УДК 004.4'22:004.891

ПРОТОТИПИРОВАНИЕ ПРЕЦЕДЕНТНЫХ БАЗ ЗНАНИЙ НА ОСНОВЕ МОДЕЛЬНЫХ ТРАНСФОРМАЦИЙ

Юрин Александр Юрьевич,
канд. техн. наук, заведующий лабораторией, доц.,
e-mail: iskander@icc.ru,
Институт динамики систем и теории управления им. А.М. Матросова СО РАН (ИДСТУ СО РАН),
http://www.icc.irk.ru,
Иркутский национальный исследовательский технический университет (ИрНИТУ),
http://www.istu.edu

В статье рассматривается специализация модельно-ориентированного (управляемого) подхода, обеспечивающая преобразование предметных концептуальных моделей в структуры прецедентной баз знаний, их последующее уточнение и исполнение. Выделяются основные принципы модельно-управляемого подхода, излагается реализация модельных трансформаций в процессе создания интеллектуальных систем и базы знаний. Представлена формализованная постановка задачи прототипирования прецедентных баз знаний. Рассматривается архитектура авторского программного инструментального средства, реализующего методику модельно-управляемого подхода с учетом специфики прецедентных баз знаний предметной области. Описывается технология создания баз знаний на основе модельных трансформаций как последовательное создание и преобразование моделей различной степени абстракции. Излагаются опыт разработки прецедентных экспертных систем и баз знаний с применением модельно-управляемого подхода и наработки в области трансформации моделей.

Ключевые слова: базы знаний, модельные трансформации, модельно-управляемый подход, инструментальное средство, интеллектуальные системы

PROTOTYPING OF CASE BASES ON THE BASIS OF MODEL TRANSFORMATIONS

Yurin A.Yu.,
Ph.D, head of laboratory,
e-mail: iskander@icc.ru,
Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of RAS,
http://www.icc.irk.ru,
Irkutsk National Research Technical University,
http://www.istu.edu

The paper describes a specialization of a model-driven development approach for transformation of subject domain conceptual models to the structures of a case base, their subsequent refinement and execution. The basic principles of the model-driven development approach are highlighted, the implementation of model transformations in the process of creating intelligent systems and knowledge base is stated. The formalized statement of the problem of prototyping of case bases is presented. The architecture of the author's software that implements the methodology of the model-driven approach, taking into account the specifics of the case bases of the subject domain is considered. The technology for knowledge bases engineering on the basis of consecutive transformations of models with various degree of abstraction is presented. Experience of development of case-based expert systems and case bases with application of model-driven approach and developments in the field of model transformations is stated.

Keywords: knowledge bases, model transformations, model-driven approach, software, intelligent system

DOI 10.21777/2500-2112-2019-2-47-58

1. Введение

Создание баз знаний интеллектуальных систем продолжает оставаться трудоемким процессом, требующим привлечения не только высококвалифицированных специалистов в различных областях (экспертов предметной области, аналитиков, программистов), но и использования специализированного инструментария. Перспективным в данной области является разработка и использование программных инструментальных средств, реализующих принципы визуального программирования и автоматической генерации кодов. Примерами подобных систем могут являться Visual Expert System Designer, Expert System Designer, Expert System Creator, Exsys (Expert System Development Tool) Corvid, ClipsWin, ES-Builder Web, VisiRule, Generic Knowledge Base Editor и др. В большинстве случаев данные системы ориентированны на определенный формализм представления знания, например, продукции. Другим классом систем, способным решать задачи разработки баз знаний с использованием визуального программирования являются системы, обеспечивающие построение концептуальных моделей в форме концепт-карт, карт знаний, диаграмм «сущность-связь», древовидных семантических структур (схем) или UML-моделей. Большинство из подобных систем являются универсальными средствами моделирования и не учитывают специфику создания интеллектуальных систем, т.е. используемые в данной области формализмы, языки и платформы. Соответственно, обеспечивая наглядное визуальное представление структур знаний предметной области, они не обеспечивают их адекватную кодификацию (формализацию) на существующих языках представления знаний. Компромиссом в данном вопросе является использование методов и подходов, реализующих принципы модельно-ориентированного (управляемого) подхода [9, 11]. Подобные решения [3] обеспечивают анализ и преобразование предметных концептуальных моделей в структуры баз знаний, их последующее уточнение и исполнение. В ходе обработки моделей происходит реализация модельных трансформаций [10]. В ряде случаев для этого используются специализированные проблемно-ориентированные языки, например, TMRL (Transformation Model Representation Language) [12].

В данной работе предлагается рассмотреть пример применения подобного инструментального средства – Personal Knowledge Base Designer (PKBD) [4] для прототипирования баз знаний прецедентного типа. При этом в качестве входных моделей используются UML диаграммы классов и произведенная ранее формализация подхода [13].

2. Состояние вопроса

В процессе рассмотрения состояния вопроса представим основные понятия и определения в области прецедентного подхода, модельных трансформаций, а также имеющийся задел.

2.1. Прецедентный подход

Прецедентный подход или Case-Based Reasoning (CBR) [8] – это методология решения задач путем повторного использования и адаптации (при необходимости) решений, которые были получены ранее при решении подобных задач. Данная методология базируется на принципе принятия решений «по аналогии». Основным понятием является прецедент – структурированное представление накопленного опыта в виде данных и знаний, обеспечивающее его последующую автоматизированную обработку при помощи специализированных программных систем. Приведем основные характеристики прецедентов:

- прецедент представляет особое знание, привязанное к контексту, что позволяет использовать знания на прикладном уровне;

- прецеденты могут принимать различную форму (вид): охватывая разные по продолжительности промежутки времени; связывая решения с описаниями проблем; результаты с ситуациями и т.д.;

- прецедент фиксирует только тот опыт, который может обучить (быть полезным), фиксируемые прецеденты потенциально могут помочь специалисту (ЛПР) достичь цели, облегчить ее формулирование в будущем или предупредить его о возможной неудаче или непредвиденной проблеме.

Прецедент позволяет структурировать единицы опыта, при этом выбор структуры зависит от задач, при решении которых необходимо обеспечить повторное использование опыта. Общая структура прецедента включает два основных компонента:

- идентифицирующая (характеризующая) часть (или описание проблемы) – описывает опыт таким способом, который позволяет оценить возможность его повторного использования в определенной ситуации;

- обучающая часть – описывает урок (обучающее знание, решение) как часть единицы опыта, например, решение проблемы или его часть, доказательство (вывод) решения, альтернативные или неудавшиеся решения.

Прецедентный подход был применен к решению задач в различных предметных областях, включая: планирование [16], диагностику [2] и др. Среди инструментария, реализующего данный подход, можно выделить такие системы как CBR-Express, CasePoint, CasePower, Esteem, Expert Advisor, ReMind, CBR/text, ReCall, RATE-CBR, S3-Case, INRECA, and CASUEL.

2.2. Модельные трансформации

В последнее время в области программной инженерии появилась тенденция использования подходов, рассматривающих модели не только как артефакты документации (технического задания), но и как центральные артефакты при разработке программного обеспечения, обеспечивая автоматический синтез программных кодов на их основе. Это позволяет значительно сократить время разработки, снизить риск ошибок программирования, вовлечь в процесс создания программной системы непрограммирующих пользователей. В целом эти положения принято относить к области MDE (иначе – Model Driven Development, MDD) [11]. На сегодняшний день примерами реализациями (инициативами) MDE являются: OMG Model Driven Architecture (MDA), Eclipse Modeling Framework (EMF), Model Integrated Computing (MIC), Microsoft Software Factories, JetBrains MPS.

Центральными понятиями MDE/MDD являются:

1. Модель – абстрактное описание на некотором формальном языке характеристик системы (процесса), важных с точки зрения цели моделирования. При этом, как правило, модели визуально описываются на определенном языке моделирования и сериализуются (представляются) на XML.

2. Метамодель – модель языка, используемого для создания моделей (модель моделей).

3. Мета-метамодель – язык, на котором описываются метамодели. Для построения метамоделей используются специальные языки метамоделирования. Наиболее распространенными языками метамоделирования являются: MOF (Meta-Object Facility), Ecore, KM3 (Kernel Meta Meta Model) и др.

4. Четырехуровневая архитектура метамоделирования – концепция определяющая разные слои абстракции (M0-M3), где на самом нижнем уровне (M0) представлены объекты реальности, далее идут уровень моделей (M1), уровень метамоделей (M2) и уровень мета-метамодели (M3).

5. Модельные трансформации – автоматическая генерация целевой модели из исходной в соответствии с набором преобразующих правил. При этом каждое правило описывает соответствия между элементами исходной и целевой метамодели.

Более детальное описание данных концепций приведено в работе [11].

Существует множество работ посвященных модельным трансформациям, в том числе, с целью их классификации. Например, в [15] приводится следующие основания классификации:

- по типу результата: Model-to-Model (M2M); Model-to-Text (M2T) and Text-to-Model (T2M). В данном случае результирующий текст может представлять собой исходные программные коды, документацию, спецификации и др.;
- по типу использованных языков моделирования;
- по уровню абстракции моделей;
- по направлению преобразований.

В настоящий момент существует несколько подходов к реализации трансформаций на основе использования:

- графовых грамматик;
- визуального программирования для создания правил трансформаций;
- стандартов трансформаций (например, Query/View/Transformation);
- гибридных (декларативно-императивных) языков трансформаций (например, ATLAS Transformation);
- языков программирования общего назначения;
- языков трансформации XML документов (например, eXtensible Stylesheet Language Transformations (XSLT)).

В данной работе предлагается рассмотреть применение модельных трансформаций в процессе создания интеллектуальных систем и БЗ на основе реализации MDE/MDD.

2.3. Задел

В качестве основы для создания прецедентных баз знаний на основе модельно-управляемого подхода использовались результаты завершенных разработок с участием автора. В частности опыт разработки прецедентных экспертных систем и баз знаний в нефтехимии [2], а также наработки в области трансформации моделей [12] и использования принципов MDE/MDD для создания продукционных баз знаний и экспертных систем [5].

3. Протипирование прецедентных баз знаний

3.1. Постановка задачи

Задачу прототипирования прецедентных баз знаний на основе модельно-ориентированного подхода можно представить в виде совокупности двух подзадач:

1. Создание модели прецедента, соответствующей текущей проблеме.
2. Создание системы, обеспечивающей преобразование модели прецедента в соответствии с принципами MDA/MDE/MDD.

В рамках решения подзадачи 1 формализуем модель прецедента:

$c_i = \{Prop_i^{Problem}, Prop_i^{Decision}\}$, где c_i – прецедент, $Prop_i^{Problem}$ – описательный блок прецедента, $Prop_i^{Decision}$ – блок решения прецедента. При этом каждый из этих блоков содержит в себе свойства решаемой задачи: $Prop_i^{Problem} = \{p_1 \ldots p_m\}$, $Prop_i^{Decision} = \{p_{m+1} \ldots p_N\}$, $Prop_i^{Problem} \cup Prop_i^{Decision} = Prop$, $Prop_i^{Problem} \cap Prop_i^{Decision} = \varnothing$.

В зависимости от предметных задач возможно создание прецедентов с разным набором свойств в идентифицирующей и обучающих частях. В качестве средства описания структуры прецедентов предлагается использовать UML.

Для решения подзадачи 2 используем формализацию из [13], согласно которой:

$$MDA = \left\langle \begin{matrix} L, CIM, PIM, PSM, PDM, \\ F_{CIM-to-PIM}, F_{PIM-to-PSM}, F_{PSM-to-CODE} \end{matrix} \right\rangle,$$

где L – языковые средства визуального моделирования, $L=\{UML\}$; CIM, PIM, PSM, PDM – соответствующие модели: CIM – Computation-Independent Model (вычислительно-независимая модель), PIM – Platform-Independent Model (платформо-независимая модель), PSM – Platform-Specific Model (платформо-зависимая модель), PDM – Platform Description Model (модель описания платформы). $F_{CIM\text{-to-}PIM}: CIM \rightarrow PIM$, $F_{PIM\text{-to-}PSM}: PIM \rightarrow PSM$, $F_{PSM\text{-to-}CODE}: PSM \rightarrow CODE$ – правила преобразования моделей.

Таким образом, подзадача 2 состоит в том, чтобы специфицировать (конкретизировать, переопределить) эти элементы в контексте создания баз знаний прецедентного типа, т.е. определить MDE^{CBR}:

$$MDA^{CBR} = \left\langle \begin{matrix} L^{CBR}, CIM^{CBR}, PIM^{CBR}, PSM^{CBR}, \\ PDM^{CBR}, F_{CIM-to-PIM}^{CBR}, F_{PIM-to-PSM}^{CBR}, \\ F_{PSM-to-CODE}^{CBR} \end{matrix} \right\rangle.$$

3.2. Методика

Общепринятая технология создания баз знаний и экспертных систем [14] состоит из следующих основных этапов: анализ и идентификация, концептуализация, формализация, реализация, тестирование. В свою очередь, в соответствии с формализацией [13] создание баз знаний на основе модельных трансформаций может быть рассмотрено как последовательное создание и преобразование моделей различной степени абстракции, соответствующих определенным этапам:

- вычислительно-независимой модели, описывающей ключевые абстракции. Для ее описания может быть использован UML;
- платформо-независимой модели, зависящей от предыдущей и учитывающей особенности формализма представления знаний, но независящей от средств программной реализации;
- платформо-зависимой модели, учитывающей особенности средств реализации. Данная модель может быть использована для генерации спецификаций или программных кодов.

3.3. Программное средство

В качестве платформы (программного инструментального средства), реализующей методику, был использован авторский инструментарий – Personal Knowledge Base Designer (PKBD) [4]. PKBD обеспечивает создание предметно-ориентированных редакторов баз знаний посредством реализации основных принципов MDA/MDE/MDD.

Архитектура PKBD включает в себя основные модули для: управления базой знаний, поддержки языков представления знаний, управления словарями, интеграции с CASE-средствами, интерпретации моделей и генерации графического пользовательского интерфейса (GUI). Интерфейс генерируется как в виде всплывающих окон, так и в виде мастеров. Мастеры представляют собой последовательности графических форм, которые сегментируют и упорядочивают процессы ввода и редактирования элементов базы знаний.

Расширение PKBD позволяет создавать базы прецедентов на основе трансформации концептуальных моделей.

Далее рассмотрим применение PKBD как в рамках лабораторного практикума по дисциплине «Инструментальные средства информационных систем», преподаваемой в институте высоких технологий Иркутского национального исследовательского технического университета (ИрНИТУ), так и при решении практической задачи в сфере нефтехимии.

4. Применение программного инструментального средства PKBD

4.1. Решение учебных задач

В рамках решения учебных задач студентам предлагалось произвести проектирование и программную реализацию прототипа прецедентной базы знаний для экспертной системы. В частности, решались следующие задачи [7]:

1. Произвести анализ предметной области, с целью выявления основных понятий и отношений (их идентификации и концептуализации).
2. Формализовать выявленные понятия (понятие) в форме диаграмм классов уровня проектирования в нотации UML.
3. Создать проект базы знаний с использованием специализированного редактора (PKBD).
4. Произвести импортирование диаграммы классов.
5. Уточнить описание шаблона для формирования прецедентов.
6. Описать прецеденты (не менее 10 экз.), согласно шаблону.
7. Проверить работоспособность разработанной базы знаний.
8. Сгенерировать комплект документации (отчет по базе знаний).

Рассмотрим выполнение данных задач на примере одного из вариантов задания.

Предметной областью для создания базы знаний является выбор или подбор настольной игры для покупки в зависимости от необходимых характеристик. Тип ЭС: статическая (решает задачи при неизменяемых в процессе решения данных и знаний). Доступные ресурсы: эксперт, база данных настольных игр с перечнем характеристик.

Цель системы (использования базы знаний): оказать пользователю системы экспертную поддержку принятия решений при решении задачи выбора настольной игры для покупки в зависимости от необходимых характеристик.

Критерии оценки: оценка экспертов, процент правильных решений по сравнению с обучающей выборкой.

В результате анализа предметной области было решено выделить единственное понятие «Настольная игра» (таблица 1) и формализовать его в форме класса UML (рисунок 1). При этом результаты процедуры анализа и концептуализации были рассмотрены как вычислительно-независимая модель. В свою очередь, формализованная модель рассматривалась уже как платформо-независимая, в которой описательную часть прецедента составили все основные свойства класса, а обучающую часть составило только наименование (название) настольной игры.

Таблица 1 – Результат анализа предметной области

Свойство	Описание
Название	Символьное обозначение настольной игры
Жанр	Символьное обозначение жанра настольной игры
Тема/сеттинг	Время, место или условия, в которых происходят события настольной игры, например, средневековье
Цена	Стоимость настольной игры в денежном эквиваленте
Минимальное кол-во игроков	Минимальное количество игроков, требуемое для партии в определенную настольную игру
Максимальное кол-во игроков	Максимальное количество игроков, которые могут принять участие в партии в настольную игру
Время игры	Приблизительное количество времени в минутах, которое уйдет на одну партию в настольную игру
Возраст	Возрастное ограничение, указывающее минимальный допустимый возраст игроков
Сложность	Обозначает уровень сложности правил настольной игры

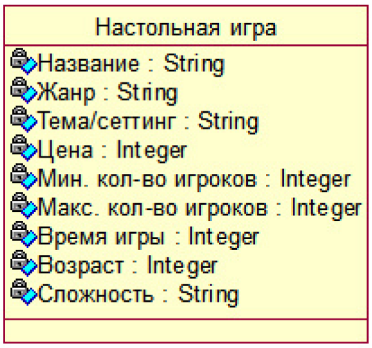

Рисунок 1 – Понятие «Настольная игра»
(CASE-средство IBM Rational Rose)

В результате импортирования построенной модели был создан проект в PKBD, при этом модель прецедента была представлена с учетом инструментальной платформы в форме шаблона для создания прецедентов. Одной из форм сериализации данной модели является использование XML-подобного формата EKB.

В дальнейшем полученная модель могла быть изменена (рисунок 2), а также произведено наполнение базы знаний (рисунок 3). Для автоматизации наполнения может быть использована функция импорта структурированных описаний из CSV файлов.

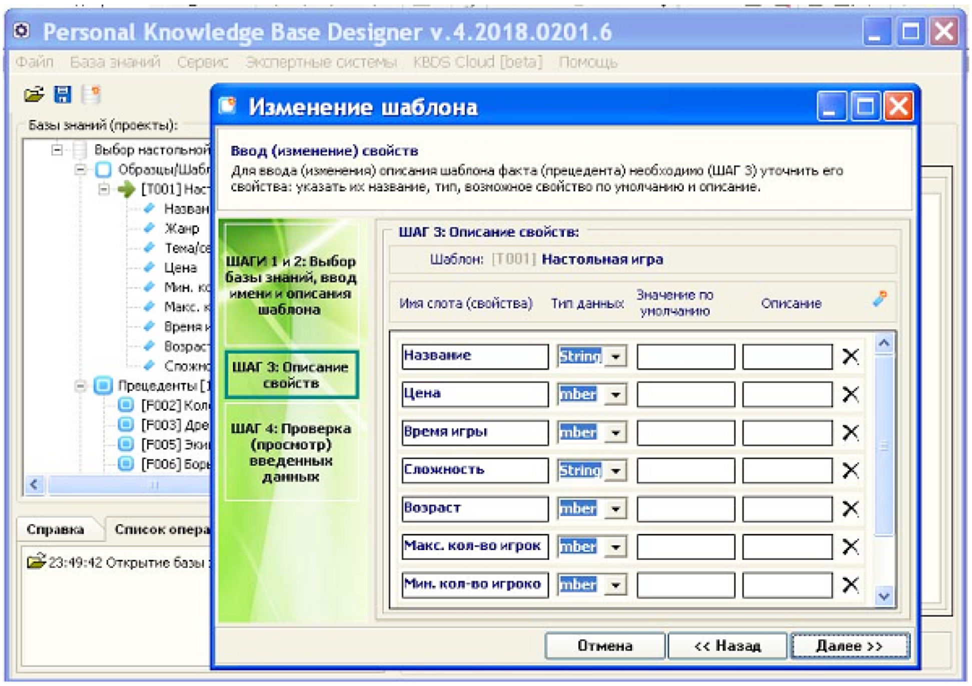

Рисунок 2 – Пример интерфейса PKBD: изменение шаблона прецедента

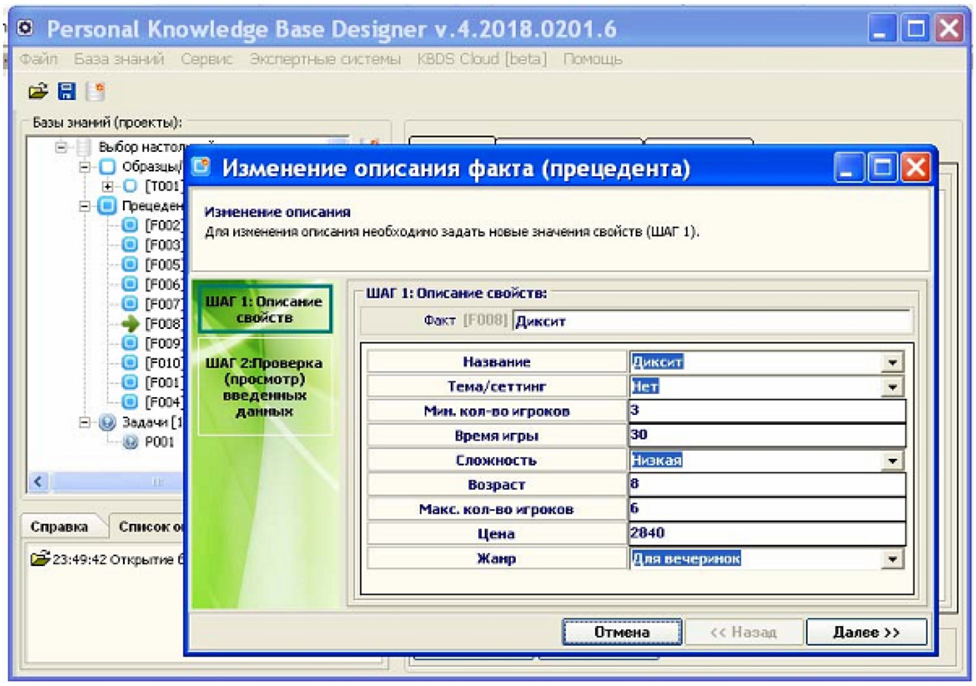

Рисунок 3 – Пример интерфейса PKBD: ввод/изменение прецедента

Далее осуществлялось тестирование полученной базы знаний путем построения запросов (рисунок 4, блок 1) с указанием важности определенных свойств в описательной части прецедентов. С учетом заданной важности происходила оценка степени близости сформированного описания и описаний прецедентов в базе знаний с использованием метрики Журавлева [6] их извлечение (рисунок 4, блок 2). Для каждого извлеченного прецедента возможен детальный просмотр и сравнение описаний (рисунок 4, блок 3).

Рассмотренные выше действия могут быть повторены с целью подбора релевантного решаемой задаче набора свойств и их важности.

Рисунок 4 – Пример интерфейса PKBD: запрос к базе прецедентов, просмотр результатов запроса, детальный просмотр (сравнение) прецедентов

4.2. Прикладная задача

Практической задачей, в рамках которой осуществлялась апробация методики и средства, являлось прототипирование баз знаний для задач автоматизации проведения экспертизы промышленной безопасности (ЭПБ). ЭПБ остается важным мероприятием обеспечения безопасной эксплуатации объектов, в частности, нефтехимии, в рамках которого проводится регулярная оценка технического состояния и расчет остаточного ресурса. Повысить эффективность проведения обследования можно с помощью систем информационно-аналитической поддержки [1]. При этом целесообразным является использование информации о приведенных ранее обследованиях (диагностированиях), представленных в виде прецедентов.

Рассмотрение процедуры ЭПБ [1, 17] показало возможность использования прецедентного подхода при выполнении этапов: «разработка программы ЭПБ», «анализ результатов технического диагностирования», «принятие технических решений по ремонту», «формирования заключения ЭПБ». При этом решение каждой из рассмотренных задач потребовало разработки отдельной модели прецедента (таблица 2), которая формировалась на основе полной концептуальной модели (рисунок 5):

1) Модель прецедента для задачи «разработка программы ЭПБ», обеспечивает связь между понятиями «экспертиза» и «программа» (таблица 2, задача 1). В результате решения данной задачи на основании информации о технической характеристике и данных по эксплуатации объекта экспертизы выбираются подобные (похожие, близкие) объекты, извлекаются программы ЭПБ по этим объектам, которые затем анализируются исследователем и на основе анализа формируется (адаптируется) программа текущей ЭПБ.

2) Модель прецедента для задачи «анализ результатов технического диагностирования», обеспечивает связь между понятиями «экспертиза» и «техническое состояние» (таблица 2, задача 2). В результате решения данной задачи на основании информации о технической характеристике и данных по эксплуатации объекта экспертизы выбираются подобные объекты, извлекаются результаты технического диагностирования по этим объектам, которые затем анализируются исследователем и на основе анализа формируется выводы о причинах возникновения текущего состояния объекта экспертизы.

3) Модель прецедента для задачи «принятие технических решений по ремонту», обеспечивает связь между понятиями «экспертиза» и «решения по ремонту» (таблица 2, задача 3). В результате решения данной задачи, на основании информации о технической характеристике и данных по эксплуатации

ции объекта экспертизы выбираются подобные объекты, извлекаются решения ЭПБ по этим объектам, которые затем анализируются исследователем и на основе анализа формируется (адаптируется) решения по текущей ЭПБ.

4) Модель прецедента для задачи «формирования заключения ЭПБ», обеспечивает связь между понятиями «экспертиза» и «заключение ЭПБ» (таблица 2, задача 4). На основании информации о технической характеристике объекта экспертизы выбирается подобный объект, извлекается заключение ЭПБ по этому объекту, которое принимается в качестве шаблона для формирования заключения по рассматриваемому объекту экспертизы.

Таблица 2 – Описание моделей прецедентов для задач ЭПБ

<table>
<tr><th>Задача</th><th>Описательная часть прецедента</th><th>Обучающая часть прецедента</th></tr>
<tr><td>1. Разработка программы ЭПБ</td><td rowspan="3">Технический объект;
Данные по эксплуатации; Параметры;
Результаты контроля;
Данные о технических диагностированиях;
Сведения о ремонтах;
Технические характеристики объекта;
Технические характеристики;
Параметры эксплуатации</td><td>Программа работ;
Работы;
Виды работ</td></tr>
<tr><td>2. Анализ результатов технического диагностирования</td><td>Заключительная часть экспертизы;
Решения по результатам экспертизы;
Решения по ремонту</td></tr>
<tr><td>3. Принятие технических решений по ремонту</td><td>Заключительная часть экспертизы;
Решения по результатам экспертизы;
Решения по техническому состоянию</td></tr>
<tr><td>4. Формирование заключения ЭПБ</td><td>Технический объект; Технические характеристики объекта; Технические характеристики</td><td>Программа работ;
Работы;
Виды работ;
Заключительная часть экспертизы;
Решения по результатам экспертизы;
Решения по техническому состоянию;
Решения по ремонту;
Данные по эксплуатации;
Параметры;
Результаты контроля;
Данные о технических диагностированиях;
Сведения о ремонтах;
Параметры эксплуатации</td></tr>
</table>

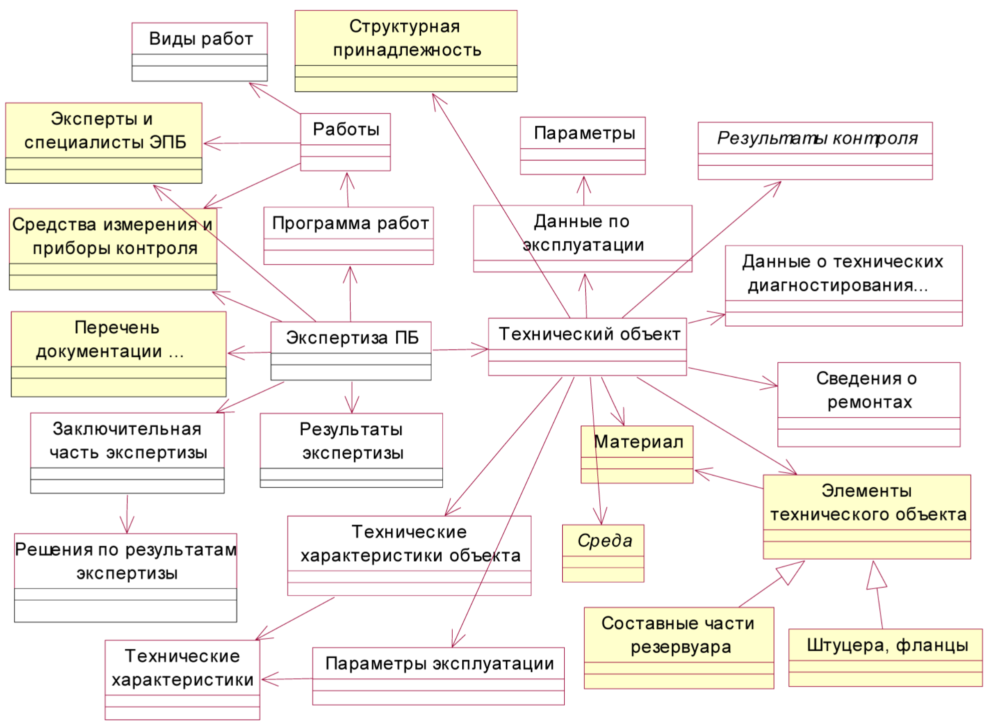

Рисунок 5 – Фрагмент концептуальной модели для задач ЭПБ

Использование PKBD позволило рассмотреть различные варианты моделей прецедентов и подобрать, одобренные экспертами, информационные веса (важность) для свойств.

В дальнейшем на основе данных моделей разрабатывалось алгоритмическое и программное обеспечение информационно-аналитической системы поддержки принятия решений «Экспертиза ПБ» [1].

5. Заключение

Создание баз знаний интеллектуальных систем продолжает оставаться трудоемким процессом, эффективность которого может быть повышена, путем применения подходов, основанных на порождающем программировании и визуализации.

В данной работе рассматривается пример применения принципов модельно-управляемого подхода для прототипирования прецедентных баз знаний на основе модельных трансформаций. В качестве исходных концептуальных моделей использовались UML диаграммы классов. В качестве средства реализации подхода использовался программный инструментарий авторской разработки Personal Knowledge Base Designer (PKBD). Подход был апробирован как при решении учебных задач, так и прикладной задачи поддержки принятия решений при проведении экспертизы промышленной безопасности [1].

В дальнейшем планируется модификация инструментария с целью обеспечения генерации отчуждаемых программных кодов для создания веб-ориентированных встраиваемых экспертных систем.

Список литературы

1. *Берман, А.Ф.* 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.

2. *Берман, А.Ф.* 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.

3. *Грищенко, М.А.* 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.

4. *Грищенко, М.А.* Система программирования продукционных баз знаний: Personal Knowledge Base Designer / М.А. Грищенко, Н.О. Дородных, А.Ю. Юрин // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2016): материалы VI междунар. науч.-техн. конф. / редкол. В.В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР, 2016. – С. 209–212.

5. *Грищенко, М.А.* Инструментальное средство создания продукционных экспертных систем на основе MDA / М.А. Грищенко, О.А. Николайчук, Павлов А.И., Юрин А.Ю. // Образовательные ресурсы и технологии. – 2016. – № 2 (14). – С. 144–151.

6. Распознавание, классификация, прогноз. Математические методы и их применение. Вып. 2. / Под ред. Ю.И. Журавлева. – М.: Наука, 1989. – 302 с.

7. *Юрин А.Ю.* Инструментальные средства информационных систем: Методические указания по выполнению лабораторных работ. – Иркутск: ИРНИТУ, 2018. – 146 с.

8. *Aamodt A., Plaza E.* Case-based reasoning: foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications. – 1994. – vol. 7(1). – pp. 39–59.

9. *Brambilla M., Cabot J., Wimmer M.* Model Driven Software Engineering in Practice. Morgan & Claypool Publishers, 2012.

10. *Czarnecki K., Helsen S.* Feature-based survey of model transformation approaches. IBM Systems Journal. – 2006. – vol. 45(3). – pp. 621–645.

11. *Da Silva A.R.* Model-driven engineering: A survey supported by the unified conceptual model // Computer Languages, Systems & Structures. – 2015. – vol. 43. – pp. 139–155.

12. *Dorodnykh N.O., Korshunov S.A., Pavlov N.Yu., Yurin A.Yu.* Model Transformations for Intelligent Systems Engineering. Open Semantic Technologies for Intelligent Systems. – 2018. – vol.2 (8). – pp. 77–81.
13. *Dorodnykh, N.O., Yurin A.Yu.* About the specialization of model-driven approach for creation of case-based intelligence decision support systems. Open Semantic Technologies for Intelligent Systems. – 2017. – vol. 7. – pp. 151–154.
14. *Jackson, P.* (1999). Introduction to expert systems (3rd ed.). – Harlow: Addison-Wesley.
15. *Mens T., Gorp P.V.* A Taxonomy of Model Transformations // Electronic Notes in Theoretical Computer Science. – 2006. – vol. 152. – pp. 125–142.
16. *Popa A., Wood W.* Application of case-based reasoning for well fracturing planning and execution. Journal of Natural Gas Science and Engineering. – 2011. – vol.3 (6). – pp. 687–696.
17. *Yurin A.Yu., Dorodnykh N.O., Nikolaychuk O.A., Berman A.F., Pavlov A.I.* ISI models, Mendeley Data, v1 (2019). doi: http://dx.doi.org/10.17632/f9h2t766tk.1

References

1. *Berman, A.F.* Support of Decision-Making Based on a Production Approach in the Performance of an Industrial Safety Re-view / A.F. Berman, O.A. Nikolaichuk, A.Yu. Yurin, K.A. Kuznetsov // Chemical and Petroleum Engineering 50(1-2), 730–738 (2015).
2. *Berman, A.F.* Obespechenie bezopasnosti tekhnicheskih ob"ektov metodom precedentnyh ekspertnyh system / A.F. Berman, O.A. Nikolaichuk, A.Yu. Yurin // Problemy bezopasnosti i chrezvychajnyh situacij. – 2008. – № 5. – S. 83–93.
3. *Grishchenko, M.A.* Primenenie model'no-upravlyaemogo podhoda dlya cozdaniya produkcionnyh ekspertnyh sistem i baz znanij / M.A. Grishchenko, N.O. Dorodnykh, O.A. Nikolajchuk, A.Yu. Yurin // Iskusstvennyj intellekt i prinyatie reshenij. – 2016. – № 2. – S. 16–29.
4. *Grishchenko, M.A.* Sistema programmirovaniya produkcionnyh baz znanij: Personal Knowledge Base Designer / M.A. Grishchenko, N.O. Dorodnykh, A.Yu. Yurin // Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nyh sistem = Open Semantic Technologies for Intelligent Systems (OSTIS-2016): materialy VI mezhdunar. nauch.-tekhn. konf. / redkol.: V.V. Golenkov (otv. red.) [i dr.]. – Minsk: BGUIR, 2016. – C. 209–212.
5. *Grishchenko, M.A.* Instrumental'noe sredstvo sozdaniya produkcionnyh ekspertnyh sistem na osnove MDA / M.A. Grishchenko, O.A. Nikolaichuk, A.I. Pavlov, A.Yu. Yurin // Obrazovatel'nye resursy i tekhnologii. – 2016. – № 2 (14). – S. 144–151.
6. *Zhuravlev I.Yu., Gurevitch, I.B.* Pattern recognition and image recognition. In Yu.I. Zhuravlev (Ed.), Pattern recognition, classification, forecasting: Mathematical techniques and their application, issue 2(pp. 5–72). – M.: Nauka (1989).
7. *Yurin, A.Yu.* Instrumental'nye sredstva informacionnyh sistem: Metodicheskie ukazaniya po vypolneniyu laboratornyh rabot. – Irkutsk: IRNITU, 2018. – 146 c.
8. *Aamodt A., Plaza E.* Case-based reasoning: foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications. – 1994. – vol. 7(1). – pp. 39–59.
9. *Brambilla M., Cabot J., Wimmer M.* Model Driven Software Engineering in Practice. Morgan & Claypool Publishers, 2012.
10. *Czarnecki K., Helsen S.* Feature-based survey of model transformation approaches. IBM Systems Journal. – 2006. – vol. 45(3). – pp. 621–645.
11. *Da Silva* A.R. Model-driven engineering: A survey supported by the unified conceptual model // Computer Languages, Systems & Structures. – 2015. – vol. 43. – pp. 139–155.
12. *Dorodnykh N.O.* Model Transformations for Intelligent Systems Engineering. Open Semantic Technologies for Intelligent Systems / N.O. Dorodnykh, S.A. Korshunov, N.Yu. Pavlov, A.Yu. Yurin. – 2018. – vol. 2(8). – pp. 77–81.
13. *Dorodnykh, N.O., Yurin A.Yu.* About the specialization of model-driven approach for creation of case-based intelligence decision support systems. Open Semantic Technologies for Intelligent Systems. – 2017. – vol. 7. – pp. 151–154.
14. *Jackson, P.* (1999). Introduction to expert systems (3rd ed.). – Harlow: Addison-Wesley.

15. *Mens T., Gorp P.V.* A Taxonomy of Model Transformations // Electronic Notes in Theoretical Computer Science. – 2006. – vol. 152. – pp. 125–142.
16. *Popa A., Wood W.* Application of case-based reasoning for well fracturing planning and execution. Journal of Natural Gas Science and Engineering. – 2011. – vol.3 (6). – pp. 687–696.
17. *Yurin A.Yu., Dorodnykh N.O., Nikolaychuk O.A., Berman A.F., Pavlov A.I.* ISI models, Mendeley Data, v1 (2019). doi: http://dx.doi.org/10.17632/f9h2t766tk.1