applications. 2011; 38(9): 11770-11786. – DOI: 10.1016/j.eswa.2011.03.064.
- [27] **Afacan Y, Demirkan H.** An ontology-based universal design knowledge support system. Knowledge-Based Systems. 2011; 24(4): 530-541. – DOI: 10.1016/j.knosys.2011.01.002.
- [28] **Macek K, MaĀik K, Stluka P.** Ontology-Driven Design of an Energy Management System. In Proceedings of the 21st European Symposium on Computer Aided Process Engineering, 1st Edition. 2011; 29: 2009-2014. – DOI: 10.1016/b978-0-444-54298-4.50180-x.
- [29] **Arsene O, Dumitrache I, Mihu I.** Medicine expert system dynamic Bayesian Network and ontology based. Expert Systems with Applications. 2011; 38(12): 15253-15261. – DOI: 10.1016/j.eswa.2011.05.074.
- [30] **Parreiras FS, Staab S.** Using ontologies with UML class-based modeling: The TwoUse approach. Data & Knowledge Engineering. 2010; 69(11): 1194-1207. – DOI: 10.1016/j.datak.2010.07.009.
- [31] **Cretu LG, Florin D.** Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice. Apple Academic Press; 2014. – DOI: 10.1201/b17480.
- [32] **da Silva AR.** Model-driven engineering: A survey supported by the unified conceptual model. Computer Languages, Systems & Structures. 2015; 43: 139-155. – DOI: 10.1016/j.cl.2015.06.001.
- [33] **SamĀ B, Book B, Gruhn V.** Model-Driven Software Development. Springer; 2005. – DOI: 10.1007/3-540-28554-7.
- [34] **Kleppe A, Warmer J, Bast W.** MDA Explained: The Model Driven Architecture: Practice and Promise, 1st ed. Addison-Wesley; 2003.
- [35] **DjuriĀ D, GaševiĀ D, DevedĀiĀ V.** Ontology Modeling and MDA. Journal of Object technology. 2005; 4(1): 109-128. – DOI: 10.5381/jot.2005.4.1.a3.
- [36] **Staab S, Walter T, Gröner G, Parreiras FS.** Model Driven Engineering with Ontology Technologies. Reasoning Web. Lecture Notes in Computer Science. Springer Berlin Heidelberg. 2010; 6325: 62-98. – DOI: 10.1007/978-3-642-15543-7\_3.
- [37] CLIPS: A Tool for Building Expert Systems. – <http://www.clipsrules.sourceforge.net>.
- [38] **Yurin AY, Dorodnykh NO.** A web-service for knowledge base generation on the basis of conceptual models [In Russian]. Software & Systems. 2014; 4: 103-107.
- [39] **Dorodnykh NO, Grischenko MA, Yurin AY.** Software for rule knowledge bases design: Personal Knowledge Base Designer [In Russian]. Open Semantic Technologies for Intelligent Systems. 2016; 6: 209-212.
- [40] **Yurin AY, Grischenko MA.** Knowledge base editor for CLIPS [In Russian]. Software & Systems. 2102; 4: 83-87.
- [41] **Jackson P.** Introduction to Expert Systems. - Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA; 1998.
- [42] **Nikolaychuk OA.** Modeling of knowledge for the study of the dynamics of the technical condition of unique objects [In Russian]. Problems of Control. 2009; 4: 58-65.
- [43] OWL 2 Web Ontology Language, Document Overview (Second Edition). – <https://www.w3.org/TR/owl2-overview/>.

- [44] **Buch G, Rambo D, Yakobson I.** UML language. User Guide. 2nd ed. [In Russian]. Moscow: DMK Press; 2006.
- [45] **Yurin AY.** Notation for design of knowledge bases of rule-based expert systems [In Russian]. Object Systems. 2016; 12: C.48-54.
- [46] **Berman AF, Nikolaychuk OA, Yurin AY, Kuznetsov KA.** Support of decision making based on production approach during industrial safety review [In Russian]. Chemical and Petroleum Engineering. 2014; 11: 28-35.
- [47] **Czarnecki K, Helsen S.** Feature-based survey of model transformation approaches. IBM Systems Journal. 2006; 45(3): 621-645. – DOI: 10.1147/sj.453.0621.
- [48] **Bychkov IV, Dorodnykh NO, Yurin AY.** Approach to the development of software components for generation of knowledge bases based on conceptual models [In Russian]. Computational Technologies. 2016; 21(4): 16-36.
- [49] **Berman AF.** Degradation of mechanical systems [In Russian]. Novosibirsk: Science; 1998.
- [50] **Berman AF, Nikolaychuk OA, Yurin AY.** Intellectual data system for analyzing failures [In Russian]. Journal of Machinery Manufacture and Reliability. 2012; 4: 88-96.

### Сведения об авторах



**Грищенко Максим Андреевич**, 1987 г. рождения. Окончил Иркутский национальный исследовательский технический университет (ИрННТУ) в 2010 году. Программист ООО «ЦентраСиб». Автор 15 печатных работ. Сфера научных интересов: базы знаний, системы поддержки принятия решений, экспертные системы.

**Maxim Andreevich Grishchenko** (b. 1987) graduated from Irkutsk National Research Technical University (INRTU) in 2010. He is a programmer of CenterSib LLC and a co-author of 15 publications in the field of knowledge bases and intelligent decision support systems.

**Дородных Никита Олегович**, 1990 г. рождения. Окончил ИрННТУ в 2012 г. Младший научный сотрудник Института динамики систем и теории управления им. В.М. Матросова СО РАН (ИДСТУ СО РАН). В списке научных трудов около 45 работ в области автоматизации разработки интеллектуальных систем и баз знаний, получения знаний на основе трансформации концептуальных моделей.

**Nikita Olegovich Dorodnykh** (b. 1990) graduated from INRTU in 2012. He is an associate researcher of Matrosov Institute for System Dynamics and Control Theory of SB RAS (ISDCT SB RAS). Co-author of about 45 publications in the field of computer-aided development of intelligent systems and knowledge bases, knowledge acquisition based on the transformation of conceptual models.



**Коршунов Сергей Андреевич**, 1991 г. рождения. Окончил ИрННТУ в 2012 г. Программист ООО «ЦентраСиб». Автор 17 печатных работ. Сфера научных интересов: базы знаний, онтологии, системы имитационного моделирования, визуализация.

**Sergey Andreevich Korshunov** (b. 1991) graduated from the INRTU in 2012. He is a programmer of CenterSib LLC. Author of 17 publications in the field of ontologies, simulation modeling systems and visualization.

**Юрин Александр Юрьевич**, 1980 г. рождения. Окончил ИрННТУ в 2002 г., к.т.н. (2005). Заведующий лабораторией ИДСТУ СО РАН, доцент кафедры автоматизированных систем Института высоких технологий ИрННТУ. Автор 72 печатных работ. Проводит исследования в области разработки систем поддержки принятия решений, экспертных систем и баз знаний, использования прецедентного подхода и семантических технологий при проектировании интеллектуальных диагностических систем, обеспечения надежности и безопасности сложных технических систем.

**Alexander Yurievich Yurin** (b.1980) graduated from the INRTU in 2002, PhD (2005). He is Head of a laboratory at ISDCT SB RAS and associate professor of the Department of Automated Systems of the Institute of High Technologies of INRTU. Co-author 72 publications. He conducts research in the field of development of decision support systems, expert systems and knowledge bases, application of the case-based reasoning and semantic technologies in the design of diagnostic intelligent systems, maintenance of reliability and safety of complex technical systems.

