

УЧРЕЖДЕНИЕ РОССИЙСКОЙ АКАДЕМИИ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
**ИНСТИТУТ ДИНАМИКИ СИСТЕМ И ТЕОРИИ УПРАВЛЕНИЯ имени
В.М. МАТРОСОВА
(ИДСТУ СО РАН)**

На правах рукописи

Юрин Александр Юрьевич

**Методы и программные средства создания интеллектуальных систем
с декларативными базами знаний на основе модельных трансформаций**

Специальность 05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
доктора технических наук

Иркутск – 2022

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук (ИДСТУ СО РАН)

Научный консультант: **Бычков Игорь Вячеславович** – академик РАН, доктор технических наук, профессор, Институт динамики систем и теории управления имени В.М. Матросова СО РАН, директор

Официальные оппоненты: **Грибова Валерия Викторовна** – доктор технических наук, профессор, ИАПУ ДВО РАН, заместитель директора по научной работе

Федоров Максим Валериевич – кандидат физико-математических наук, доктор химических наук, профессор, НТУ «Сириус», ректор

...

Ведущая организация: Федеральный исследовательский центр информационных и вычислительных технологий (ФИЦ ИВТ), г. Новосибирск

Защита состоится «__» _____ 2022 г. в ____ часов на заседании диссертационного совета Д 003.021.01 при Федеральном государственном бюджетном учреждении науки Институте динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук (ИДСТУСО РАН) по адресу: 664033, г. Иркутск, ул. Лермонтова, 134.

С диссертацией можно ознакомиться в библиотеке и на официальном сайте www.idstu.irk.ru ИДСТУ СО РАН.

Автореферат разослан «__» _____ 2022 г.

Ученый секретарь
диссертационного совета,
к.ф.-м.н.

Т.В. Груздева

Введение

Актуальность. Одним из направлений повышения эффективности и надежности процессов обработки и передачи данных и знаний в вычислительных машинах, комплексах и компьютерных сетях является переход к передовым цифровым, интеллектуальным производственным технологиям, роботизированным системам, новым материалам и способам конструирования, создание систем обработки больших объемов данных (Big Data), машинного обучения и искусственного интеллекта (ИИ), что, в свою очередь, требует создания нового алгоритмического и программного обеспечения¹.

Современный уровень исследований в данных областях достаточно высок, однако проблема повышения эффективности и качества разработки программного обеспечения систем ИИ, включая базы знаний (БЗ), сохраняет свою актуальность, поскольку данный процесс остается трудоемким и требует привлечения и обеспечения взаимодействия специалистов различных специализаций и квалификаций, в том числе, профессиональных программистов. Для решения поставленной проблемы предлагаются новые языки и инструментальные программные средства, методы и технологии разработки. При этом перспективным является более полное включение в процесс создания систем ИИ конечных пользователей (end users) с передачей им отдельных функций, которые исторически реализовывались программистами. Подходы подобной направленности объединены в направление, известное как End-User Development (EUD)², основная идея которого – предоставить возможность конечному пользователю самому создавать и настраивать приложение. В рамках этого направления многообещающими способами спецификации предметной области и бизнес-логики приложений являются³: использование принципов визуального программирования, предметно/проблемно-ориентированных языков, а также модельных трансформаций с целью автоматизированной генерации программных кодов и спецификаций.

Значительный вклад в разработку и исследование моделей, методов и средств создания интеллектуальных систем, включая онтологии, БЗ и программные средства для их проектирования, внесли Абрамова Н.К., Аверкин А.Н., Баадер Ф., Берман А.Ф., Бычков И.В., Вагин В.Н., Ван Хармелен Ф., Варшавский П.Р., Васильев С.Н., Гаврилова Т.А., Голенков В.В., Грау Б., Грибова В.В., Грубер Т., Гуарино Н., Джарратано Дж., Джексон П., Еремеев А.П., Желтов С.Ю., Загорулько Ю.А., Клещев А.С., Колесников А.В., Кудрявцев Д.В., Люгер Г., Массель Л.В., МакГиннесс Д., Мешалкин В.П., Мотик Б., Николайчук О.А., Ноженкова Л.Ф., Норвиг П., Осипов Г.С., Осуга С., Патель-Шнайдер П., Попов Э.В., Поспелов Д.А., Райли Г., Рассел С., Рыбина Г.В., Стааб С., Смирнов Б.В., Смирнов С.В., Федоров

¹ Указ Президента РФ от 01.12.2016 N 642 (ред. от 15.03.2021) О Стратегии научно-технологического развития Российской Федерации [Электронный ресурс]. – Режим доступа: <https://sudact.ru/law/ukaz-prezidenta-rf-ot-01122016-n-642/strategiia-nauchno-tekhnologicheskogo-razvitiia-rossiiskoi-federatsii/> (дата обращения: 18.03.2022).

² Barricelli B.R. End-user development, end-user programming and end-user software engineering: A systematic mapping study / B.R. Barricelli, F. Cassano, D. Fogli, A. Piccinno // Journal of Systems and Software. – 2019. – Vol.149. – P.101–137.

³ Coronado E. Visual Programming Environments for End-User Development of Intelligent and Social Robots, a Systematic Review / E. Coronado, F. Mastrogiovanni, B. Indurkha, G. Venture // Journal of Computer Languages. – 2020. – Vol. 58. – 100970.

М.В., Финн В.К., Фоминых И.Б., Хорошевский В.Ф., Хоррокс Я., Шрайбер Г., Штудер Р., Частиков А.П., Черняховская Л.Р. и др. В области автоматизации создания программных систем и их компонентов, разработки трансляторов, а также подходов трансформации моделей и программ можно отметить работы исследователей Агафонова В.Н., Ахо А., Баричели Б.Р., Гасевика Д., Горбунова-Посадова М.М., Гринфилда Дж., Д’Сильва А.Р., Ершова А.П., Клеппе А., Корандо Е., Крету Л.Г., Менса Т., Опарина Г.А., Сабельфельда В.К., Сендала С., Ульмана Дж., Фаулера М., Франкеля Д., Чарнецки К. и др.

Однако, существующие решения в области создания программного обеспечения систем ИИ и БЗ, в том числе, ориентированные на конечных пользователей, обладают рядом недостатков, в частности: акцентом на концептуализацию и формализацию знаний; слабой интеграцией, в том числе «по данным» с другим программным обеспечением, как в контексте использования разработанных ранее концептуальных моделей, так и создания отчуждаемых программных кодов и спецификаций.

Таким образом, актуальна проблема, имеющая важное научно-техническое значение – разработка новой технологии создания программного обеспечения интеллектуальных систем, повышающего эффективность и надежность процессов обработки и передачи данных и знаний в вычислительных машинах, комплексах и компьютерных сетях за счет повторного использования и трансформации концептуальных моделей.

Цель исследования – разработать языки, методы, алгоритмы и программные средства, образующие технологию создания интеллектуальных систем с декларативными базами знаний продукционного и прецедентного типа, особенностью которой является ориентация на конечного пользователя, повторное использование концептуальных моделей и их трансформации.

Задачи исследования:

- Провести анализ языков, инструментальных средств и технологий создания интеллектуальных систем с декларативными базами знаний, в том числе, с использованием модельных трансформаций и ориентированных на конечных пользователей.
- Разработать метод проектирования декларативных баз знаний интеллектуальных систем продукционного и прецедентного типа на основе модельных трансформаций;
- Разработать методы проектирования программ трансформаций концептуальных моделей и программных компонентов-конверторов концептуальных моделей;
- Разработать языки поддержки разработанных методов в части визуального программирования баз знаний и трансформаций концептуальных моделей.
- Разработать алгоритмы и программные средства создания интеллектуальных систем с декларативными базами знаний на основе модельных трансформаций.
- Провести апробацию и оценку эффективности разработки прикладных интеллектуальных систем с декларативными базами знаний на основе модельных трансформаций.

Объектом исследования являются интеллектуальные системы, модели, методы, алгоритмическое и программное обеспечение создания программных средств обработки знаний в вычислительных машинах, комплексах и компьютерных сетях.

Предметом исследования является модели, методы, алгоритмы и программное обеспечение создания интеллектуальных систем с декларативными базами знаний на основе модельных трансформаций.

Методы исследования. В работе использовались методы объектно-ориентированного и визуального программирования, трансформации моделей, построения трансляторов и предметно-ориентированных языков, а также методы и средства искусственного интеллекта и онтологического моделирования.

Научная новизна диссертации заключается в разработке технологии создания интеллектуальных систем с продукционными и прецедентными базами знаний на основе модельных трансформаций, при создании которой получены следующие результаты, выносимые на защиту:

1) Оригинальный метод проектирования декларативных баз знаний интеллектуальных систем, ориентированный на конечных пользователей и расширяющий принципы модельно-ориентированного подхода за счет новых моделей, языков и платформ в контексте инженерии знаний;

2) Методы проектирования программ трансформаций концептуальных моделей и программных компонентов-конверторов концептуальных моделей, отличающиеся от известных использованием оригинального языка описания моделей трансформаций и принципов визуального программирования;

3) Визуальный язык программирования продукционных баз знаний – RVML (Rule Visual Modeling Language), основанный на UML и расширяющий его выразительные способности в контексте моделирования декларативных баз знаний;

4) Оригинальный текстовый декларативный язык описания моделей трансформаций – TMRL (Transformation Model Representation Language), включающий конструкции для описания не только преобразуемых структур и связей между ними, но и механизма взаимодействия с внешними программными компонентами трансформаций;

5) Алгоритмы и архитектура программных средств, обеспечивающих поддержку разработанных языков и методов, объединенных общей идеологией модельных трансформаций и формирующих технологическую платформу создания интеллектуальных систем с декларативными базами знаний продукционного и прецедентного типа.

Соответствие диссертации паспорту научной специальности. Тема и основные результаты диссертации соответствуют следующим областям исследований паспорта специальности 05.13.11 – «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»:

- Модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования (Результаты 1, 2, 5).
- Языки программирования и системы программирования, семантика программ (Результаты 3, 4, 5).

Теоретическая значимость работы заключается в развитии методов и средств создания интеллектуальных систем на основе модельных трансформаций. Результаты диссертационного исследования использованы при выполнении государственных заданий и научных исследований ИДСТУ СО РАН, проектов РФФИ 15-37-20655 (рук.), 15-07-03088 (рук.), 19-07-00927 (рук.), соглашения № 8770 от 04.10.2012 ФЦП «Научные и научно-педагогические кадры инновационной России» (рук.), грантов Президента России по проектам НШ-9508.2006.1, НШ-

1676.2008.1, СП-2012.2012.5 (рук.), работ по гранту Фонда содействия отечественной науке (рук.), проектов РФФИ 18-71-10001 (отв.исп.), 22-21-00099 (рук.), гранта № 075-15-2020-787 Министерства науки и высшего образования РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (проект «Фундаментальные основы, методы и технологии цифрового мониторинга и прогнозирования экологической обстановки Байкальской природной территории»).

Практическая значимость результатов. Предложенные в рамках диссертационной работы модели, методы, алгоритмы и программное обеспечение позволяют снизить трудозатраты и сократить сроки разработки интеллектуальных систем и их баз знаний. Автором получено 14 свидетельств о регистрации программ для электронных вычислительных машин (ЭВМ), основными из которых являются: Personal Knowledge Base Designer (PKBD, рег.№№ 2016617733, 2012614093, 2007613714) – система разработки декларативных баз знаний и интеллектуальных систем и ее веб версия: Web PKBD; Knowledge Base Development System (KBDS, рег.№ 2019661803) – система создания программных компонентов трансформации концептуальных моделей; TreeEditorET/Extended Event Tree Editor (EETE, рег.№ 2012614092) – система визуального проектирования баз знаний на основе деревьев событий.

Практическая значимость результатов подтверждена полученными актами внедрения и справками использования программных систем АО «ИркутскНИИхиммаш», ООО «Смарт Технологии», ООО «ЦентраСиб», ИрННТУ, МГТУ ГА.

Достоверность результатов проведенных исследований подтверждается обоснованным использованием методов модельных трансформаций, публикацией и индексацией полученных результатов в РИНЦ, WoS, Scopus, работоспособностью разработанного программного обеспечения, решением прикладных и тестовых задач.

Апробация результатов диссертации. Основные результаты диссертационного исследования докладывались автором на следующих научных мероприятиях: Национальная конференция по искусственному интеллекту (КИИ, 2008, 2010, 2014, 2016, 2018, 2020, 2021 гг.); Международная конференция «Знания – Онтологии – Теории» (ЗОНТ, 2015, 2017, 2021 гг.); Международная конференция «Системный анализ и информационные технологии» (САИТ, 2009, 2013, 2015, 2019 гг.); Международная научная конференция «Интеллектуальные информационные технологии в технике и на производстве» (ИТИ, Сочи, 2018 и 2021 гг.); Artificial intelligence and digital technologies in technical systems (Волгоград, 2020 и 2021 гг.); 3rd International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE, Иркутск, 2021); Scientific-practical Workshop Information Technologies: Algorithms, Models, Systems (ITAMS, Иркутск, 2018-2021 гг.); Байкальская Всероссийская конференция с международным участием «Информационные и математические технологии в науке и управлении» (ИМТ, Иркутск, 2009, 2011, 2013, 2014, 2016, 2019-2021 гг.); X Международная научно-практическая конференция «Актуальные проблемы и перспективы развития гражданской авиации» (Иркутск, 2021 г.); Международная конференция «Иванниковские чтения» (Нижний Новгород, 2021 г.); 4th Artificial Intelligence and Cloud Computing Conference (AICCC, Киото, Япония, 2021 г.); 24th International Database Engineering & Applications Symposium

(IDEAS, Seoul, Korea, 2020 г.); International Conference «Information and Communication Technologies for Research and Industry» (ICIT, Saratov, 2019-2020 гг.); International Conference on Modelling and Development of Intelligent Systems (MDIS, Sibiu, Romania, 2019-2020 гг.); Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBREIT, Yekaterinburg, 2020 г.); VIIth International Workshop «Critical Infrastructures: Contingency Management, Intelligent, Agent-Based, Cloud Computing and Cyber Security» (IWCI, Байкальск, 2020 г.); VII Всероссийская конференция «Безопасность и мониторинг природных и техногенных систем» (БиМПТС, Кемерово, 2020 г.); 13-я мультikonференция по проблемам управления (МТУиП, Санкт-Петербург, 2020 г.); 1st International Workshop on Advanced Information and Computation Technologies and Systems (Irkutsk, 2020 г.); International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON, Yekaterinburg, 2019 г.); International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO, Opatija, Croatia, 2018-2019 гг.); 3th International Conference «Computational Methods in Systems and Software» (CoMeSySo, 2019 г.); IV Всероссийская Поспеловская конференция с международным участием Гибридные и синергетические интеллектуальные системы (ГИСИС, Светлогорск, 2018 г.); Открытые семантические технологии проектирования интеллектуальных систем (ОСТИС, Минск, Белоруссия, 2016, 2018 гг.); Международная научно-практическая конференция «Программная инженерия: методы и технологии разработки информационно-вычислительных систем» (ПНИВС, Донецк, ДНР, 2016, 2018 гг.); Russian-Pacific Conference on Computer Technology and Applications (RPC, Владивосток, 2017 г.); 12th International Forum on Knowledge Asset Dynamics (IFKAD, St. Petersburg, 2017 г.); XII Международная научно-практическая конференции Объектные системы (Ростов-на-Дону, 2016 г.); 8-й Всероссийская мультikonференция по проблемам управления (Дивноморское, 2015 г.); VI Всероссийская научно-техническая конференция с международным участием «Безопасность критических инфраструктур и территорий» (Абзаково, 2014 г.); Российско-монгольская конференция молодых ученых по математическому моделированию, вычислительно-информационным технологиям и управлению (Ханх, Монголия, 2011, 2013, 2015 гг.); 3 Научная конференция «Автоматизация в промышленности» (Москва, 2009 г.); Международная научно-техническая конференция «Интеллектуальные системы» (AIS) и «Интеллектуальные САПР» (CAD) (Дивноморское, 2008 г.); 16 Международная конференция «Проблемы управления безопасностью сложных систем» (Москва, 2008 г.), а также семинарах ИДСТУ СО РАН.

Личный вклад автора. Все выносимые на защиту научные положения получены соискателем лично. Из совместных исследований в диссертацию включены только те результаты, которые принадлежат непосредственно автору.

Публикации. Результаты диссертационного исследования отражены в 110 научных работах. Основные публикации представлены монографией [1], статьями в российских журналах [2-28], рекомендованных Высшей аттестационной комиссией для опубликования научных результатов диссертации, а также проиндексированных в международных базах цитирования Web of Science и Scopus [29-57], при этом 9 из них [29-37] относятся к Q1 и Q2 по рейтингу SJR. Получено 14 свидетельств о регистрации программ для электронных вычислительных машин (ЭВМ) [58-70].

Структура диссертации. Диссертация состоит из введения, трех разделов и семи глав, заключения, библиографии из 374 наименований, списка принятых сокращений и 9 приложений. Общий объем основного текста работы – 258 страниц, включая 22 таблицы и 118 рисунков.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертационной работы, сформулированы цели и задачи исследования, отражена научная новизна и практическая значимость полученных результатов, приводятся сведения об апробации диссертации, использовании результатов и опубликованных работах, дана общая характеристика работы.

В первом разделе, включающем две главы, приведены результаты аналитического обзора в области технологий, языков и средств создания интеллектуальных систем (ИС) и декларативных баз знаний (БЗ) (**первая глава**), также рассмотрен вопрос использования модельных трансформаций (**вторая глава**) в контексте создания ИС.

Процесс создания ИС и их БЗ остается сложной и трудоемкой задачей, решение которой требуется разработки нового математического и программного обеспечения. Исследования в данной области ведутся с момента появления ИС при этом перспективным является использование методологий (подходов), ориентированных на конечных пользователей (end users). Подобные подходы объединены в направление, известное как End-User Development (EUD), включающее поднаправления: программирование для конечных пользователей (End-User Programming, EUP) и разработка (включая моделирование и проектирование) для конечных пользователей (End-User Software Engineering, EUSE).

Для целей диссертационного исследования перспективным является использование таких принципов EUD подхода, как модельно-ориентированные трансформации, визуальное программирование и предметно/проблемно-ориентированные языки.

Рассмотрение наиболее известных технологий разработки ИС и БЗ (OSTIS, IASPaaS, AT-ТЕХНОЛОГИЯ, САКЕ, МОКА и др.), реализующих отдельные принципы EUD, выявило: акцент на концептуализацию и формализацию знаний, а также крупно-блочное моделирование; наличие ограниченного набора программных средств в открытом доступе готовых к использованию без модификации (настройки) программистами; поддержку ограниченного набора языков программирования баз знаний; слабую возможность интеграции, в том числе «по данным» с другим программным обеспечением и отсутствие возможности создания отчуждаемых программных модулей.

Анализ текстовых и графических языков разработки декларативных БЗ продукционного типа (RuleML, SWRL, R2ML, URML, MML, SCg и др.), которые могут быть использованы конечными пользователями, показал необходимость создания новых средств визуального программирования БЗ, учитывающих особенности продукций и логических отношений и являющихся специализацией общесистемных нотаций, в частности, UML.

В результате анализа программных инструментальных средств создания ИС и БЗ (VisiRule, Visual JESS, Exsys Corvid, Expert System Designer, СИМПР-2015, ПРОБАЗ и др.) было установлено, что только часть из них обеспечивает

использование методов EUD и ориентированы на конечных пользователей, при этом еще меньшее количество систем позволяют создавать отчуждаемые коды и интегрируются с внешними источниками информации, в частности, в форме концептуальных моделей.

Анализ существующих реализаций модельно-управляемого подхода (EMF, MDA, MISC и др.) показал, что наиболее документированной и стандартизированной его разновидностью является MDA (Model-Driven Architecture) – архитектура-управляемая моделью.

Высокие квалификационные требования к пользователю (специалисту) при разработке правил (сценария) трансформаций моделей являются основным недостатком существующих языков трансформаций (QVT, ATL, VIATRA2, Henshin, Epsilon и др.). При этом решения в области применения модельных трансформаций для создания интеллектуальных систем не объединены единой унифицированной технологией разработки и ориентированы на программирующих пользователей.

Проведенный анализ технологий, языков и инструментальных средств позволил сделать вывод об актуальности диссертационного исследования по разработке новой технологии создания ИС с декларативными БЗ на основе модельных трансформаций, включающей следующие основные элементы:

- Оригинальные языки визуального программирования декларативных БЗ и описания трансформаций концептуальных моделей, ориентированные на конечных пользователей;
- Новые методы создания ИС и декларативных БЗ, использующие разработанные языки;
- Комплекс взаимосвязанных программных средств для поддержки создаваемых языков и методов.

Во **втором разделе**, включающем три главы, приведено описание предлагаемых элементов технологии. В **третьей главе** представлены оригинальные языки: язык визуального программирования декларативных баз знаний продукционного типа – Rule Visual Modeling Language (RVML) и язык описания трансформаций концептуальных моделей – Transformation Model Representation Language (TMRL).

RVML является профилем расширения UML, использующим модифицированные базовые элементы «класс» и «ассоциация», позволяет представить логические правила в обобщенном виде, абстрагируясь от особенностей конкретных языков программирования. В то же время он содержит средства для определения приоритетов правил и значений слотов «по умолчанию». RVML имеет расширение, называемое FuzzyRVML, поддерживающее использование лингвистических (нечетких) переменных и коэффициентов уверенности для учета нечеткости и неопределенности в рассуждениях.

RVML включает 8 графических элементов (Рисунки 1 и 2), при этом 5 из них являются основными или базовыми (шаблон факта, узловой элемент правила, факт, условие, связи элементов с указанием действия), 3 используются в расширении FuzzyRVML (лингвистическая (нечеткая) переменная, терм, связь типа «зависимость»). Элементы RVML/FuzzyRVML интегрируются и могут использоваться совместно для описания нечетких продукций.

Одним из назначений RVML является преодоление семантического разрыва между визуальными моделями и программными кодами, в связи с этим обеспечено

однозначное соответствие графических элементов RVML/FuzzyRVML и языковых конструкций языков программирования правил для интеллектуальных систем (Рисунок 3), в частности, CLIPS/FuzzyCLIPS и Drools, а также языка общего назначения РНР. Данное соответствие используется для реализации автоматизированной кодогенерации.



Рисунок 1 – Основные элементы RVML: 1) шаблон факта; 2) узловой элемент правила; 3) факт; 4) условие; 5) связи элементов с указанием действий

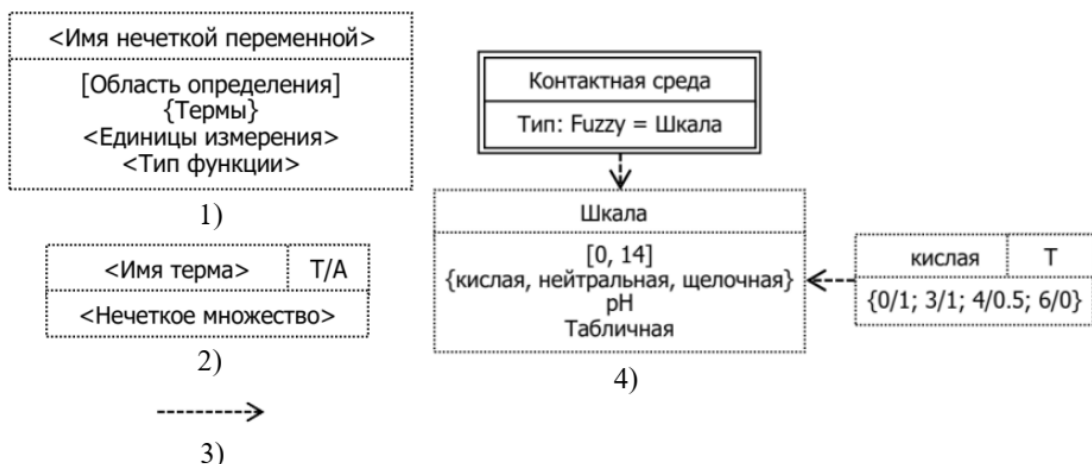


Рисунок 2 – Основные элементы FuzzyRVML: 1) лингвистическая (нечеткая) переменная; 2) терм; 3) связь типа «зависимость»; 4) совместное использование элементов RVML и FuzzyRVML

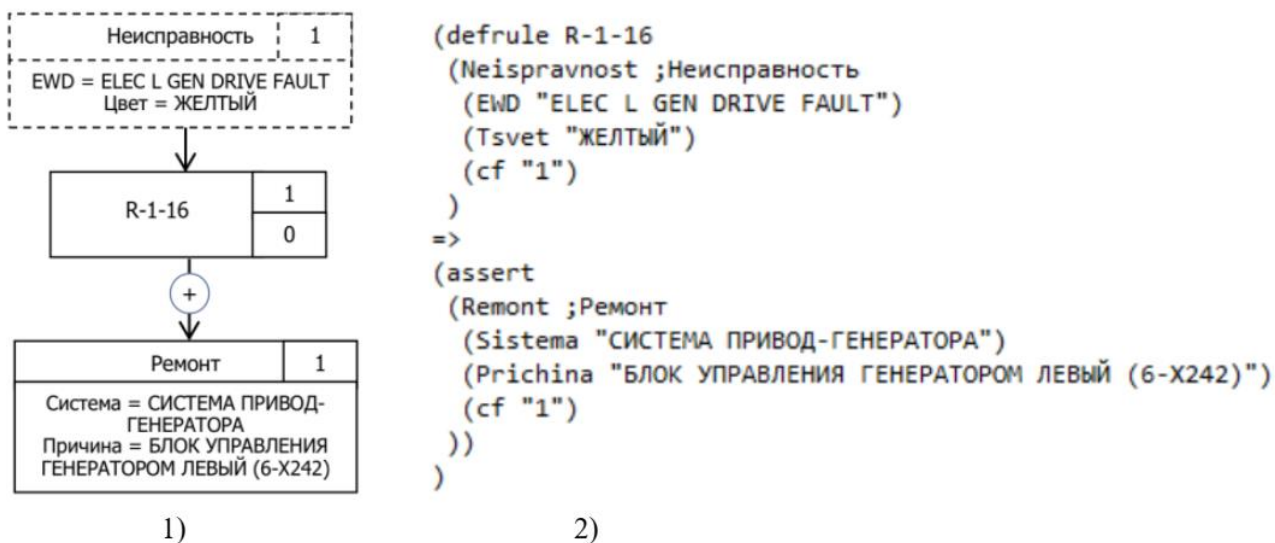


Рисунок 3 – Соответствие конструкций: 1) RVML и 2) CLIPS

По сравнению с аналогами, RVML/FuzzyRVML обладает следующими преимуществами, определяющими его новизну:

- Использует специализированные графические элементы для всех компонентов правил, а не один типовой элемент, отличающийся стереотипом (как в UML).
- Обеспечивает однозначную визуальную индикацию действий, продуцируемых правилами: добавления, удаления, изменения фактов, остановка логического вывода.
- Содержит специализированные элементы для моделирования неполноты и нечеткости: тип данных (Fuzzy); лингвистическая (нечеткая) переменная (FuzzyVar) и набор нечетких термов как возможные значения лингвистической переменной; коэффициент уверенности (Certainty Factor).
- Поддерживает прямое отображение графических элементов в код на CLIPS, FuzzyCLIPS, Drools и др., т.е. может быть использован конечными пользователями в процессе визуального программирования БЗ.

Для описания трансформаций концептуальных моделей разработан текстовый декларативный язык – Transformation Model Representation Language (TMRL). Грамматика TMRL принадлежит к классу контекстно-свободных грамматик (КС-грамматик – LL(1)). Программы на TMRL удовлетворяют требованиям полноты, понятности и точности⁴, т.е. объекты модели хорошо формализованы, содержат необходимую информацию, а спецификации понятны (читабельны) и компактны.

TMRL включает 15 служебных лексем (элементов), при этом: 7 лексем используются для описания исходной и целевой метамодели; 7 – описывают трансформации; 1 – обеспечивает вызов (взаимодействие) внешних программных компонентов трансформации. Формальное описание TMRL (РБНФ):

Модель трансформации на TMRL = "**Transformation Model**" , Заголовок, "{" , Модель трансформации , "}" , Оператор вызова.

Модель трансформации = Исходная метамодель , Целевая метамодель , Оператор трансформации.

Исходная метамодель = "**Source Meta-Model**" , Заголовок, "{" , Тело метамодели, "}" .

Целевая метамодель = "**Target Meta-Model**" , Заголовок, "{" , Тело метамодели, "}" .

Оператор трансформации = "**Transformation**" , Заголовок оператора трансформации , "{" , Тело трансформации, "}" .

Заголовок = Буква , { Буква | Цифра | Символ } .

Буква = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z" | "А" | "а" | "Б" | "б" | "В" | "в" | "Г" | "г" | "Д" | "д" | "Е" | "е" | "Ё" | "ё" | "Ж" | "ж" | "З" | "з" | "И" | "и" | "Й" | "й" | "К" | "к" | "Л" | "л" | "М" | "м" | "Н" | "н" | "О" | "о" | "П" | "п" | "Р" | "р" | "С" | "с" | "Т" | "т" | "У" | "у" | "Ф" | "ф" | "Х" | "х" | "Ц" | "ц" | "Ч" | "ч" | "Ш" | "ш" | "Щ" | "щ" | "Ъ" | "ъ" | "Ы" | "ы" | "Ь" | "ь" | "Э" | "э" | "Ю" | "ю" | "Я" | "я" .

Цифра = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" .

Символ = "-" | "." .

Тело метамодели = Элементы , Связи .

Элементы = "**Elements**" , "[" , { Элемент } , "]" .

⁴ Агафонов В.Н. Требования и спецификации в разработке программ / В.Н. Агафонов. – М.: Мир, 1984. – 344 с.

Элемент = Наименование , [Атрибуты] , ",".
 Атрибуты = **"attributes"** , "(" , { Атрибут } , ")".
 Атрибут = Наименование , ",".
 Наименование = Буква , { Буква | Цифра | Символ }.
 Связи = **"Relationships"** , "[" , { Связь } , "]".
 Связь = Ассоциация | Связь по идентификатору.
 Ассоциация = Левый элемент ассоциации , **"is associated with"** , Правый элемент ассоциации , ",".
 Левый элемент ассоциации = Наименование.
 Правый элемент ассоциации = Наименование.
 Связь по идентификатору = Левый элемент связи по идентификатору , **"is"** , Правый элемент связи по идентификатору , ",".
 Левый элемент связи по идентификатору = Наименование элемента , "(" , Наименование атрибута , ")".
 Правый элемент связи по идентификатору = Наименование элемента , "(" , Наименование атрибута , ")".
 Наименование элемента = Наименование.
 Наименование атрибута = Наименование.
 Заголовок оператора трансформации = Название исходной метамодели , **"to"** , Название целевой метамодели.
 Название исходной метамодели = Заголовок.
 Название целевой метамодели = Заголовок.
 Тело трансформации = { Правило }.
 Правило = Заголовок правила , "[" , Тело правила , "]".
 Заголовок правила = **"Rule"** , Исходные элементы , **"to"** , Целевые элементы , **"priority"** , Цифра.
 Исходные элементы = Единичный исходный элемент | Множество исходных элементов.
 Единичный исходный элемент = Наименование.
 Множество исходных элементов = "(" , Исходный элемент , { Дополнительный исходный элемент } , ")".
 Дополнительный исходный элемент = "," , Исходный элемент.
 Исходный элемент = Наименование.
 Целевые элементы = Единичный целевой элемент | Множество целевых элементов.
 Единичный целевой элемент = Наименование.
 Множество целевых элементов = "(" , Целевой элемент , { Дополнительный целевой элемент } , ")".
 Дополнительный целевой элемент = "," , Целевой элемент.
 Целевой элемент = Наименование.
 Тело правила = { Выражение определения } , { Условное выражение }.
 Выражение определения = Левый элемент выражения , **"is"** , Правый элемент выражения.
 Левый элемент выражения = Целевой элемент , "(" , Атрибут целевого элемента , ")".
 Атрибут целевого элемента = Наименование.
 Правый элемент выражения = Исходный элемент , "(" , Атрибут исходного элемента , ")" , [Дополнительный правый элемент выражения] .
 Атрибут исходного элемента = Наименование.
 Дополнительный правый элемент выражения = { **"or"** Исходный элемент , "(" , Атрибут исходного элемента , ")" } .
 Условное выражение = Левый элемент выражения , **"is"** , Значение выражения , [Условие] , [Альтернативное значение выражения] .
 Условие = "[" , Описание условия , "]"
 Альтернативное значение выражения = **"or"** , Значение выражения , "[" , Условие , "]" .
 Описание условия = **"if"** , "(" , Правый элемент выражения , **"is"** , Значение выражения ")" , [Дополнительное условие] .
 Дополнительное условие = **"and"** , "(" , Правый элемент выражения , **"is"** , Значение выражения ")" .
 Значение выражения = "\"" , { Буква | Цифра | Символы } , "\"".

Символы = "@" | "#" | "\$" | "%" | "^" | "&" | "*" | "/" | "+" | "-" | "." | "?" | "!" | "," | ";" | ":" | "/" | "[" | "]" | "{" | "}" | "=" | "_" | " " | "\".

Оператор вызова = "Call" , Название программного компонента трансформации , ", " , Путь к концептуальной модели , [", " , Путь сохранения базы знаний] .

Название программного компонента трансформации = { Буква | Цифра | Символы } .

Путь к концептуальной модели = { Буква | Цифра | Символы } .

Путь сохранения базы знаний = { Буква | Цифра | Символы } .

Нечеткое множество = Вид описания [Вид функции принадлежности (Параметры функции принадлежности)] .

Вид описания = Табличное | Аналитическое

Вид функции принадлежности = Треугольная | Трапециевидная | S-образная сплайн-функция | Z-образная сплайн-функция | Линейная S-образная функция | Кинейная Z-образная функция | П-образная.

Программа на TMRL включает четыре основных раздела (блока):

- 1) Описание структуры исходной модели (основные понятия и отношения): «Source Meta-Model»;
- 2) Описание структуры целевой метамодели (основные понятия и отношения): «Target Meta-Model»;
- 3) Описание правил трансформации: «Transformation»;
- 4) Вызов внешних модулей: «Call».

Далее приведен пример фрагмента программы на TMRL, описывающей трансформацию UML диаграмм классов в онтологии:

```
Source Meta-Model UML-diagram-class {
  Elements [
    Model,
    Class attributes (xmi.id, name),
    ...
  ]
  Relationships [
    Model is associated with Namespace.ownedElement,
    Namespace.ownedElement is associated with Class,
    DataType(xmi.id) is Attribute(type),
    ...
  ]
}
Target Meta-Model Ontology {
  Elements [
    ExtendedOntology attributes (id, name),
    Class attributes (id, name),
    ...
  ]
  Relationships [
    Ontology is associated with Class,
    ...
  ]
}
Transformation UML-diagram-class to Ontology {
  Rule (Class, ModelElement.name) to Class priority 2 [
    Class(name) is Class(name) or ModelElement.name
    Class(id) is Class(xmi.id)
  ]
  ...
}
```

Основные отличия и особенности TMRL от других языков трансформации моделей общего назначения (универсальных):

- Самодостаточность (полнота) – TMRL не является расширением других языков и не требует применения других языков, например, OCL для создания программ;
- Возможность взаимодействия с другими TMRL программами.

В главе четыре представлены оригинальные модели, метод и программные средства для создания программного обеспечения систем ИИ с декларативными БЗ на основе следующих EUD подходов: визуального программирования и модифицированного модельно-ориентированного подхода (MDA/MDE) от OMG.

Согласно MDA/MDE⁵, разрабатываемое программное обеспечение, включая описание основных понятий, взаимосвязей и методов их обработки, представляется в виде набора информационных, в частном случае, концептуальных моделей, определяющих его состав, структуру и поведение. При этом процесс разработки программного обеспечения представляет собой последовательный переход от моделей с большей абстракцией, к моделям с меньшей абстракцией и программным кодам и спецификациям БЗ и ИС.

Приведем формальное описание MDA/MDD⁶ в следующем виде:

$$MDA/MDD = \left\langle \begin{array}{l} MOF, UML, CIM, PIM, PSM, PDM, \\ F_{CIM-to-PIM}, F_{PIM-to-PSM}, F_{PSM-to-CODE} \end{array} \right\rangle,$$

где *MOF* (*Meta Object Facility*) – абстрактный язык для описания моделей (язык описания мета-моделей); *UML* (*Unified Modelling Language*) – унифицированный язык моделирования, как основной язык для описания всех моделей; *CIM* (*Computation Independent Model*) – вычислительно-независимая модель – модель, скрывающая любые детали реализации, описывает только требования к системе и ее окружению; *PIM* (*Platform Independent Model*) – платформу-независимая модель – модель, скрывающая детали реализации системы, зависимые от платформы, и содержащая элементы, не изменяющиеся при взаимодействии системы с любой платформой; *PSM* (*Platform Specific Model*) – платформу-зависимая модель – модель системы с учетом деталей реализации и процессов, зависимых от конкретной платформы; *PDM* (*Platform Description Model*) – модель платформы – набор технических характеристик и описаний технологий и интерфейсов, составляющих программную (технологическую) платформу; $F_{CIM-to-PIM}: CIM \rightarrow PIM$, $F_{PIM-to-PSM}: PIM \rightarrow PSM$, $F_{PSM-to-CODE}: PSM \rightarrow CODE$ – правила (операторы) преобразования моделей.

Тогда предлагаемый метод в контексте решения задачи разработки БЗ и ИС/ЭС на основе модельных трансформаций будет иметь следующий вид:

$$MDA/MDD^{ES} = \left\langle \begin{array}{l} MOF, L^{ES}, CIM^{ES}, PIM^{ES}, PSM^{ES}, PDM^{ES}, \\ F_{CIM-to-PIM}^{ES}, F_{PIM-to-PSM}^{ES}, F_{PSM-to-CODE}^{ES} \end{array} \right\rangle,$$

где L^{ES} – набор языков и формализмов, используемых для моделирования, в рамках решаемой задачи:

$$L^{ES} = \{UML, CM, DT, ET, CT, RVML\}$$

⁵ 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.

⁶ Cretu L.G. Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice. / L.G. Cretu, D. Florin. Apple Academic Press. – 2014.

где *UML* – унифицированный язык моделирования (стандартное средство MDA/MD подхода); *CM* – формализм концепт-карт; *DT* – формализм представления таблиц решений; *ET* – формализм представления деревьев событий; *CT* – формализм представления канонических таблиц (специализированной унифицированной формы представления электронных таблиц); *RVML* – Rule Visual Modeling Language, как специализация *UML* для моделирования логических правил;

CIM^{ES} – вычислительно-независимая модель в контексте решаемой задачи, представляемая посредством L^{ES} ;

PIM^{ES} – платформу-независимая модель в контексте решаемой задачи, описывает архитектурные элементы ИС и структуру БЗ в нотации *RVML*;

PSM^{ES} – платформу-зависимая модель в контексте решаемой задачи, учитывает особенности целевых платформ, строится с использованием *RVML*;

PDM^{ES} – набор моделей описания платформ, в рамках решаемой задачи:

$$PDM^{ES} = \{CLIPS, DROOLS, PHP, PKBD\}$$

где *CLIPS* и *DROOLS* – специализированные языки программирования ИС и БЗ; *PHP* – язык программирования общего назначения; *PKBD* (Personal Knowledge Base Designer) – инструментальное средство диссертанта, поддерживающее предлагаемый метод.

$F_{CIM \rightarrow PIM}^{ES}$, $F_{PIM \rightarrow PSM}^{ES}$, $F_{PSM \rightarrow CODE}^{ES}$ – правила трансформаций реализованные либо императивно, с использованием языка разработки *PKBD*, либо декларативно, с использованием *TMRL*.

С точки зрения архитектуры (схемы) метамоделирования⁵ предлагаемый метод может быть представлен следующим образом (Рисунок 4), где серым цветом отмечены блоки, предлагаемые диссертантом и расширяющие базовую схему.

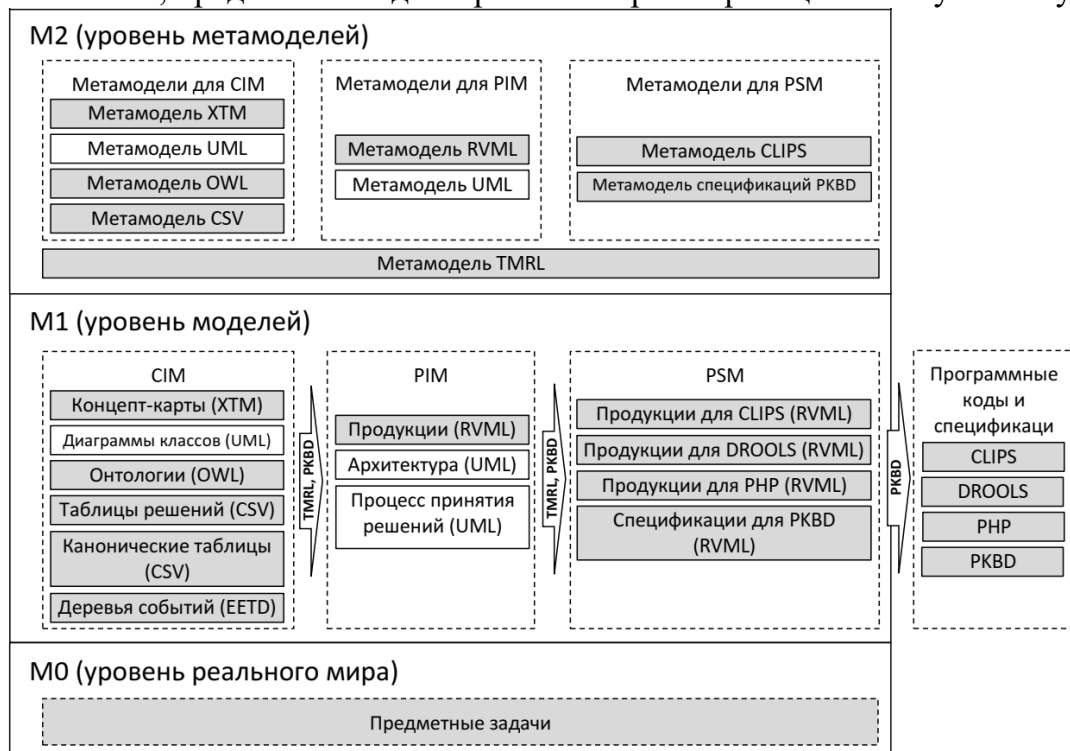


Рисунок 4 – Фрагмент архитектуры (схемы) метамоделирования, описывающей трансформации modelей в рамках предлагаемого метода на различных уровнях абстракции

Процесс разработки БЗ при помощи предлагаемого метода представлен следующей последовательностью этапов (Рисунок 5), обеспечивающих создание и трансформацию концептуальных моделей:

1) Построение модели предметной области. Выходная информация этого этапа: модель предметной области и концептуальная модель ИС/ЭС. Полученные модели рассматриваются как вычислительно-независимые и могут быть представлены в виде OWL-онтологии, UML-моделей (в частности, в виде UML-диаграмм классов) или концепт-карт. Способы формирования этих моделей в рамках метода не конкретизируются. Соответствующие понятия могут быть выявлены как при извлечении знаний из экспертов, так и при анализе существующих онтологий и моделей данных.

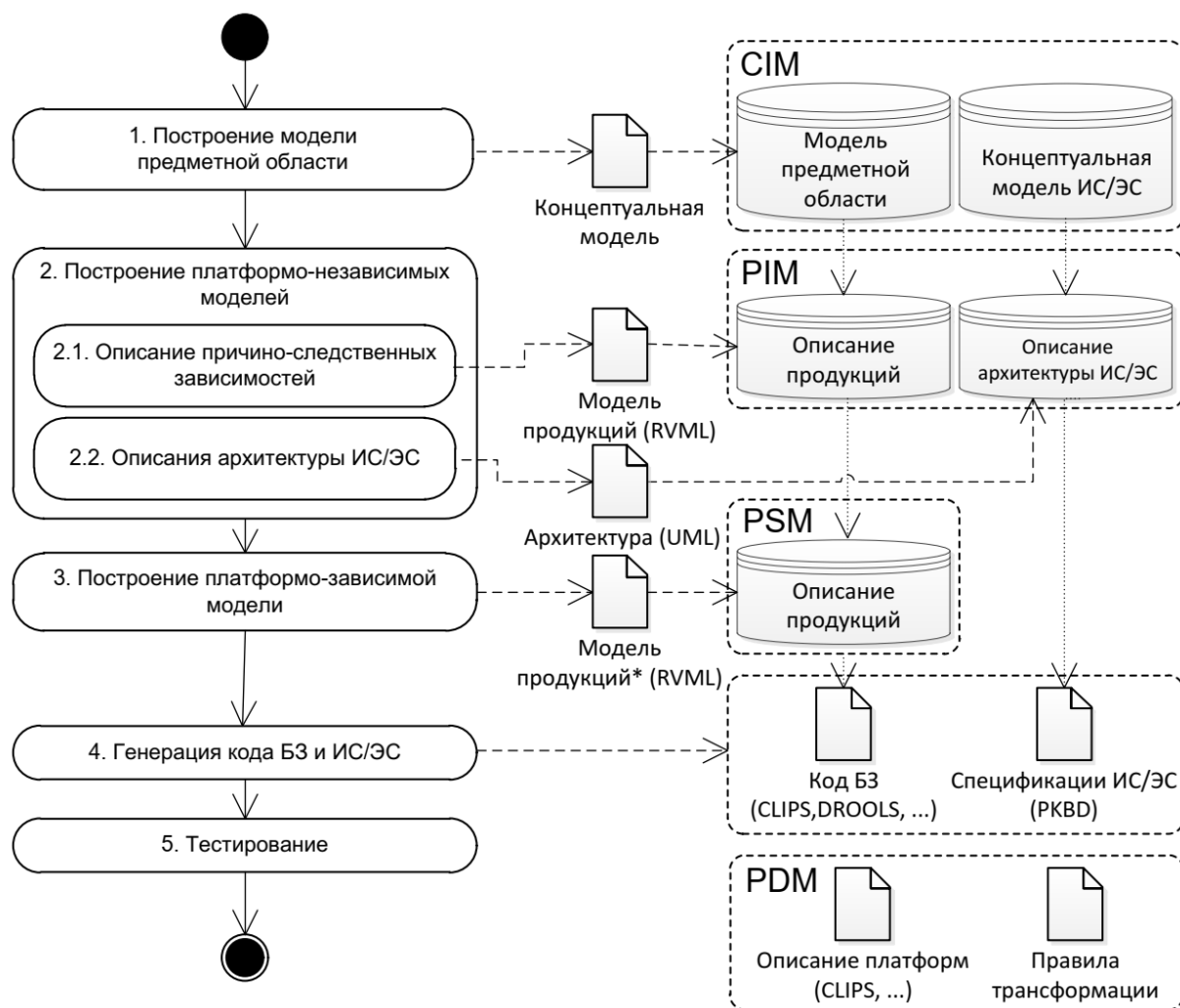


Рисунок 5 – Основные этапы и модели предлагаемого метода проектирования БЗ на примере производственной ИС

Эффективность данного этапа может быть повышена путем повторного использования существующих концептуальных моделей, созданных с использованием различных онтологических и концептуальных (когнитивных) редакторов или CASE-средств (Protégé, SmartTools, FreeMind, Xmind, IBM Rational Rose Enterprise, StarUML, и т.д.). В случае создания производственных БЗ и ИС важно, чтобы полученная на данном этапе предметная модель была семантически значимой, то есть описывала какие-либо причинно-следственные связи в конкретной предметной области, чтобы на ее основе можно было создать БЗ.

Таким образом при использовании онтологий в качестве средства формализации вычислительно-независимых моделей (их унифицированного представления), в дополнение к отношениям «является частью» и «является», вводится отношения «зависит от» – связь обеспечивает описание причинно-следственных связей. В случае диаграмм классов UML типы отношений определяются с помощью механизма стереотипов.

Концептуальная модель ИС формируется на уровне требований к составу ее основных модулей. Поскольку тип создаваемых систем определен заранее – это продукционные или прецедентные ИС, то для формирования этой модели используется соответствующий шаблон, который определяет основные архитектурные элементы и включает соответствующие понятия).

2) Построение платформо-независимых моделей. Выходная информация этого этапа: описание логических правил; подробное описание архитектуры ИС. Модели данного этапа строятся путем преобразования вычислительно-независимых моделей в инструментальном программном обеспечении.

3) Построение платформо-зависимых моделей. Количество этих моделей определяется количеством платформ, для которых создается ИС. Платформо-зависимые является результатом автоматических преобразований платформо-независимой модели с помощью специальных инструментов с последующей модификацией конечным пользователем, который должен уточнить RVML модели правил с учетом особенностей определенного языка представления/программирования знаний (например, CLIPS), такие как: приоритеты правил, значения слотов «по умолчанию» и коэффициент уверенности (определенности).

4) Генерация программных кодов и спецификаций БЗ и ИС. На этом этапе выполняется интерпретация диаграмм, описывающих архитектуру программного обеспечения и БЗ.

5) Тестирование, связанное, в том числе с проверкой корректности полученных структур знаний. Поскольку генерация кода полностью автоматизирована и синтаксически корректна, конечный пользователь может проверить только семантическую корректность разработанных моделей, «выполнив» их для разных значений исходных фактов. Критерием тестирования является правильность логического вывода и корректность его результатов.

Следует отметить, что конечные пользователи (эксперты предметной области или аналитики) активно участвуют в разработке вычислительно-независимой и платформо-независимой моделей и лишь частично платформо-зависимой. Все преобразования моделей и генерация программных кодов осуществляются с помощью специализированного программного обеспечения, включающего модель платформы.

Правила преобразования моделей при переходе между этапами могут быть реализованы в виде спецификаций на императивном языке программирования, либо в декларативной форме с использованием TMRL. В упрощенном виде трансформации представлены в форме таблицы соответствий (Таблица 1).

Особенностью метода является использование EUD подходов, в частности: визуального программирования и модельно-ориентированной разработки в контексте создания ИС и БЗ, оригинальных языков RVML и TMRL для описания элементов баз знаний и моделей трансформаций, а также концепт-карт, онтологий,

деревьев событий, канонических таблиц и таблиц решений при построении вычислительно-независимых моделей.

Таблица 1 – Фрагмент таблицы отображения элементов моделей (вычислительно-независимой модели в платформу-независимую и платформу-независимой в CLIPS)

Элементы онтологии (вычислительно- независимая модель)	Элементы БЗ (платформу-независимая модель)	Элементы CLIPS кода
Project (name, description)	Knowledge base (name, description)	-
Class (name, description)	Template (name, description)	deftemplate
Object (name, description)	Fact (name, description)	deffacts
Method	-	-
Property	Slot (description, value)	slot
Property value	Slot value	default
Property type	Slot type	type
Relationship	Rule (nodal element)	defrule

В пятой главе рассмотрены оригинальные модели, методы и система программирования трансформаций концептуальных моделей и создания программных компонентов трансформации концептуальных моделей, основанные на применении языка TMRL. В контексте данной работы, трансформация – генерация целевой модели из исходной в соответствии с некоторым набором правил преобразования. Трансформация может применяться к нескольким исходным и/или целевым моделям, например, для объединения (слияния) нескольких исходных моделей, либо для преобразования одной общей платформу-независимой модели в несколько платформу-зависимых моделей по количеству целевых платформ. Основными целевыми платформами в рамках диссертационного исследования являются OWL онтологии и продукции на CLIPS.

Метод предполагает формальное представление моделей трансформаций согласно⁷ в следующем виде: $M_T = \langle MM_{IN}, MM_{OUT}, T \rangle$, где

MM_{IN} – метамодель входной концептуальной модели; MM_{OUT} – метамодель целевой (выходной) модели представления знаний (модели БЗ); T – оператор преобразования, при этом:

$$MM_{IN} \in \{MM_{CM}, MM_{PR}, MM_{ONT}\},$$

$$MM_{OUT} \in \{MM_{PR}MM_{ONT}, MM_{CLIPS}, MM_{OWL}\},$$

где MM_{CM} – метамодель концептуальной модели CM_{XML} ; MM_{PR} – метамодель обобщенной модели продукций M_{PR} ; MM_{ONT} – метамодель обобщенной модели онтологии M_{ONT} ; MM_{CLIPS} – метамодель языка CLIPS; MM_{OWL} – метамодель языка OWL.

Процесс описания модели трансформации представлен в виде следующей последовательности этапов:

- 1) Анализ структуры исходной и целевой метамодели, при этом

⁷ Sendall S. Model Transformation: The Heart and Soul of Model-Driven Software Development / S. Sendall, W. Kozaczynski // IEEE Software. – 2003. – Vol. 20, № 5. – P. 42–45.

формируются списки основных элементов моделей.

2) Формирование частичной модели трансформации, содержащей только выделенные наборы элементов исходной и целевой метамодели (MM_{CM} и MM_{KB}).

3) Формирование полной модели трансформации, включая правила трансформации T путем установления соответствий между исходными элементами метамодели MM_{CM} и целевыми элементами метамодели MM_{KB} .

4) Генерация кода модели трансформации на TMRL.

5) Уточнение кода модели трансформации на TMRL.

Особенностями метода проектирования (программирования) трансформаций концептуальных моделей, определяющими его новизну, является: использование оригинального языка описания моделей трансформации TMRL, обладающего простым человекочитаемым синтаксисом, обеспечивающего взаимодействие с разработанными ранее программами и ориентированного на описание трансформаций концептуальных моделей, сериализованных в XML-подобных форматах; использование онтологической модели и обобщенной модели продукций для внутреннего унифицированного представления преобразуемых и целевых форматов; использование оригинальных метамodelей, описывающих ЯПЗ OWL и CLIPS.

Для реализации возможности трансформации различных концептуальных моделей с использованием TMRL разработан метод создания специализированных программных компонентов, под которыми в общем случае понимаются⁸ элементы структуры программной системы, определенным образом выделенные среди окружения, решающие некоторые подзадачи в рамках общих задач системы и взаимодействующие с окружением через определенный интерфейс.

В данной работе программный компонент – это прикладная программа, обеспечивающая автоматизированное создание БЗ на целевом языке путем трансформации исходных концептуальных моделей, построенных при помощи различных систем концептуального, когнитивного, онтологического моделирования или CASE-средств и представленных в виде XML-документов.

Основными элементами метода создания программных компонентов-конверторов являются:

1) Модель типового программного компонента трансформации, представляющего собой каркас (шаблон) со слотами⁹, который может быть настроен на определенные форматы концептуальной модели и БЗ путем интерпретации блока модели трансформации в форме декларативной программы.

2) Принцип создания программных компонентов путем копирования типового программного компонента и его настройки на основе модели трансформации в форме декларативной программы на TMRL, содержащей описание трансформаций и вызываемых блоков модулей анализа и генерации.

Используя определение программного компонента⁸, модель типового компонента-конвертора описывается следующим образом:

$$M_{TPC} = \langle M_T, A_{IN}, CG_{OUT}, I \rangle,$$

⁸ Добрынин В.Ю. Технологии компонентного программирования / В.Ю. Добрынин. – СПб.: СПбГУ, 2003. – 121 с.

⁹ Горбунов-Посадов М.М. Облик многократно используемого компонента / М.М. Горбунов-Посадов // Открытые системы. – 1998. – № 3. – С. 45–49.

где M_T – модель трансформации, используемая для специализации программного компонента (слот сценария преобразования); A_{IN} – анализатор (парсер) входных моделей (слот для модуля анализатора); CG_{OUT} – генератор выходных моделей (слот для модуля генератора); I – интерфейс взаимодействия с внешними системами, обеспечивающий доступ к анализатору и генератору.

Для реализации предлагаемых методов создано оригинальное программное обеспечение: Personal Knowledge Base Designer (PKBD, рег.№№ 2016617733, 2012614093, 2007613714) – система разработки декларативных баз знаний и интеллектуальных систем и ее веб версия: Web PKBD; TreeEditorET/Extended Event Tree Editor (EETE, рег.№ 2012614092) – система визуального проектирования баз знаний на основе деревьев событий, а также система поддержки модельных трансформаций на TMRL - Knowledge Base Development System (KBDS) (рег.№ 2019661803). Разработанные программные средства объединены общей идеологией и формируют технологическую платформу (Рисунок 6) создания ИС.

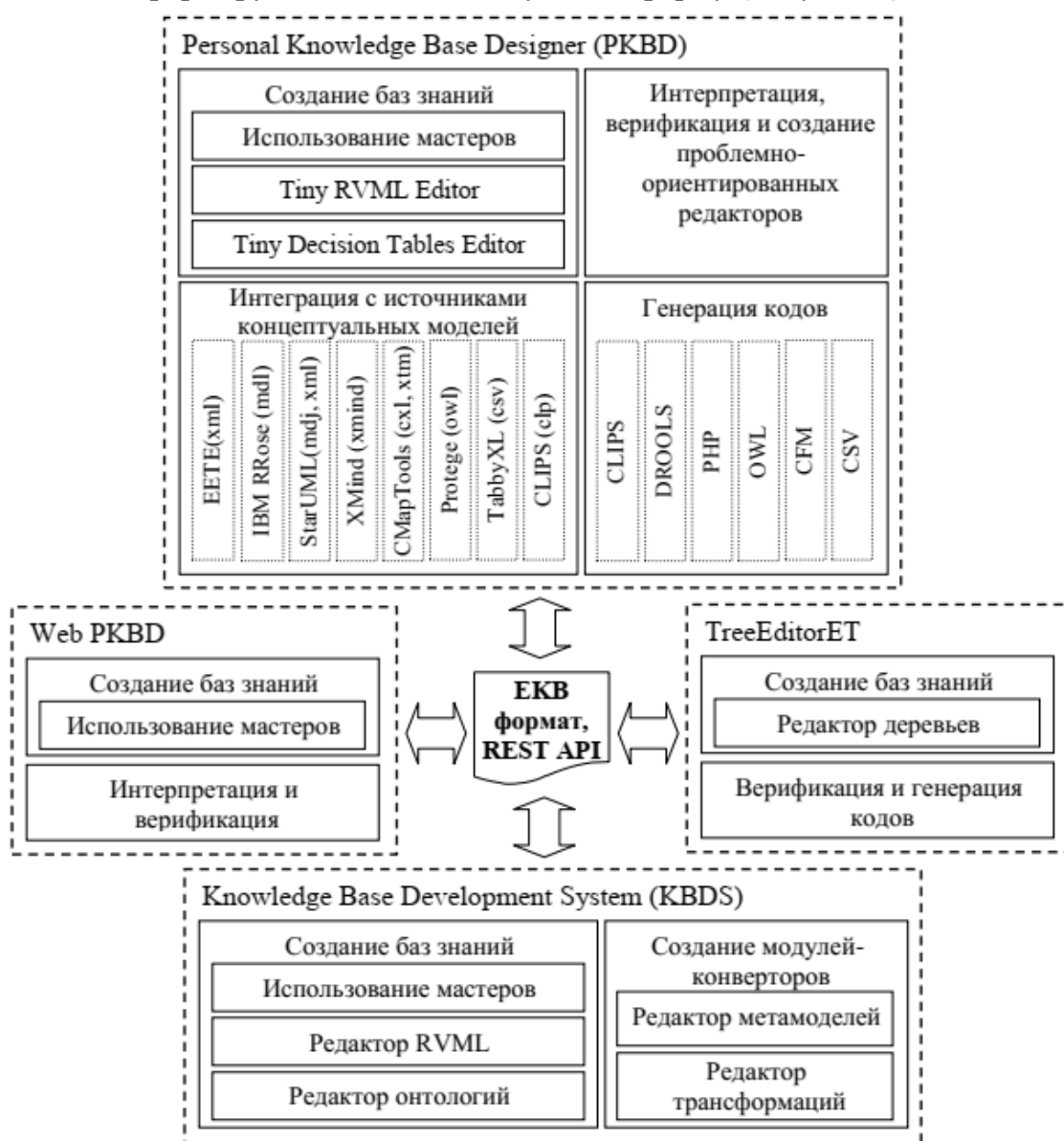


Рисунок 6 – Структура комплекса программных систем с указанием основных функций и подсистем

Интеграция программных средств реализуется путем использования единого формата межпрограммного обмена данными и знаниями, основанного на XML – ЕКВ (External Knowledge Base), а также взаимодействия через REST API.

Методика создания ИС и БЗ на основе модельных трансформаций конкретизирует, описанный ранее метод проектирования БЗ в части применения определенного инструментария и схематически представлена на Рисунке 7. При этом содержание этапов может качественно переопределяться в зависимости от особенностей решаемой задачи (предметной области), однако их назначение в части получаемого результата принципиально не изменяется:

Этап 1. Анализ предметной области и выделение основных сущностей и отношений выполняется с использованием комплекса программных средств PKBD, EETE или KBDS. В зависимости от формы представления исходных данных: «0. Информация предметной области» могут быть использованы те или иные модули трансформации: диаграмм классов UML, концепт-карт, канонических таблиц TabbyXL, деревьев событий или таблиц решений при этом результат: «1. Модель предметной области» представляется в форме набора основных понятий и отношений, формирующие вычислительно-независимую модель.

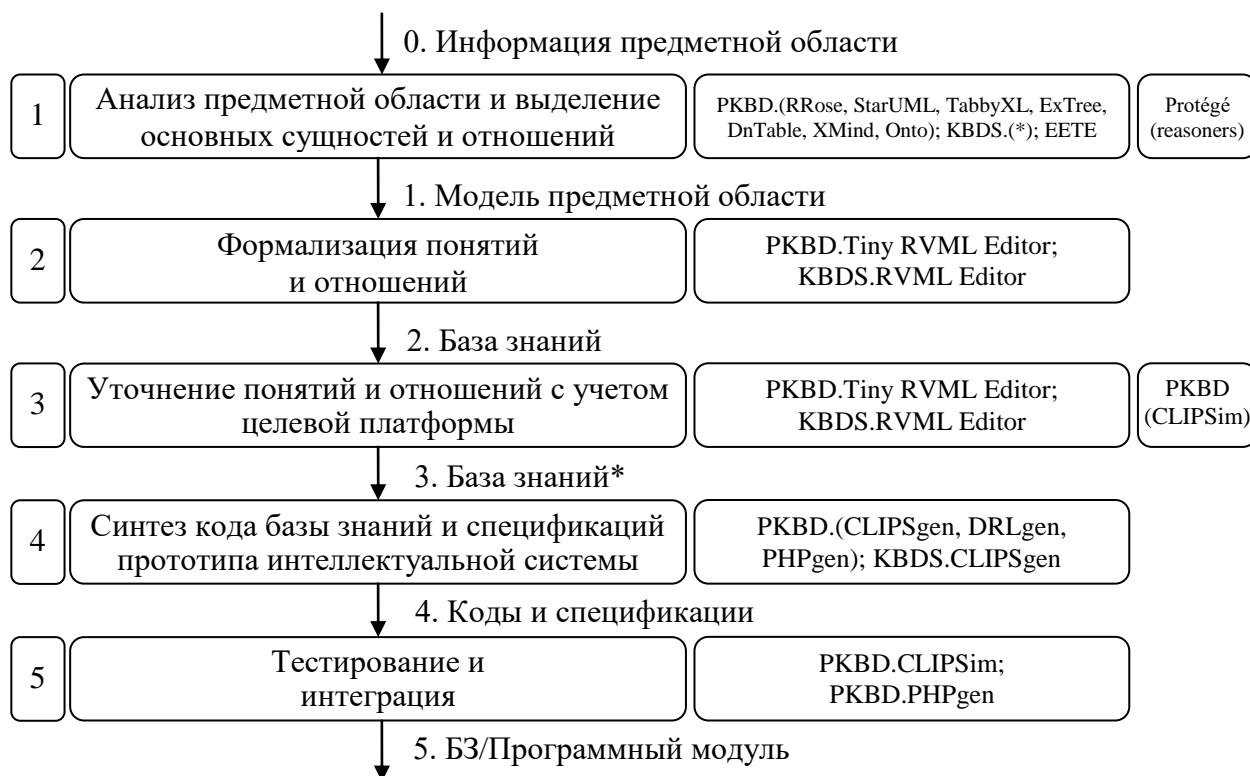


Рисунок 7 – Схема методики создания ИС и БЗ на основе модельных трансформаций с указанием основных этапов и программных средств

Этап 2. Формализация понятий и отношений выполняется с помощью реализованных в PKBD и KBDS подходов EUD: визуального программирования с поддержкой RVML и программ-мастеров. Основной результат данного уточнения – это «2. База знаний», представляющая собой платформу-независимую модель, обеспечивающую формализацию модели предметной области (вычислительно-независимой модели), в частном случае, в форме диаграмм RVML или таблиц решений.

Этап 3. Уточнение понятий и отношений с учетом целевой платформы включает, в том числе, указание типов данных, значений «по умолчанию», приоритетов и пр. Основным результатом этапа «3. База знаний*» представляется в форме уточненных диаграмм RVML, представляющих собой платформо-зависимые модели.

Этап 4. Синтез кода базы знаний и спецификаций прототипа интеллектуальной системы выполняется с помощью реализованных в РКВД и KBDS генераторов программных кодов под CLIPS, DROOLS, PHP. Основным результатом этапа «4. Коды и спецификации».

Этап 5. Тестирование и интеграция осуществляется средствами РКВД за счет использования встроенной машины вывода, обеспечивающей «прогонку» полученных БЗ на предмет выявления логических ошибок. Интеграция осуществляется вне средств разработки и в некоторых случаях не требуется (если исполнение ЭС осуществляется в интерпретаторе РКВД), поэтому данный этап может быть пропущен, а тестирование совмещено с предыдущим этапом 4.

На рисунке 8 представлен пример создания БЗ согласно описанной методике.

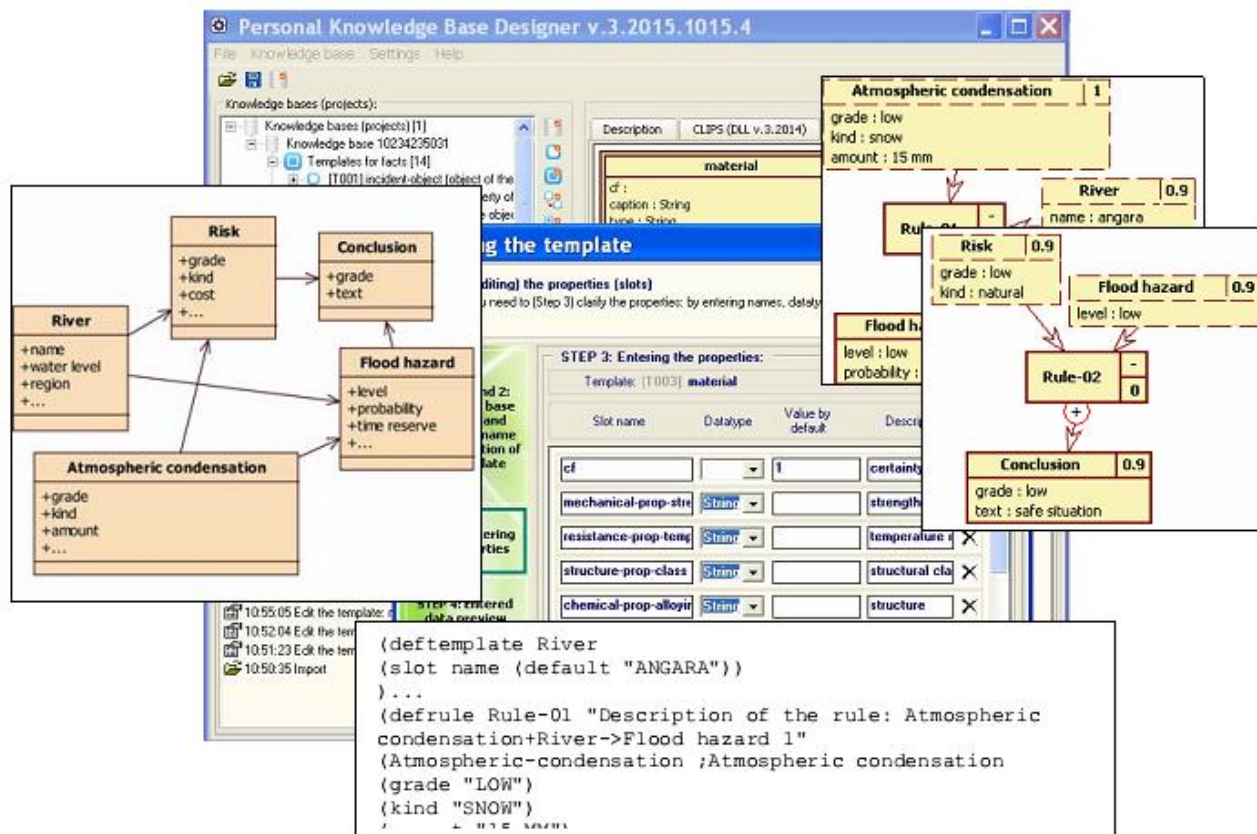


Рисунок 8 – Иллюстрация этапов создания БЗ на основе модельных трансформаций

На рисунке 9 приведены примеры процесса описания трансформаций в KBDS.

В **третьем разделе**, включающем две главы, представлены применения и оценки эффективности разработанных методов и средств при решении ряда практических и учебных задач. В том числе, в **шестой главе** рассмотрены задачи обеспечения надежности и безопасности оборудования в нефтехимии (АО «ИркутскНИИХимМаш»), в том числе:

1) Разработка БЗ для прогнозирования развития деградиционных процессов в нефтехимии для ИАС «Экспертиза ПБ» (рег.№ 2016610757).

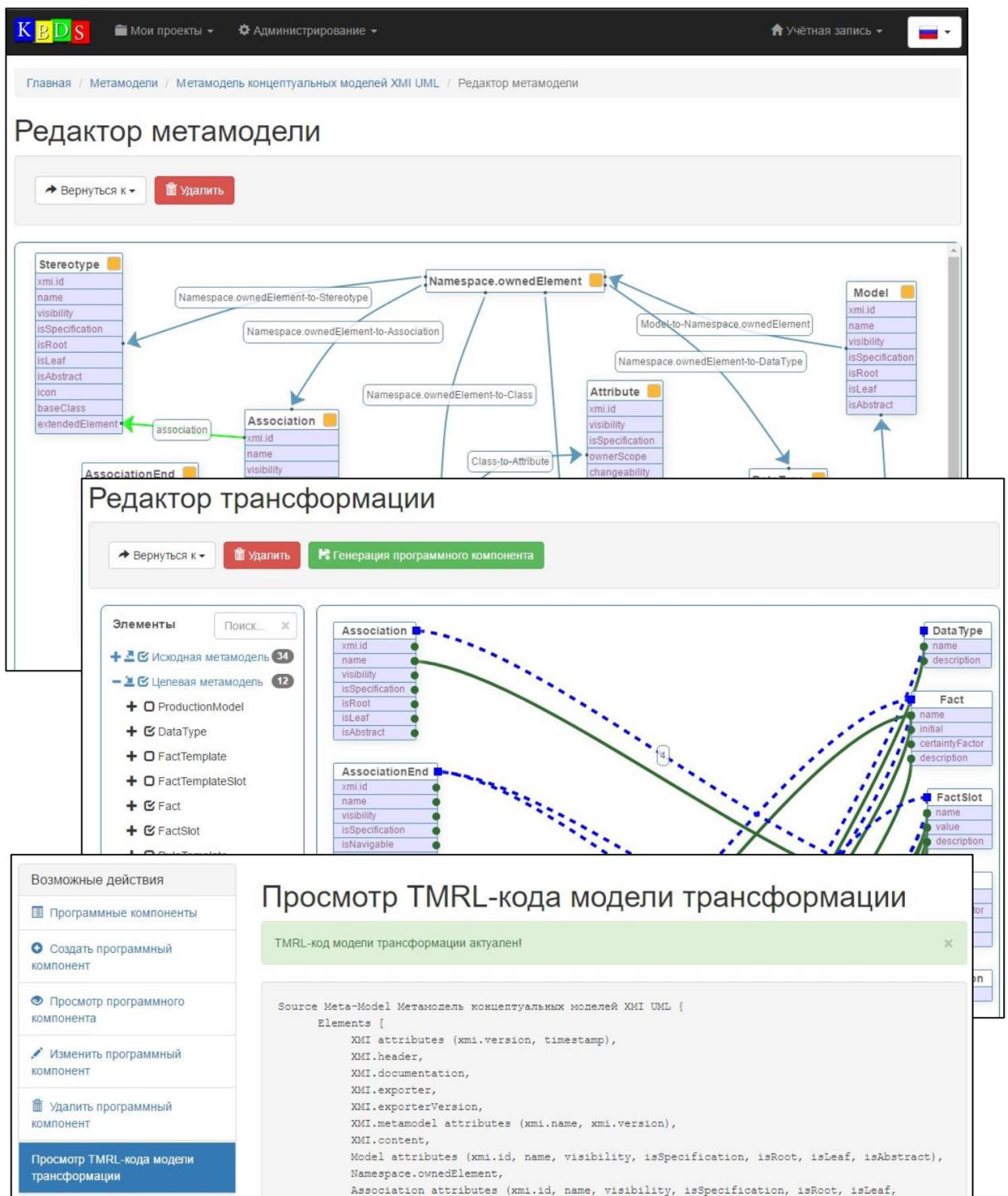


Рисунок 9 – Экранные формы KBDS при описании трансформаций

Повышение безопасности промышленных объектов с каждым годом становится все более актуальной проблемой, так как старение оборудования во многих отраслях промышленности опережает темпы его модернизации и замены, как по объективным, так и субъективным причинам. Особенно это относится к длительно эксплуатирующемуся нефтеперерабатывающему, нефтехимическому и химическому оборудованию. В связи с этим, возникает необходимость 100% обследования такого оборудования с целью определения потенциально опасных зон и принятия соответствующих решений для исключения катастрофических отказов. Качество решения этих задач определяет величину потерь при авариях,

стоимость проведения диагностирования, а также периодических и восстановительных ремонтов.

Существенное повышение надежности и безопасности нефтехимического оборудования можно обеспечить разработкой и использованием методов и средств искусственного интеллекта, в частности ИС. ИС могут быть применены для интерпретации условий и параметров эксплуатации, последующего обоснования программы технического диагностирования, интерпретации параметров диагностирования, которые обеспечат качественную оценку технического состояния, а также поддержки проведения экспертизы промышленной безопасности (ЭПБ). В рамках диссертационной работы была решена задача создания продукционной БЗ для интеллектуализации следующих этапов процедуры ЭПБ: «разработка программы ЭПБ», «анализ (в том числе и интерпретация) результатов» и «принятие решений по ремонту».

При этом использовались следующие концептуальные модели: деревья событий, диаграммы Исикавы, диаграммы UML, а также концепт-карты. Выбор нескольких концептуальных моделей был обусловлен не только исследовательским характером выполняемых работ, но и многоаспектностью объекта исследования. Практика показала, что деревья событий позволяют наглядно моделировать динамику систем, диаграммы Исикавы – причинно-следственные отношения, концепт-карты – построить детальные модели отдельных понятий, а диаграммы классов UML – отразить структурный аспект предметной области (Рисунок 10). Процесс создания БЗ соответствует рисунку 7, при этом для каждого вида концептуальных моделей были разработаны свои конверторы.

Благодаря использованию разработанного инструментария удалось не только обеспечить возможность использования разных видов концептуальных моделей, но и реализовать автоматическую кодогенерацию. По результатам проекта из полученных моделей был сформирован и опубликован набор данных¹⁰, созданы БЗ для деградиционных процессов водородное охрупчивание и коррозионная усталость. Объем последней составил 14 шаблонов фактов, 12 шаблонов правил, 21 правило (622 строки CLIPS кода).

2) Применение модельных трансформаций для создания проблемно-ориентированного редактора БЗ (рег.№ 2012614093). Одной из задач поддержки ИАС «Экспертиза ПБ» являлось обеспечение возможности наполнения созданных БЗ конечными пользователями, которые в общем случае не являются специалистами в области инженерии знаний или программирования. При этом модельные трансформации использовались при создании CFM (Content Formal Model) спецификаций для генерации CRUD интерфейса пользователя не только в рамках ИАС «Экспертиза ПБ», но и для проблемно-ориентированного редактора, созданного на основе РКВД.

Назначение данного редактора – предоставить ограниченную среду разработки с предварительно формализованными структурами предметной области, которые бы описывали:

- основные предметные понятия в форме шаблонов фактов: «событие», «материал», «воздействующие факторы», «нежелательный процесс» и др.;

¹⁰ ISI Models, Mendeley Data, v1, 2019. <http://dx.doi.org/10.17632/f9h2t766tk.1>

- основные предметные отношения между понятиями в форме шаблонов правил, в частности, связь между нежелательными состояниями.

3) Прототипирование продукционных и прецедентных ИС на основе модельных трансформаций, в том числе для задач выявления причин повреждений и разрушения элементов технических системы, а также подбора конструкционного материала на этапе проектирования изделия.

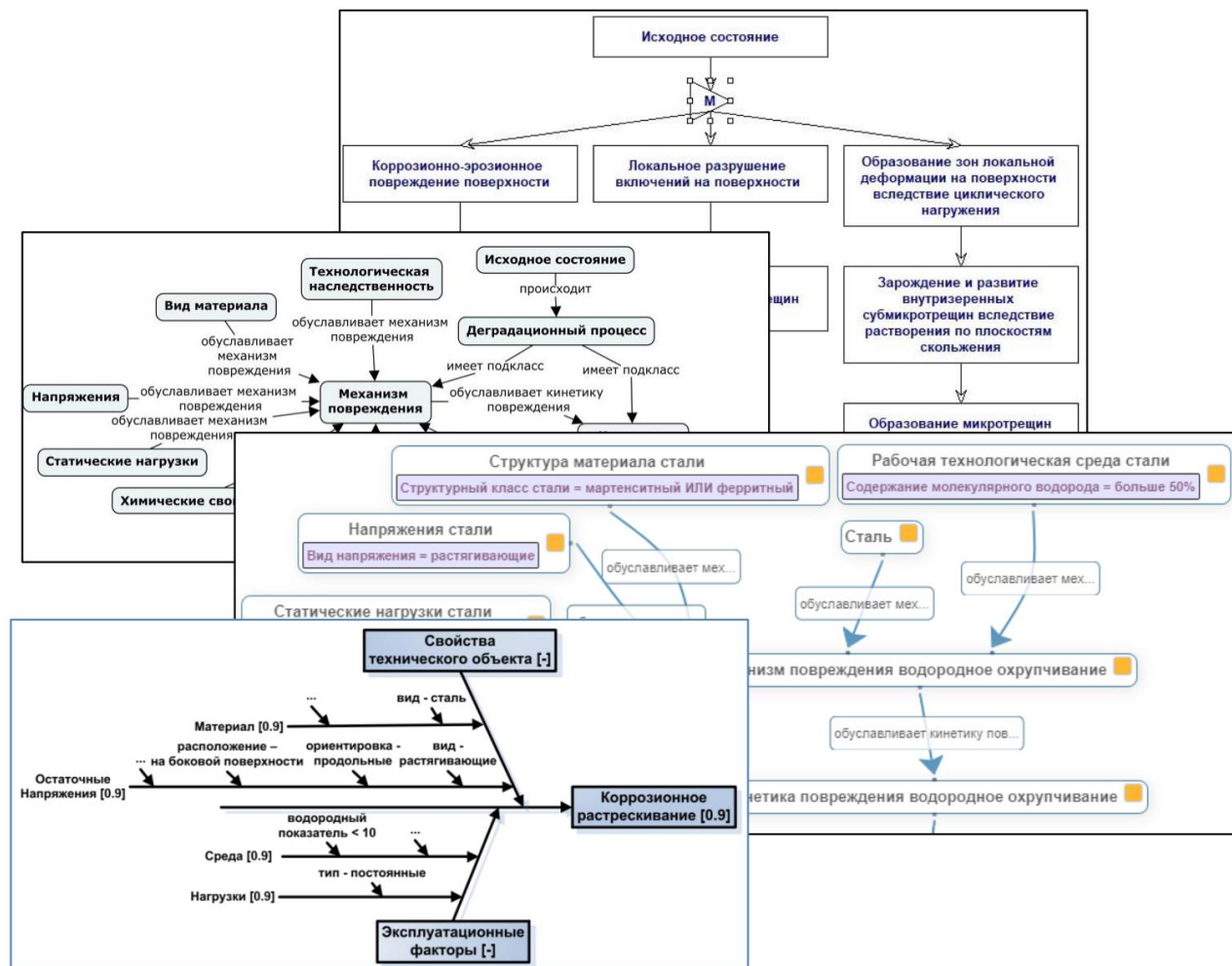


Рисунок 10 – Фрагменты диаграмм, разработанных для БЗ ИАС «Экспертиза ПБ»

Одной из конструкторских задач в области машиностроения, в частности, при проектировании нового изделия (детали), является задача выбора конструкционного материала. Это многовариантная задача, так как для изготовления какой-либо детали можно выбрать некоторое множество материалов, удовлетворяющих условиям ее эксплуатации, изготовления и обслуживания в составе конструкции или машины. Как правило, при этом необходимо учитывать целый ряд условий: назначение детали, условия эксплуатации, технологию изготовления и ремонта, стоимость и доступность и др., которые неоднозначно оцениваются специалистами, принимающими решения.

В рамках диссертационного исследования задача подбора конструкционного материала решалась путем создания прецедентной ИС, в качестве концептуальных моделей использовались UML диаграммы классов, в качестве инструментария - PKBD.

4) Применение модельных трансформаций в интеллектуальном планировщике ИС «INFOT-3» (рег.№ 2007613715) для динамического формирования плана анализа отказов нефтехимического оборудования. Планы формировались с использованием продукционного подхода. Последовательность этапов процесса создания БЗ планировщика совпадает с основными этапам метода и методики, предложенных диссертантом. В качестве вычислительно-независимой модели использовалась модель динамики технического состояния аппарата или конструкции¹¹, представленная в теоретико-множественной форме. Далее, на основе модели динамики осуществлялось построение диаграммы (графа) переходов состояний (Рисунок 11), описывающей возможные последовательности этапов алгоритма.

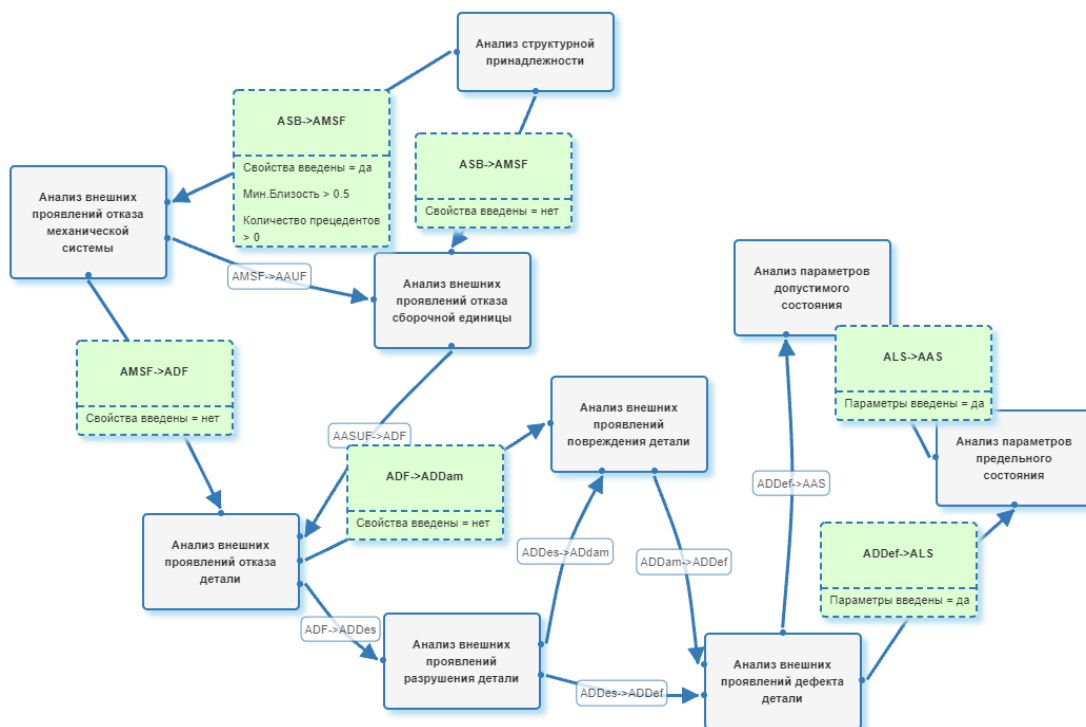


Рисунок 11 – Фрагмент диаграммы (графа) переходов состояний, описывающей алгоритм анализа отказов

В свою очередь диаграмма переходов преобразовывалась в продукционную модель, для представления которой использовались RVML диаграммы, отображаемые в программные коды на CLIPS для анализ отказа трубопровода подачи синтез-газа в колонну синтеза аммиака при решении задачи генезиса.

5) Применение модельных трансформаций для создания систем идентификации технических состояний конструкций с помощью программы-оболочки «E-INFOT» (рег.№ 2005611217). «E-INFOT» реализует прецедентный и продукционный подходы, целевые платформы – СУБД Cache и CLIPS. В качестве исходных концептуальных моделей использовались UML диаграммы классов. Для реализации трансформаций использовалось стороннее средство Rose-Cache-Link. А также был разработан конвертор для формата MDL (IBM Rational Rose), который обеспечил формирование структур БЗ на CLIPS.

¹¹ Берман А.Ф. Пространство технических состояний уникальных механических систем / А.Ф. Берман, О.А. Николайчук // Проблемы машиностроения и надежности машин. – 2007. – №1. – С.14–22.

б) Применение модельных трансформаций при прототипировании онтологий для задач ЭПБ. В качестве исходных данных использовались электронные таблицы, извлеченные из отчетов по ЭПБ. При этом были решены следующие подзадачи: извлечение исходных электронных таблиц с произвольной компоновкой и преобразование их в оригинальную каноническую форму; формализация и представление в виде фрагментов модели предметной области, извлеченных из канонизированных таблиц данных, и объединение фрагментов в полную модель предметной области; синтез кодов БЗ на целевом языке представления знаний из полной модели предметной области. В рамках диссертационного исследования решались последние две подзадачи.

Для создания онтологии использовался набор данных из 216 таблиц, извлеченных из 6 отчетов по ЭПБ, 173 таблицы из набора имели уникальную компоновку и содержали 5817 ячеек. Была отобрана 161 таблица с уникальным содержанием и преобразована в каноническую форму, получившийся набор данных был зарегистрирован под именем ISI-161¹². В результате преобразования канонических таблиц из набора ISI-161 с помощью PKBD было получено 429 сущностей, включая 59 понятий (классов), 338 свойств (атрибутов) и 32 отношения (ассоциации), формирующих отдельные фрагменты модели предметной области. Необходимо отметить, что в результате экспертной оценки только 56% из извлеченных сущностей были отобраны для дальнейшей обработки. Объединение полученных фрагментов в полную концептуальную модель сократило их количество до 242 сущностей, включая 25 понятий, 196 свойств и 21 отношение.

Для содержательной (качественной) оценки полученной модели был использован набор моделей ЭПБ ISI-Models (разработанный ранее). При сравнении моделей наборов ISI-161 и ISI-Models было обнаружено, что 17% (69 из 400) понятий из набора ISI-Models могут быть соотнесены с понятиями, полученными в результате анализа набора данных ISI-161, включая сущности, свойства и отношения. Качественная оценка итоговой модели показала, что ее использование для построения БЗ без дополнения предметной информацией (например, из ISI-Models) затруднительно. Был проведен эксперимент по дополнению полученной модели, в результате которого процент соответствия понятий в наборах достиг 24% (106 из 400).

В дальнейшем, полученные концептуальные модели (понятия и отношения) использовались для синтеза программных кодов прототипов онтологий и БЗ ИАС «Экспертиза ПБ».

Была произведена оценка правильности произведенных трансформаций с формальной точки зрения (т.е. учитывая только синтаксический аспект) с использованием классических формул определения полноты, точности и F-меры (Таблица 2).

Таблица 2 – Результаты оценки трансформаций

Трансформация	Оценка		
	Recall	Precision	F1
Канонические таблицы в концептуальные модели	0.96	0.97	0.97
Концептуальные модели в программные коды	0.99	0.99	0.99

¹² ISI-161: Spreadsheet tables, Mendeley Data, v1, 2019. <http://dx.doi.org/10.17632/8zdyng4y96.1>

Оценки показали, что с помощью предлагаемого подхода удается преобразовать большинство канонических таблиц и концептуальных моделей. В результате анализа таблиц из отчетов по ЭПБ удалось создать концептуальные модели, полезность которых с содержательной (семантической) точки зрения была оценена как 17% и 24%. Основная причина таких оценок – разный акцент (направленность) используемых наборов данных: набор данных ISI-161 содержит информацию о результатах технической диагностики оборудования, в то время как ISI-Models ориентирован на описание (моделирование) всей процедуры ЭПБ, включая такие задачи, как разработка программы исследования, анализ и интерпретация результатов диагностики, а также принятие решений по ремонту и формирование заключения (отчета). По этой причине 173 сущности из ISI-161 в дальнейшем не использовались для построения БЗ.

Общим недостатком трансформации концептуальных моделей и генерации программных кодов является ограничение на сложность создаваемых логических правил, которые не поддерживают такие элементы, как переменные, вычисляемые выражения, функции и т.д. Однако такие структуры могут быть добавлены после синтеза кодов при их отладке и интеграции.

Применение предлагаемых в диссертации методов и программных систем позволило повысить эффективность решения данных задач за счет использования визуального программирования, концептуальных моделей, автоматической кодогенерации и вовлечения в процесс разработки конечных пользователей.

Седьмая глава посвящена оценке эффективности применения языков, методов и средств по временному критерию в сравнении с классическим методом разработки продукционных БЗ, который не предусматривает автоматическую генерацию кода БЗ и использование трансформации концептуальных моделей. Оценка включала косвенный и прямой способ. Косвенный способ состоял в использовании информации о трудоемкости выполнения отдельных этапов разработки ЭС из литературных источников, в частности¹³. Согласно которому, разработка ЭС стандартным методом в среднем продолжается от 4 до 9 месяцев. При применении подходов, основанных на модельных трансформациях, из процесса разработки исключается этап реализации, т.к. используется автоматическая кодогенерация, что обеспечивает сокращение времени разработки ЭС на 1-2 месяца. Кроме того, использование конверторов концептуальных моделей сокращается время этапов получения, структурирования и формализации знаний на 25%, т.е. на 0,5-1,5 месяца. Итоговое сокращение времени составит – от 1,5 до 3,5 месяцев (37,5-38%). Прямой способ состоял в решении учебных задач на базе ИрНИТУ и показал, что эффективность в отдельных случаях в среднем может быть повышена на 48-50% (в зависимости от типа исходной концептуальной модели) благодаря использованию концептуального моделирования, в совокупности с генераторами и интерпретаторами кодов (Рисунок 12).

В целом, результаты косвенной и прямой оценки эффективности позволяют сделать вывод о возможности увеличения скорости разработки ИС/ЭС и БЗ с помощью предлагаемых методов и средств. Однако, данный показатель не учитывает временные затраты необходимые на их изучение, а также сложность создаваемых БЗ.

¹³ Джексон П. Введение в экспертные системы / П. Джексон. – М: Вильямс, 2001. – 624 с.

В приложениях приведены акты и справки, подтверждающие использование результатов диссертационного исследования, а также описание применения разработанных методов и средств в других областях, а именно:

1) При разработке базы знаний модуля идентификации лицевых признаков «EmSi-Interpreter» для программной системы «HR-Robot» (ООО «Смарт Технологии»). Основное назначение «HR-Robot» – поддержка принятия решений при отборе кандидатов на вакансии и проверка персонала на мотивацию (исследование психологической ситуации в коллективе) на основе анализа эмоций по видео.

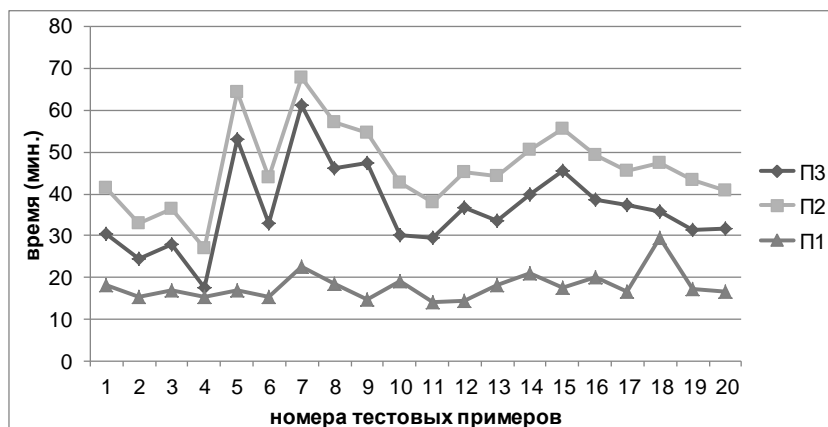


Рисунок 12 – Результаты оценки временных затрат на разработку БЗ разными подходами: П1 – концептуальное моделирование + KBDS/PKBD; П2 – концептуальное моделирование + CLIPSWin; П3 – CLIPSWin

В качестве исходной информации использовались диаграммы классов UML, а также экспертная информация, которая в дальнейшем была представлена в форме таблиц решений специального вида (Рисунок 13), со следующими особенностями: возможностью включения столбца с именами правил в структуру таблицы; возможностью индикации зависимых столбцов символом "#"; возможностью задания составных имен столбцов, включающих имя сущности (или имя класса) и имя ее свойства, разделенные строкой "::"; отсутствием ограничений на значения в ячейках, т. е. они могут состоять не только из набора значений {да, нет}, представляя собой вырожденную форму таблиц истинности, но позволяют использовать конкретные произвольные значения в качестве значений ячеек, а не только значения, указывающие на наличие или отсутствие определенного свойства (компонента) в структуре правила.

	1	2	3	4	5	6	7	8	9
1	RuleName	Risk::grade	Risk::kind	...	Flood-haz:...	#Conclusion::grade	#Conclusion::text	#Conclusion::cf	
2	Risk+Flood hazard->	low	natural		low	low	safe situation	0.9	

Рисунок 13. Пример фрагмента таблицы решений

В рамках решаемой задачи таблицы решений содержали информацию о комбинациях признаков, описывающих эмоции, например, «страх». При этом каждая строка таблицы представляет собой логическое правило. Далее, с использованием РКВД было произведено преобразование таблиц решений в логические правила с возможностью их отображения в виде RVML диаграмм для дальнейшего уточнения и генерации программного кода.

Было создано несколько вариантов БЗ для идентификации определенных эмоций, максимальный объем которых достигал 225 правил с учетом Action Units и

интенсивности их проявления, листинг программного кода на CLIPS составил 1327 строк.

Оценка эффективности решения задачи осуществлялась с двух точек зрения: 1) эффективность создания БЗ; 2) эффективность решения задачи распознавания эмоций. Эффективность создания БЗ оценивалась на примере сравнения временных затрат при использовании предлагаемого подхода с ручным кодированием и визуальным программированием при помощи PKBD. Его результаты представлены на рисунке 14, согласно которым использование таблиц решений может уменьшить временные затраты по сравнению с ручным кодированием (в среднем до 52%), при этом, они менее эффективны в среднем на 38,8% чем использование специализированных мастеров PKBD или манипулирования графическими примитивами (например, RVML диаграммами).

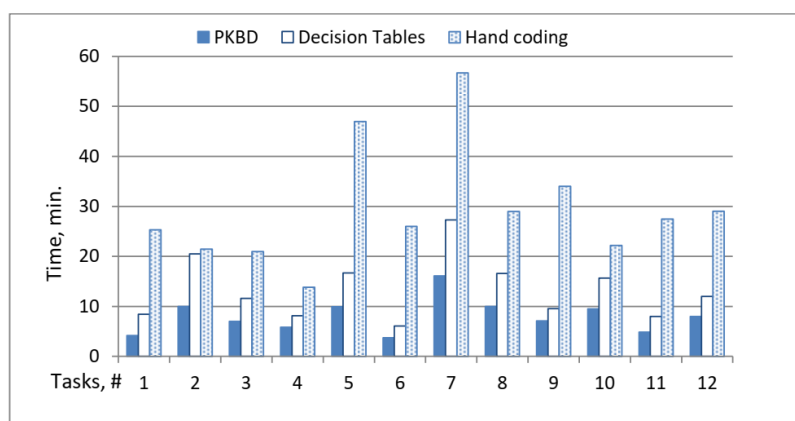


Рисунок 14 – Результаты сравнения PKBD, таблицы решений и ручное кодирование по временному критерию

Однако, позже было определено, что при размере баз знаний более 150 правил и автоматической генерации таблиц решений, например, на основе баз данных, этот подход оказался более предпочтителен.

Оценка эффективности распознавания эмоций с использованием разработанных БЗ осуществлялась на двух наборах данных: тестовом видео (студийные записи с актерами, явно выражающими эмоции) и интервью реальных респондентов (съемка с помощью камеры ноутбука или смартфона реального человека). В первом случае оценка составила 65%, во втором – 20%. Основные причины достаточно низких показателей: необходимость калибровки системы, т.е. настройки на определенного респондента в части индивидуальной интенсивности проявления эмоций; необходимость повышения качества видео за счет минимизации теней, бликов, дрожания камеры при съемке.

2) При разработке базы знаний модуля обнаружения нежелательных сообщений «Детектор» (рег.№ 2020614257) для телекоммуникационной платформы «СМС-Органайзер» (рег.№ 2022613512) (ООО «ЦентраСиб»). Назначение БЗ и модуля – обнаружение сообщений, нарушающих положения 38-ФЗ «О рекламе»¹⁴, а также отправляющих их абонентов.

При решении задачи была построена концептуальная модель минимальным набором ключевых предметных понятий: «СМС-сообщение» (Message), «ключевые

¹⁴ Федеральный закон «О рекламе» [Электронный ресурс]. – Режим доступа: <https://base.garant.ru/12145525/> (дата обращения: 18.03.2022).

слова» (Keyword) или, так называемые, «маркеры» СПАМ-сообщений, отправитель сообщений (Sender). Для выделения комбинаций маркеров (ключевых слов) был произведен анализ базы данных из 1366490 сообщений и отобраны 829 сообщений клиентов, уличенных ранее в рассылке СПАМ-сообщений. Выделенные сообщения сформировали обучающую выборку (тестовый набор), который был использован при построении платформу-независимых моделей, в том числе в форме таблицы решений. В дальнейшем средствами РКВД было синтезировано 6871 строк РНР кода, описывающих 487 правил. При этом каждое логическое правило было представлено в форме условного оператора, анализирующего вхождение определенных ключевых слов в отправляемое сообщение.

Тестирование разработанного модуля «Детектор» произведено с использованием выделенной ранее базы сообщений. В частности, после модификации база знаний модуля составила 498 правил, которые обеспечили обнаружение 653 нежелательных сообщений из тестового набора, таким образом, точность (ассигасу) составила 0,83. Среднее время выполнения (быстродействие) составило 0,00026 с. В результате проверки подсистемой «Детектор» полного исходного набора из 1366490 сообщений было выявлено 1145 СПАМ-сообщений и 25 клиентов (отправителей), нарушивших положения 38-ФЗ «О рекламе», но которые ранее не были обнаружены модераторами.

3) При прототипировании баз знаний для системы поддержки технического персонала (ИС «АвиаТехПом») при поиске и устранении неисправностей системы электроснабжения Сухой Суперджет. Задача решалась во взаимодействии с сотрудниками Иркутского филиала МГТУ ГА.

Диагностика неисправностей воздушных судов продолжает оставаться актуальной проблемой, требующей для своего оперативного решения применения технологий семантического веба (в частности, онтологий), а также методов искусственного интеллекта в форме продукционных ИС.

Основой для формирования продукционной БЗ являлись сценарии работ по локализации и ликвидации неисправностей. Для их описания использовался аппарат деревьев событий с последующим преобразованием полученных деревьев в код.

При решении задачи был проанализирован раздел №24 технической документации по устранению неисправностей, связанный с электроснабжением Сухой Суперджет (RRJ-95), где рассматриваются неисправности 7 подсистем. По результатам анализа сформирована онтология диагностики неисправностей воздушного судна, содержащая 355 понятий и 1066 аксиом; разработаны деревья событий, описывающие последовательности работ по устранению конкретных неисправностей; синтезированы БЗ продукционного типа. Использовались программные средства, интегрированные «по данным»: Protégé, Extended Event Tree Editor (EETE) и РКВД. Интеграция «по данным» реализована путем модельных трансформаций.

4) При прототипировании баз знаний сервиса для анализа и прогнозирования риска (опасности) лесного пожара на основе информации о классе пожароопасности лесов, метеоусловий и других факторов в рамках проекта Министерства науки и высшего образования Российской Федерации грант № 075-15-2020-787 «Фундаментальные основы, методы и технологии цифрового мониторинга и прогнозирования экологической обстановки Байкальской природной территории».

В качестве исходных данных использовалась информация о термальных точках (пикселах горения), выявленных по спутниковым данным с использованием программного комплекса FireProc, созданного в ИСЗФ СО РАН. Термальные точки рассматривались как возможные лесные пожары на Байкальской природной территории. Данные охватывают период с 2017-2020 гг. и включают более 45 000 записей с информацией о координатах термальных точек, дате обнаружения и площади распространения. Для решения задач определения причин возникновения пожаров и прогнозирования их динамики в рамках работы осуществлен сбор и подготовка данных о погодных условиях, а также использовалась информация об инфраструктуре (дороги, населенные пункты), лесничествах, лесных участках и дачах, типе растительности и классах пожароопасности лесов.

В ходе применения технологии была построена концептуальная модель предметной области в форме UML диаграммы классов, описывающая факторы, влияющие на класс пожарной опасности лесного участка и риск (вероятность) возникновения пожара. Далее были разработаны таблицы решений, описывающие структурный аспект предметной области. Эти таблицы содержат информацию о сочетаниях признаков, описывающих класс пожарной опасности лесного участка и риск (опасность) лесного пожара. С помощью PKBD таблицы решений были преобразованы и представлены в виде логических правил, которые были уточнены в форме диаграмм RVMML с дальнейшей генерацией программных кодов на CLIPS и PHP.

5) При создании программных компонентов для трансформации концептуальных моделей, в том числе: диаграмм классов UML, разработанных в CASE-средстве IBM Rational Rose и представленных с использованием стандарта XMI; концепт-карт (карт знаний/интеллект-карт), разработанных в редакторе IHMC StarTools и представленных с использованием стандарта XTM; деревьев событий (ДС), разработанных в графическом редакторе TreeEditorET и сохраненных в XML-подобном формате.

При разработке конверторов использовалось программное средство KBDS, обеспечивающее визуальное программирование трансформаций и содержащее предварительно разработанные модули анализа XML файлов и генерации кодов на CLIPS. Сравнение времени разработки программ-конверторов в KBDS и путем их прямого программирования на языке высокого уровня показала значительную разницу (в 5-7 раз), которая обусловлена необходимостью создания в KBDS только модели трансформации с последующей ее интерпретацией.

Разработанные языки, методы и средства показали свою применимость для разработки интеллектуальных прикладных систем конечными пользователями. Универсальность языков, методов и средств позволила решать задачи диагностики и прогнозирования в разных предметных областях. При этом ключевыми факторами, определяющими эффективность, являются: возможность визуального программирования причинно-следственных зависимостей, повторное использование разработанных ранее концептуальных моделей, автоматическая кодогенерация, а также интерпретация результирующих кодов и спецификаций в специализированных инструментальных системах.

ЗАКЛЮЧЕНИЕ

Важная научно-техническая проблема повышения эффективности создания интеллектуальных систем с декларативными базами знаний, в том числе

конечными пользователями, не теряет свою актуальность. Для ее решения в данной работе предложена технология, при разработке которой были получены следующие новые результаты:

- Метод проектирования декларативных баз знаний интеллектуальных систем, расширяющий принципы модельно-ориентированного подхода за счет новых моделей, языков и платформ в контексте инженерии знаний;

- Методы проектирования программ трансформаций концептуальных моделей и программных компонентов-конверторов концептуальных моделей, отличающиеся от известных использованием оригинального языка описания моделей трансформаций и принципов визуального программирования;

- Визуальный язык программирования продукционных баз знаний – RVML (Rule Visual Modeling Language), основанный на UML и расширяющий его выразительные способности в контексте моделирования декларативных баз знаний;

- Оригинальный текстовый декларативный язык описания моделей трансформаций – TMRL (Transformation Model Representation Language), включающий конструкции для описания не только преобразуемых структур и связей между ними, но и механизма взаимодействия с внешними программными компонентами трансформаций;

- Алгоритмы и архитектура оригинальных программных средств, объединенных общей идеологией модельных трансформаций и формирующих технологическую платформу создания интеллектуальных систем с декларативными базами знаний;

Разработанные методы и средства обеспечивают повышение эффективности обработки данных и знаний на основе модельных трансформаций в контексте создания программного обеспечения систем ИИ конечными (непрограммирующими) пользователями.

СПИСОК ОСНОВНЫХ ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

Монография

1. Юрин А.Ю. Технология создания продукционных экспертных систем на основе модельных трансформаций / Н.О. Дородных, А.Ю. Юрин. – Новосибирск: СО РАН, 2019. – 144 с.

В изданиях, рекомендованных ВАК:

2. Юрин А.Ю. Применение модельно-управляемого подхода для создания продукционных экспертных систем и баз знаний / М.А. Грищенко, Н.О. Дородных, О.А. Николайчук, А.Ю. Юрин // Искусственный интеллект и принятие решений – 2016. – №2. – С.16-29.

3. Юрин А.Ю. Разработка метамоделей для поддержки синтеза нечетких баз знаний / Н.О. Дородных, О.А. Николайчук, А.Ю. Юрин // Вестник компьютерных и информационных технологий. – 2020. – №1. – С. 34-47.

4. Юрин А.Ю. Применение трансформаций таблиц решений при создании интеллектуального программного модуля «Детектор» для веб-приложений / А.Ю. Юрин // Программные продукты и системы. – 2020. – №4. – С. 573-581.

5. Юрин А.Ю. Подход автоматизированной разработки баз знаний на основе трансформации диаграмм Исикавы / Н.О. Дородных, О.А. Николайчук, А.Ю. Юрин // Вестник компьютерных и информационных технологий. – 2018. – №4. – С.41-51.

6. Юрин А.Ю. Разработка интеллектуальных диагностических 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.
7. Юрин А.Ю. Информационно-аналитическая поддержка экспертизы промышленной безопасности объектов химии, нефтехимии и нефтепереработки / А.Ф. Берман, К.А. Кузнецов, О.А. Николайчук, А.И. Павлов, А.Ю. Юрин // Химическое и нефтегазовое машиностроение. – 2018. – №8. – С.30-36.
8. Юрин А.Ю. Средства поддержки моделирования логических правил в нотации RVMML / Н.О. Дородных, С.А. Коршунов, А.Ю. Юрин // Программные продукты и системы. – 2018. – №4. – С. 667-672.
9. Юрин А.Ю. Обеспечение надежности и безопасности химических и нефтехимических производств методами искусственного интеллекта. Часть 2 / А.Ф. Берман, О.А. Николайчук, А.И. Павлов, А.Ю. Юрин // Прикладная информатика. – 2017. – Т.12. – №1(67). – С.26-38.
10. Юрин А.Ю. Использование концепт-карт для автоматизированного создания продукционных баз знаний / Н.О. Дородных, А.Ю. Юрин // Программные продукты и системы. – 2017. – №4. – С. 658-662.
11. Юрин А.Ю. 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.
12. Юрин А.Ю. Обеспечение надежности и безопасности химических и нефтехимических производств методами искусственного интеллекта. Часть 1 / А.Ф. Берман, О.А. Николайчук, А.И. Павлов, А.Ю. Юрин // Прикладная информатика. – 2016. – Т.11. – №5(65). – С.63-75.
13. Юрин А.Ю. Использование диаграмм классов UML для формирования продукционных баз знаний / Н.О. Дородных, А.Ю. Юрин // Программная инженерия. – 2015. – №4. – С.3-9.
14. Юрин А.Ю. Проблемно-ориентированный редактор продукционных баз знаний / А.Ф. Берман, М.А. Грищенко, О.А. Николайчук, А.Ю. Юрин // Программные продукты и системы. – 2015. – №2. – С.13-19.
15. Юрин А.Ю. Поддержка принятия решений при выборе конструкционных материалов для обеспечения безопасной эксплуатации оборудования / А.Ф. Берман, Г.С. Малтугуева, А.Ю. Юрин // Заводская лаборатория. Диагностика материалов. – 2015. - №11. – С.73-80.
16. Юрин А.Ю. 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.
17. Юрин А.Ю. Применение прецедентного подхода для поддержки принятия решений при определении причин и прогнозировании инцидентов и аварий / А.Ф. Берман, О.А. Николайчук, Г.С. Малтугуева, А.Ю. Юрин // Безопасность труда в промышленности. – 2014. – №11. – С.18-26.
18. Юрин А.Ю. Web-сервис для автоматизированного формирования продукционных баз знаний на основе концептуальных моделей / Н.О. Дородных, А.Ю. Юрин // Программные продукты и системы. – 2014. – №4. – С.103-107.

19. Юрин А.Ю. Методы группового выбора для адаптации решений, полученных в результате рассуждений на основе прецедентов / А.Ю. Юрин // Искусственный интеллект и принятие решений. – 2013. – №3. – С.78-85.
20. Юрин А.Ю. Методы и средства автоматизированного построения деревьев событий и отказов / А.Ф. Берман, О.А. Николайчук, Н.Ю. Павлов, А.Ю. Юрин // Автоматизация и современные технологии. – 2013. – №9. – С.8-16.
21. Юрин А.Ю. Редактор баз знаний в формате CLIPS / А.Ю. Юрин, М.А. Грищенко // Программные продукты и системы. – 2012. – №4. – С.83-87.
22. Юрин А.Ю. Интеллектуальная информационная система анализа отказов / А.Ф. Берман, О.А. Николайчук, А.Ю. Юрин // Проблемы машиностроения и надежности машин. – 2012. – №4. – С.88-96.
23. Юрин А.Ю. Автоматизированное построение деревьев отказов и событий на основе модели динамики технического состояния и методов искусственного интеллекта / А.Ф. Берман, О.А. Николайчук, А.Ю. Юрин, Н.Ю. Павлов // Проблемы безопасности и чрезвычайных ситуаций. – 2011. – №1. – С. 40-52.
24. Юрин А.Ю. Применение прецедентного подхода для автоматизированной идентификации технического состояния деталей механических систем / О.А. Николайчук, А.Ю. Юрин // Автоматизация и современные технологии. – 2009. – №5. – С.3-12.
25. Юрин А.Ю. Автоматизация прогнозирования технического состояния и остаточного ресурса деталей уникальных машин и аппаратуры / А.Ф. Берман, О.А. Николайчук, А.Ю. Юрин // Заводская лаборатория. Диагностика материалов. – 2009. – №3. – С.48-57.
26. Юрин А.Ю. Обеспечение безопасности технических объектов методом прецедентных экспертных систем / А.Ф. Берман, О.А. Николайчук, А.Ю. Юрин // Проблемы безопасности и чрезвычайных ситуаций. – 2008. – №5. – С.83-93.
27. Юрин А.Ю. Управление опытом при исследовании динамики технического состояния уникальных машин и конструкций: моделирование опыта / О.А. Николайчук, А.Ю. Юрин // Информационные технологии. – 2008. – №6. – С.30-37.
28. Юрин А.Ю. Интеллектуальная система поддержки принятия решений при определении причин отказов и аварий в нефтехимической промышленности / А.Ф.Берман, О.А.Николайчук, А.И. Павлов, А.Ю. Юрин // Автоматизация в промышленности. – 2006. – №6. – С.15-17.

В изданиях, входящих в базы цитирования Web of Science и Scopus:

29. Yurin A.Yu. Semi-Automated Formalization and Representation of the Engineering Knowledge Extracted From Spreadsheet Data / A.Yu. Yurin, N.O. Dorodnykh, A.O. Shigarov // IEEE Access. – 2021. – Vol. 9. – P. 157468-157481. (SJR Q1)
30. Yurin A.Yu. Personal knowledge base designer: Software for expert systems prototyping / A.Yu. Yurin, N.O. Dorodnykh // SoftwareX. – 2020. – Vol.11. – 100411. (SJR Q2)
31. Yurin A.Yu. Update (4.2020.0303) to “Personal Knowledge Base Designer: Software for expert systems prototyping”, (PII: S2352711019303334) / A.Yu. Yurin, N.O. Dorodnykh, O.A. Nikolaychuk // SoftwareX. – 2021. – Vol.16. – 100825. (SJR Q2)

32. Yurin A.Yu. Designing rule-based expert systems with the aid of the model-driven development approach / A.Yu. Yurin, N.O. Dorodnykh, O.A. Nikolaychuk, M.A. Grishenko // *Expert Systems*. – 2018. – Vol.35. – No. 5. – 12291. (SJR Q2)
33. Yurin A.Yu. An approach for definition of recommendations for prevention of repeated failures with the aid of case-based reasoning and group decision-making methods / A.Yu. Yurin // *Expert Systems with Applications*. – 2012. – Vol. 39. – P. 9282–9287. (SJR Q1)
34. Yurin A.Yu. Intelligent planner for control of failures analysis of unique mechanical systems / A.F. Berman, O.A. Nikolaychuk, A.Yu. Yurin // *Expert Systems with Applications*. – 2010. – Vol. 37. – P. 7101–7107. (SJR Q1)
35. Yurin A.Y. Computer-aided identification of mechanical system's technical state with the aid of case-based reasoning / O.A. Nikolaychuk, A.Y. Yurin // *Expert Systems with Applications*. – 2008. – Vol. 34. – P. 635-642. (SJR Q1)
36. Yurin A.Yu. 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 // *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. – 2014. – Vol. 228. – P.29-38. (SJR Q2)
37. Yurin A.Yu. Application of case-based reasoning and multi-criteria decision-making methods for material selection in petrochemistry / A.F. Berman, G.S. Maltugueva, A.Yu. Yurin // *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications*. – 2018. – Vol. 232. – Issue 3. – P. 204-212. (SJR Q2)
38. Yurin A.Yu. Support of Decision-Making Based on a Production Approach in the Performance of an Industrial Safety Review / A.F. Berman, O.A. Nikolaichuk, A.Yu. Yurin, K.A. Kuznetsov // *Chemical and Petroleum Engineering*. – 2015. – Vol.50. – Issue 1-2. – P.730-738. (SJR Q3)
39. Yurin A.Yu. The rapid development of knowledge bases using UML class diagrams / A.Yu. Yurin, N.O. Dorodnykh, O.A. Nikolaychuk // *International Journal of Computer Aided Engineering and Technology*. – 2021. – Vol.14. – No.1. – P. 39-61.
40. Yurin A.Yu. Creating Web Decision-Making Modules on the Basis of Decision Tables Transformations / A.Yu. Yurin, N.O. Dorodnykh // *Communications in Computer and Information Science*. – 2021. – Vol. 1341. – P. 167-184.
41. Yurin A.Yu. An RVML extension for modeling fuzzy rule bases / N.O. Dorodnykh, A.Yu. Yurin // *CEUR Workshop Proceedings. Proceedings of the 1st International Workshop on Advanced Information and Computation Technologies and Systems*. – 2021. – Vol. 2858. – P. 34-45.
42. Yurin A.Yu. A method and tools for prototyping components of intelligent systems based on transformations / I.V. Bychkov, A.Yu. Yurin // *Journal of Physics*. – 2021. – Vol.1864. – 012042.
43. Yurin A.Yu. End-user development of knowledge bases for semi-automated formation of task cards / N.O. Dorodnykh, Y.V. Kotlov, O.A. Nikolaychuk, V.M. Popov, A.Yu. Yurin // *CEUR Workshop Proceedings. Proceedings of the 3rd International Workshop on Information, Computation, and Control Systems for Distributed Environments*. – 2021. – Vol.2913. – P. 60-73.
44. Yurin A.Yu. Towards designing knowledge bases for aircraft malfunctions diagnostics based on model transformations / Yu.V. Kotlov, V.M. Popov, A.Yu. Yurin // *Journal of Physics*. – 2021. – Vol. 2060. – 012016.

45. Yurin A.Yu. Application of decision tables transformations for prototyping knowledge bases in the case of forest fire risk forecasting / A.Yu. Yurin, O.A. Nikolaychuk, N.O. Dorodnykh // CEUR Workshop Proceedings. Proceedings of the 4th Scientific-practical Workshop Information Technologies: Algorithms, Models, Systems. – 2021. – Vol. 2984. – P. 34-41.
46. Yurin A.Yu. A Reverse Engineering Process for Inferring Conceptual Models from Canonicalized Tables / A.Yu. Yurin, N.O. Dorodnykh // Proceedings of the 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). – 2020. – P. 485-490.
47. Yurin A.Yu. Experimental Evaluation of a Spreadsheets Transformation in the Context of Domain Model Engineering / A.Yu. Yurin, N.O. Dorodnykh // Proceedings of the 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBREIT). – 2020. – P. 388-391.
48. Yurin A.Yu. Technology for Prototyping Expert Systems Based on Transformations (PESoT): A Method / A.Yu. Yurin // CEUR Workshop Proceedings. Proceedings of the 3rd Scientific-practical Workshop Information Technologies: Algorithms, Models, Systems. – 2020. – Vol. 2677. – P. 36-50.
49. Yurin A.Yu. Computer-aided event tree synthesis on the basis of case-based reasoning / A.F. Berman, O.A. Nikolaychuk, A.Yu. Yurin // Advances in Intelligent Systems and Computing. – 2019. – Vol. 875. – P. 3-12.
50. Yurin A.Yu. Application of a Case-Based Approach for Tasks of Industrial Safety Inspection / A.Yu. Yurin, A.F. Berman, O.A. Nikolaychuk, K.A. Kuznetsov // CEUR Workshop Proceedings. Proceedings for the 2nd Scientific-practical Workshop Information Technologies: Algorithms, Models, Systems. – 2019. – Vol. 2463. – P. 32-39.
51. Yurin A.Yu. Towards Ontology Engineering Based on Transformation of Conceptual Models and Spreadsheet Data: A Case Study / N.O. Dorodnykh, A.Yu. Yurin // Advances in Intelligent Systems and Computing. – 2019. – Vol. 1046. – P. 233-247.
52. Yurin A.Yu. Prototyping Rule-Based Expert Systems with the Aid of Model Transformations / A.Yu. Yurin, N.O. Dorodnykh, O.A. Nikolaychuk, M.A. Grishenko // Journal of Computer Science. – 2018. – Vol.14. – No 5. – P. 680-698.
53. Yurin A.Y. A domain-specific language for transformation models / N.O. Dorodnykh, A.Y. Yurin // CEUR Workshop Proceedings. Proceedings for First Scientific-practical Workshop Information Technologies: Algorithms, Models, Systems. – 2018. – Vol. 2221. – P. 70-75.
54. Yurin A.Yu. The domain-specific editor for rule-based knowledge bases / A.Yu. Yurin, A.F. Berman, O.A. Nikolaychuk, N.O. Dorodnykh, M.A. Grishenko // Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). – 2018. – P. 1130-1135.
55. Yurin A.Yu. Fishbone diagrams for the development of knowledge bases / A.Yu. Yurin, A.F. Berman, N.O. Dorodnykh, O.A. Nikolaychuk, N.Yu. Pavlov // Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). – 2018. – P. 1136-1141.
56. Yurin A.Y. Ontology Driven Development of Rule-Based Expert Systems / N.O. Dorodnykh, A.Y. Yurin, A.B. Stolbov // Proceedings of the 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC). – 2018. – P. 1-6.
57. Yurin A.Y. Knowledge bases engineering on the basis of fault trees analysis / A.F. Berman, N.O. Dorodnykh, O.A. Nikolaychuk, A.Y. Yurin // CEUR

Регистрации программ для ЭВМ

58. Юрин А.Ю. Personal Knowledge Base Designer / А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2016617733 от 14.07.16.

59. Юрин А.Ю. Knowledge Base Development System / Н.О. Дородных, А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2019661803 от 09.09.19.

60. Юрин А.Ю. Knowledge Graph Augmentation System / Н.О. Дородных, А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2021664724 от 13.09.21.

61. Юрин А.Ю. Детектор / А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2020614257 от 27.03.20.

62. Юрин А.Ю. Информационно-аналитическая система «Экспертиза промышленной безопасности оборудования химической, нефтехимической и нефтеперерабатывающей промышленности» / А.М. Кузнецов, А.Ф. Берман, К.А. Кузнецов, О.А. Николайчук, А.И. Павлов, А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2016610757 от 19.01.16.

63. Юрин А.Ю. Интеллектуальная программная система автоматизированного построения деревьев событий / А.Ф. Берман, О.А. Николайчук, Н.Ю. Павлов, А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2012614092 от 04.05.12.

64. Юрин А.Ю. Редактор баз знаний CLIPS / М.А. Грищенко, А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2012614093 от 04.05.12.

65. Юрин А.Ю. Программная система для идентификации технического состояния машин и аппаратов / А.Ф. Берман, О.А. Николайчук, А.И. Павлов, А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2007613715 от 31.08.07.

66. Юрин А.Ю. Инструментальное средство создания интеллектуальных систем поддержки принятия решений для идентификации технического состояния деталей машин и конструкций / О.А. Николайчук, А.И. Павлов, А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2005611217 от 25.05.05.

67. Юрин А.Ю. Программный модуль правдоподобного вывода по прецедентам / О.А. Николайчук, А.Ю. Юрин // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2007613714 от 31.08.07.

68. Юрин А.Ю. iDSS.Desktop / А.Ю. Юрин, Н.О. Дородных // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2022617130 от 19.04.22.

69. Юрин А.Ю. СМС-Органайзер / А.Ю. Юрин, Д.Ю. Сопп // Свидетельство о государственной регистрации программы для ЭВМ. – М. – Рег. № 2022613512 от 14.03.22.

70. Юрин А.Ю. Система поддержки принятия решений в задачах группового выбора / Г.С. Малтугуева, И.А. Наумов, А.Ю. Юрин // Свидетельство о

государственной регистрации программы для ЭВМ. – М. – Рег. № 2009614243 от 12.08.09.

Редакционно-издательский отдел

Организация

Адрес организации

Подписано к печати ... 2022 г.

Формат бумаги 60×84 1/16, объем 1 п.л.

Заказ № ... Тираж 100 экз.

Отпечатано в ...