

Программная реализация алгоритмов для создания прототипов баз знаний на основе визуального моделирования и трансформаций

Н.О. Дородных¹, А.Ю. Юрин¹✉

¹ Институт динамики систем и теории управления имени В.М. Матросова СО РАН, г. Иркутск, 664033, Россия

Ссылка для цитирования

Дородных Н.О., Юрин А.Ю. Программная реализация алгоритмов для создания прототипов баз знаний на основе визуального моделирования и трансформаций // Программные продукты и системы. 2024. Т. 37. № 3. С. 324–333. doi: 10.15827/0236-235X.142.324-333

Информация о статье

Группа специальностей ВАК: 2.3.5

Поступила в редакцию: 16.02.2024

После доработки: 07.05.2024

Принята к публикации: 14.05.2024

Аннотация. В работе описывается специализированная веб-ориентированная программная система Knowledge Modeling System (KMS), рассчитанная на экспертов и системных аналитиков. Система обеспечивает возможность построения визуальных концептуальных моделей в форме диаграмм переходов состояний, деревьев событий и отказов, а также автоматического получения кода баз знаний на основе их преобразования. Полученные таким способом базы знаний могут быть применены в качестве прототипов при дальнейшей разработке интеллектуальных систем. Созданная система основана на принципах визуального моделирования и модельных трансформаций. Последний принцип обеспечивает возможность описания соответствия между элементами различных моделей (нотаций) и их преобразования. Для формирования операторов преобразования (трансформации) концептуальных моделей описаны их модели (метамоделли), включающие основные элементы и отношения. Трансформации наглядно представлены в виде набора соответствий между элементами моделей и реализованы на языке общего назначения PHP. В качестве целевых языков определены CLIPS и OWL2 DL. Приводится описание разработанной системы, включая метод создания баз знаний на основе преобразования концептуальных моделей, основные функции, архитектуру. Реализованный метод включает этапы построения концептуальной модели предметной области, представления концептуальных моделей в формате XML, анализа XML-структуры модели, формирования модели онтологии или продукций, модификации полученных знаний, генерации кода базы знаний на целевом языке. Представлены примеры использования системы KMS в контексте создания прототипов баз знаний в области техногенной безопасности в части решения задач диагностирования и прогнозирования технического состояния объектов и систем: описания динамики развития аварии при истечении нефти из резервуара и планирования алгоритма анализа отказа.

Ключевые слова: программная система, визуальное моделирование, база знаний, прототип, продукция, онтология, дерево событий, диаграмма переходов состояний, трансформация, генерация кода

Благодарности. Работа выполнена в рамках госзадания Минобрнауки России по проекту № 121030500071-2

Введение. Интеллектуальные системы поддержки принятия решений представляют собой сложные программные комплексы, ориентированные на решение слабоформализованных прикладных задач. Они широко применяются в таких предметных областях, как медицина [1], финансы, производство, транспорт [2], энергетика, а повышение эффективности их разработки является одной из важных научно-технических проблем [3, 4].

В последнее время для представления знаний в интеллектуальных системах особую популярность получили формализмы, основанные на семантических технологиях, такие как онтологии [5] и графы знаний [6]. Остаются достаточно распространенными и востребованными языки представления знаний (ЯПЗ), обеспечивающие поддержку логических правил (продукций) [7]. При этом разработка как онтологических, так и продукционных баз зна-

ний продолжает оставаться одним из самых сложных и трудоемких этапов при создании интеллектуальных систем, поскольку требует значительных ресурсов и привлечения специалистов различной квалификации. Одним из перспективных средств решения данной проблемы являются методы автоматического извлечения знаний из различных информационных источников (например, БД, электронных документов, таблиц). Как правило, для обработки данных такие методы используют алгоритмы машинного обучения [8]. Однако в ряде случаев обрабатываемые данные не сопровождаются явной семантикой, необходимой для машинной интерпретации своего содержания, а накапливаемая в этих форматах информация часто является либо неструктурированной, либо неунифицированной (форма ее представления не соответствует какому-либо общепринятому стандарту). Это затрудняет активное

применение данных методов на практике. Также важно отметить, что большинство решений, предлагаемых в области разработки баз знаний, направлены на программистов, а вовлечение конечных пользователей (экспертов предметной области, системных аналитиков) в этот процесс весьма ограничено. Поэтому создание новых методов и программных средств, повышающих эффективность построения баз знаний интеллектуальных систем на основе автоматизированного преобразования данных из различных источников информации и направленных на непрограммирующих пользователей, является актуальной и перспективной областью научных исследований.

В данной работе описывается веб-ориентированная программная система Knowledge Modeling System (KMS) (<http://kms.knowledge-core.ru/>), реализующая алгоритмы, которые позволяют в интерактивном режиме создавать концептуальные модели, отражающие знания предметной области с помощью специализированных нотаций. Построенные таким образом визуальные модели (диаграммы) выступают в качестве основного информационного источника при автоматическом формировании кодов баз знаний на определенном ЯПЗ. При этом в качестве целевых платформ, ориентированных на описание логических правил, используются ЯПЗ CLIPS и таблицы решений в формате CSV, в случае применения онтологий – OWL в формате RDF/XML. Разработанная система была использована для прототипирования баз знаний в области техногенной безопасности при решении задач диагностирования и прогнозирования технического состояния объектов и систем [9].

Подходы к разработке баз знаний

Базы знаний могут быть созданы вручную или автоматически на основе информации из различных источников. Как правило, при ручном построении баз знаний используют различные специализированное ПО, автоматизирующее этапы получения, структурирования и представления знаний. Подходы, лежащие в основе данных программных средств, можно разделить на три основные группы.

Текстовые. Обеспечивают прямое взаимодействие пользователя с конструкциями определенного ЯПЗ. Примерами реализации этого подхода для поддержки формализма правил являются средства VISUAL JESS, ClipsWin и DroolsExpert ([\[ase/6.0.0.CR1/drools-expert-docs/html/\]\(https://docs.jboss.org/drools/rel-ase/6.0.0.CR1/drools-expert-docs/html/\)\). Для онтологических баз знаний существуют такие системы, как Protégé и FluentEditor.](https://docs.jboss.org/drools/rel-</p></div><div data-bbox=)

Табличные. Обеспечивают построение пользователем таблиц решений [10, 11] и их возможную автоматическую трансляцию в коды баз знаний.

Графические. Обеспечивают создание визуальных примитивов (например, блоков, стрелок и т.д.), соответствующих элементам баз знаний, с последующей кодогенерацией на определенном ЯПЗ. Этот подход является наиболее перспективным, так как позволяет значительно расширить круг разработчиков баз знаний за счет непрограммирующих пользователей, владеющих навыками визуального моделирования.

Как правило, в графическом подходе используют либо универсальные семантические графовые модели (например, VisiRule (http://www.lpa.co.uk/ind_hom.htm), VIPR, SCg [12]), либо специализированные графические нотации, моделирующие логические и причинно-следственные зависимости (например, URML, RVML [13]). Данное направление обладает достаточно большим количеством различных решений, однако имеет и ряд значимых ограничений. В частности, большинство подходов содержат нестандартные условные обозначения (визуальные артефакты) или вводят ограничения на именование понятий и отношений, что не всегда интуитивно понятно пользователю. Более того, некоторые решения не учитывают специфику создания баз знаний на общепринятых ЯПЗ (например, CLIPS, Jess, Drools, SWRL, OWL) и ориентируются исключительно на свой собственный формат представления знаний.

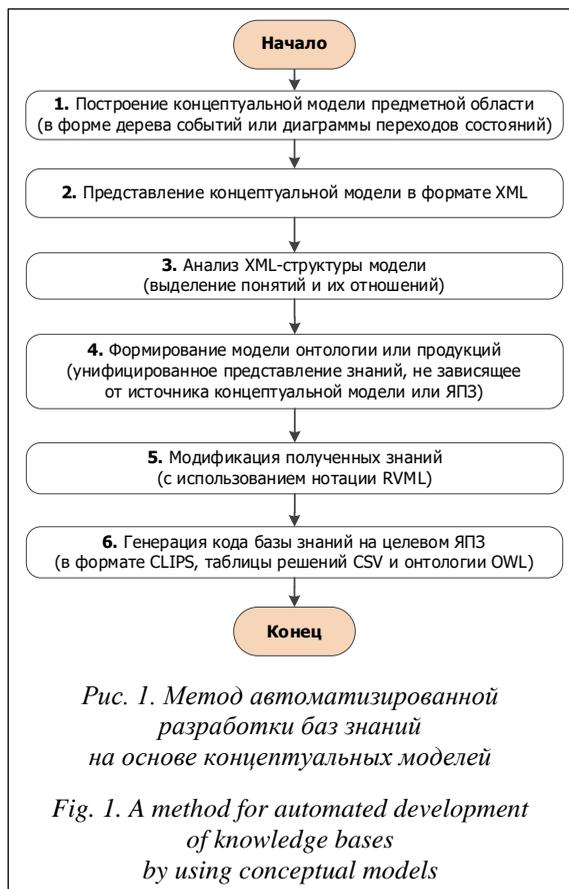
Следует отметить, что очень часто на стадиях извлечения и структурирования знаний применяют различные когнитивные или концептуальные модели: концепт-карты, деревья событий и отказов, диаграммы переходов состояний, диаграммы Исикавы и др. Данные визуальные модели являются достаточно удобным и интуитивно понятным средством представления знаний для специалистов-предметников, при этом существует ПО в виде графических редакторов, реализующих их поддержку. Однако эти редакторы не предназначены для целенаправленной разработки баз знаний и не дают какой-либо возможности для интеграции построенных визуальных моделей в базы знаний на определенном ЯПЗ. Эта особенность также затрудняет практическое использование

построенных моделей для создания баз знаний при разработке интеллектуальных систем.

Таким образом, с целью преодоления названных ограничений в данной работе предлагается специализированное программное средство Knowledge Modeling System (KMS), поддерживающее полный цикл автоматизированной разработки баз знаний, начиная от создания визуальных моделей предметной области и заканчивая кодогенерацией с возможностью модификации и проверки полученного кода.

Реализуемый метод, модели и трансформации

Разработанная KMS основана на принципах технологии PEsOT [14], которая рассматривает процесс создания интеллектуальных систем и баз знаний в контексте последовательного преобразования моделей [15, 16]. В частности, KMS реализует метод автоматизированной разработки баз знаний на основе трансформации концептуальных моделей (рис. 1).



На этапе 1 пользователь (эксперт) строит исходные концептуальные модели, описывающие некоторую предметную область. В ка-

честве исходных концептуальных моделей в KMS используются деревья событий и диаграммы переходов состояний.

Дерево событий – это графическое представление некоторой последовательности событий, которые могут произойти в результате определенного исходного события (начальной точки диаграммы) или условий (ГОСТ Р 54142-2010). Деревья событий обычно используются для анализа и оценки рисков в различных областях, таких как природная и техногенная безопасность, управление проектами, инженерия и т.д.

Модель, описывающая основные элементы данной диаграммы и их отношения, приведена на рисунке 2.

Диаграммы переходов состояний используются для моделирования поведения системы или объекта [17], позволяя визуализировать последовательность состояний и переходов между ними в виде ориентированного графа. Диаграммы переходов состояний активно используются в проектировании и анализе сложных технических и социальных систем.

Основные элементы и отношения данной диаграммы отображены на рисунке 3.

На этапе 2 построенные концептуальные модели (диаграммы) представляются (сериализуются) в формате XML с использованием специально разработанной спецификации. Формат XML удобен для представления данных диаграмм, так как позволяет описывать структуру диаграмм с помощью набора тегов и атрибутов, а также осуществлять их импорт и экспорт.

На этапах 3 и 4 выполняется анализ XML-структуры концептуальных моделей, в результате которого выделяются элементы и отношения между ними. На их основе автоматически формируется модель продукций (правил) или онтологии, при этом пользователь только выбирает необходимую целевую модель. Приведенные модели являются частью внутреннего представления знаний в системе KMS, которая не зависит от реализации конкретного ЯПЗ (например, CLIPS, Jess, Drools, SWRL, OWL и т.д.). Это дает возможность на этапе 5 визуализировать, модифицировать и проверять полученные знания в виде как правил, так и онтологии.

На этапе 6 осуществляется генерация кода баз знаний для целевой платформы путем преобразования сформированной модели продукций или онтологии.

В целом преобразование исходных деревьев событий и диаграмм переходов состояний можно описать с помощью специальных языков трансформации моделей, например TMRL [18].

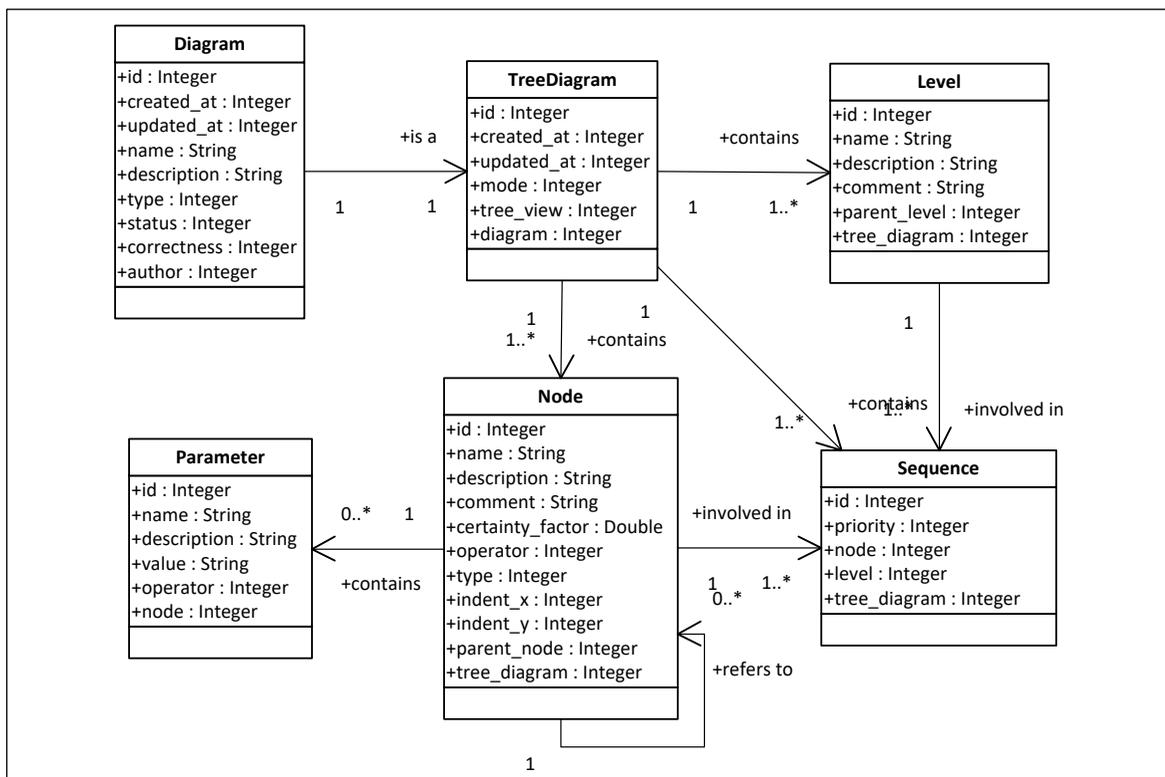


Рис. 2. Модель диаграммы деревьев событий

Fig. 2. Event tree diagram model

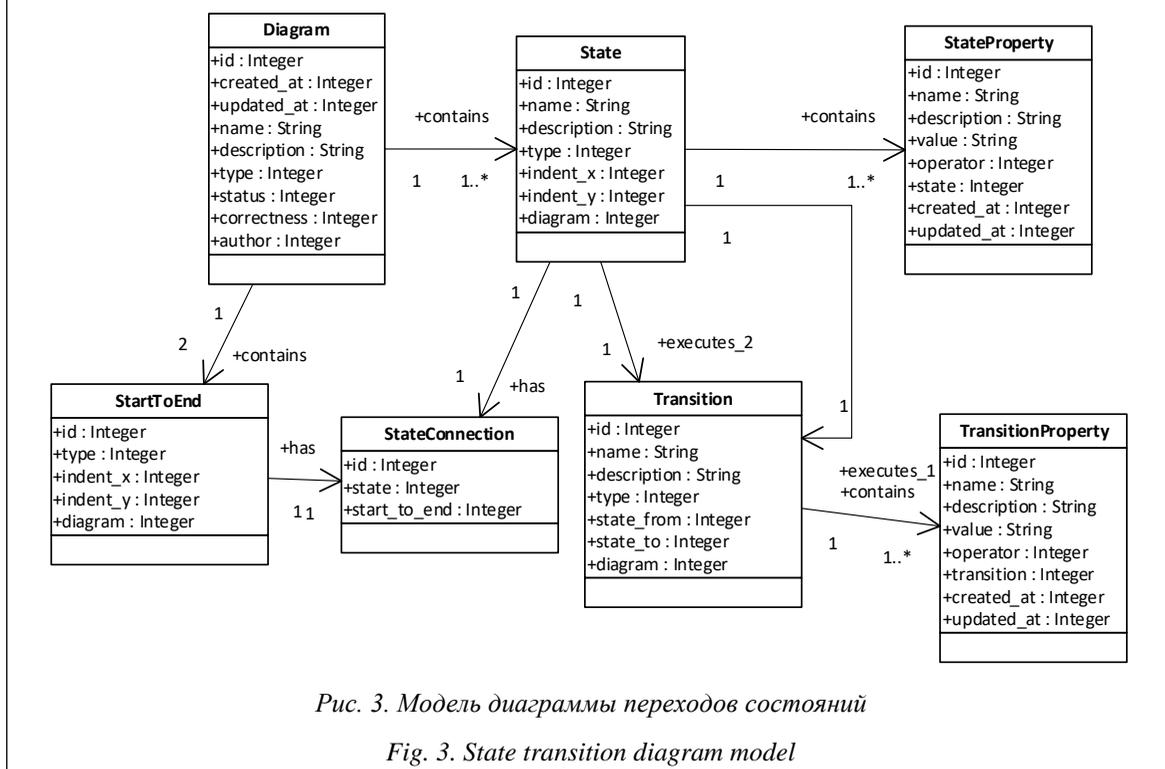


Рис. 3. Модель диаграммы переходов состояний

Fig. 3. State transition diagram model

В данной работе для реализации трансформаций использован язык программирования общего назначения PHP. При этом все трансфор-

мации можно наглядно представить в виде набора соответствий, представленных в таблицах 1, 2.

Таблица 1

Основные соответствия между элементами деревьев событий, модели продукций и продукционной базы знаний на CLIPS

Table 1

Basic correspondences between elements of event trees, a product model and a product knowledge base in the CLIPS KRL

Элемент формата представления		Элемент модели продукций	Оператор CLIPS
деревьев событий	диаграммы переходов состояний		
TreeDiagram	Diagram	Model	-
Node (name)	State (name)	FactTemplate / Fact	Deftemplate
Parameter (name)	StateProperty (name)	Slot	(slot "<name>")
Sequence	Transition (name)	RuleTemplate / Rule	defrule
Node (parent_node)	Transition (state_from)	FactTemplate / Condition	deftemplate / defrule (consequent)
Node (id)	Transition (state_to)	FactTemplate / Action	deftemplate / defrule (antecedent)

Таблица 2

Основные соответствия между элементами деревьев событий, модели онтологии и онтологической базы знаний на OWL2 DL

Table 2

Basic correspondences between elements of event trees, an ontology model and an ontology knowledge base in the OWL2 DL KRL

Формат представления		Элемент модели онтологии	OWL2 DL
деревьев событий	диаграммы переходов состояний		
TreeDiagram	Diagram	Ontology	owl:Ontology
Node	State	Class	owl:Class
Node (name)	State (name)	Class (name)	rdf:ID
Node (parent_node-id)	Transition (name)	Relationship	owl:ObjectProperty
Node (parent_node)	Transition (state_from)	Rhs	owl:ObjectProperty (rdfs:domain)
Node (id)	Transition (state_to)	Lhs	owl:ObjectProperty (rdfs:range)
Parameter	StateProperty	Property	owl:DatatypeProperty
Parameter (name)	StateProperty (name)	Property (name)	owl:DatatypeProperty (rdfs:domain)
Parameter (value)	StateProperty (value)	Property (value)	owl:DatatypeProperty (rdfs:range)
Parameter (description)	StateProperty (description)	Property (description)	rdfs:comment

Полученный таким образом код баз знаний может быть использован в сторонних програм- мных средствах, например, PKBD (<http://www.knowledge-core.ru/index.php?p=pkbd&lan=ru>) или Protégé, с целью уточнения, модификации и дальнейшего использования при разработке интеллектуальных систем.

Назначение и основные функции

Разработанная KMS предназначена для мо- делирования знаний некой предметной обла-

сти в форме концептуальных моделей и для создания на их основе прототипов баз зна- ний (<http://www.swsys.ru/uploaded/image/2024-3/10.jpg>).

Основная цель использования KMS – авто- матизировать процессы концептуализации, формализации предметных знаний, а также ко- догенерации прототипов баз знаний.

Представим основные функции KMS:

– создание, редактирование, просмотр и уда- ление пользователей;

- аутентификация и авторизация действий пользователей;
- поддержка интернационализации (английский и русский языки);
- создание, редактирование, просмотр и удаление проектов, в рамках которых ведется работа с различными диаграммами;
- создание, редактирование, просмотр и удаление диаграмм деревьев событий и диаграмм переходов состояний в рамках определенного проекта;
- проверка корректности построенных диаграмм;
- импорт и экспорт диаграмм деревьев событий и диаграмм переходов состояний в виде сериализованных файлов в формате XML;
- импорт онтологий в формате OWL (RDF/XML) в KMS, и их преобразование в диаграммы деревьев событий;
- предоставление REST API для взаимодействия с моделями продукций и онтологий с целью модификации данных моделей средствами других редакторов (RVML, KBDS);
- экспорт (генерация) построенных диаграмм в файлы форматов ЯПЗ CLIPS и OWL (RDF/XML), а также в электронные таблицы формата CSV;
- дополнительная возможность генерации спецификации виртуального ассистента на основе построенных диаграмм.

Архитектура и особенности реализации

Разработанное веб-ориентированное программное средство KMS обладает клиент-сер-

верной архитектурой (рис. 4), в которую входят следующие основные модули:

- подсистема построения концептуальных моделей, включающая два визуальных редактора: редактор классических и расширенных диаграмм деревьев событий и редактор диаграмм переходов состояний;
- модуль администрирования (авторизация/аутентификация);
- модуль управления универсальным уровнем представления знаний (предоставляет доступ к моделям продукций и онтологий);
- генераторы программных кодов и спецификаций баз знаний;
- модули импорта и экспорта диаграмм;
- модуль взаимодействия с внешними системами через REST API.

Программная реализация осуществлена с применением языков PHP 8 и JavaScript с использованием стилей Bootstrap 5. При этом визуальные редакторы реализованы с использованием библиотеки JsPlumb Toolkit. Для хранения данных применяется СУБД PostgreSQL 15. Исходный код системы открыт и доступен на GitHub (<https://github.com/Lab42-Team/kms>).

KMS поддерживает работу трех типов пользователей: *администратор* – зарегистрированный пользователь, который может производить любые действия в рамках KMS; *обычный пользователь* – зарегистрированный пользователь, который может создавать и изменять различные диаграммы в рамках определенного проекта; *гость* – незарегистрированный пользователь, который может только просматривать от-

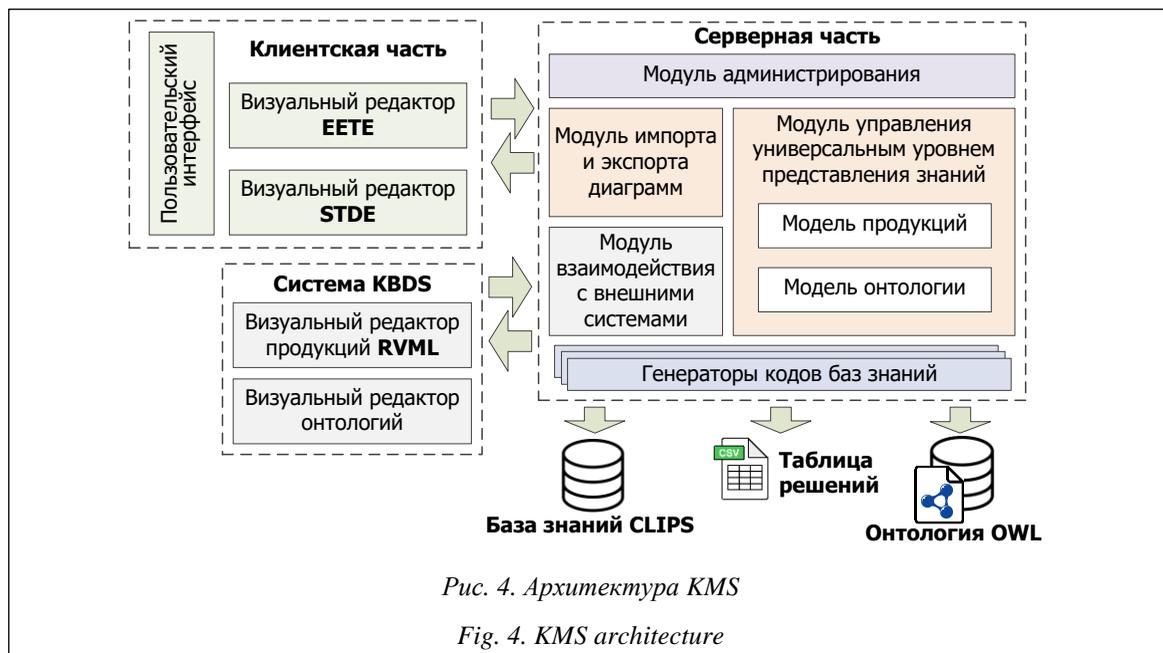


Рис. 4. Архитектура KMS

Fig. 4. KMS architecture

крытые (публичные) диаграммы и общую информацию по KMS.

Пример применения

Одним из примеров применения KMS является создание прототипов баз знаний в области техногенной безопасности при решении задач диагностирования и прогнозирования технического состояния объектов и систем [9].

Проблема оценки и повышения безопасности промышленных объектов с течением времени сохраняет свою актуальность, что обусловлено высокими темпами старения (деградации) оборудования во многих отраслях промышленности, превышающими темпы его замены и модернизации, как по субъективным, так и по объективным причинам. Значительное повышение надежности и безопасности нефтехимического оборудования можно обеспечить путем создания и активного использования методов и средств искусственного интеллекта, в частности, интеллектуальных систем поддержки принятия решений.

В рамках разработки прототипов баз знаний для ИАС «Экспертиза ПБ» были построены деревья событий, описывающие динамику развития аварий, в частности, истечение нефти (рис. 5).

В результате преобразования были получены наборы шаблонов, фактов и правил (<http://www.swsys.ru/uploaded/image/2024-3/11.jpg>).

После уточнения построенных диаграмм и тестовых прогонов (отладки) в PKBD сгенерирован программный код базы знаний на CLIPS, который затем был перенесен в ИАС «Экспертиза ПБ». Приведем фрагмент полученного кода:

```

;***** Templates *****
(deftemplate Event-0
  (slot name (default "РАЗРУШЕНИЕ СОСЕДНЕГО ОБОРУДОВАНИЯ"))
  (slot veroyatnost)
)
;***** Rules *****
(defrule 0-Event-0->Event-0+Event-0
(declare (salience 1))
(Event-0 ;Event-0
  (name "ФАКЕЛЬНОЕ ГОРЕНИЕ СТРУИ")
  (veroyatnost "0.04")
)
=>
(assert
  (Event-0 ;Event-0
    (name "РАЗРУШЕНИЕ СОСЕДНЕГО ОБОРУДОВАНИЯ")
    (veroyatnost "0.02")
  )
...

```

В рамках решения задачи информационного моделирования области анализа отказов технических систем была разработана специальная онтологическая база знаний. В частности, на рисунке 6 приведена визуальная модель в форме диаграммы переходов состояний (для ее построения использована разработанная авторами система), описывающая алгоритм анализа отказа.

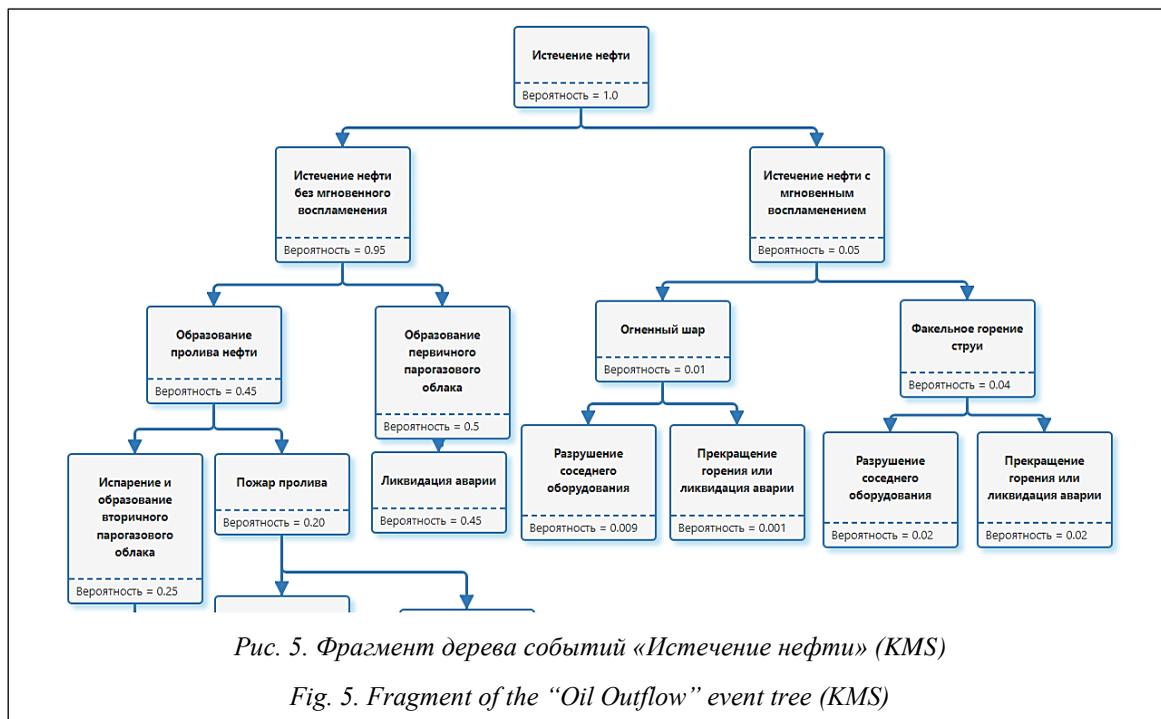


Рис. 5. Фрагмент дерева событий «Истечение нефти» (KMS)

Fig. 5. Fragment of the “Oil Outflow” event tree (KMS)

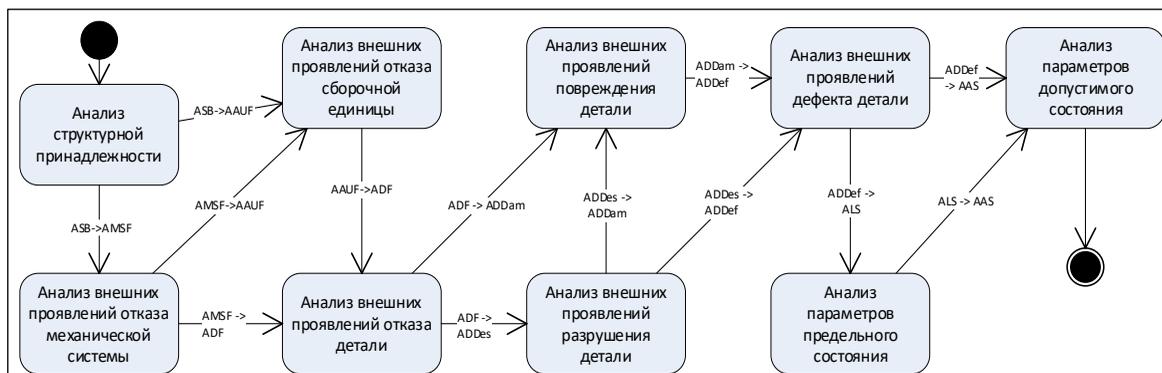


Рис. 6. Фрагмент диаграммы переходов состояний, описывающий алгоритм анализа отказов (KMS)

Fig. 6. Fragment of the "failure analysis algorithm" state transition diagram (KMS)

В результате выполнения преобразования данной диаграммы синтезирован код онтологии в формате OWL2 DL:

```

<owl:Class rdf:ID="Состояние"/>
<owl:Class rdf:ID="Переход"/>

<owl:Class
rdf:ID="АнализВнешнихПроявленийОтказаДе
тали">
  <rdfs:subClassOf rdf:re-
source="Состояние"/>
</owl:Class>

<owl:Class
rdf:ID="АнализВнешнихПроявленийПоврежде
нияДетали">
  <rdfs:subClassOf rdf:re-
source="Состояние"/>
</owl:Class>
<owl:DatatypeProperty
rdf:ID="свойстваВведены">
  <rdfs:domain rdf:re-
source="Переход"/>
  <rdfs:range rdf:re-
source="http://www.w3.org/2001/XMLSchema
#string"/>
</owl:DatatypeProperty>
...
    
```

При этом все состояния интерпретировались как классы (owl:Class), а переходы как объекты-свойства (owl:ObjectProperty). Свойства (характеристики) переходов выражались через свойства-значения (owl:DatatypeProperty), которые задаются для специального служебного класса перехода.

Заключение

Актуальность проблемы автоматизации разработки интеллектуальных систем обуслови-

вает необходимость создания нового методологического и программного инструментария, направленного на повышение эффективности процессов извлечения, структурирования и формализации знаний.

В данном контексте автоматизация создания баз знаний интеллектуальных систем остается перспективной областью научных исследований. Особенно, когда речь идет о вовлечении в данный процесс непрограммирующих пользователей, а также об использовании уже накопленной ранее информации, представленной, в частности, в форме визуальных концептуальных моделей.

В данной работе предлагается веб-ориентированное программное средство KMS, реализующее алгоритмы визуального моделирования знаний предметной области в форме диаграмм переходов состояний и деревьев событий, которые в дальнейшем являются основным информационным источником для автоматического формирования кодов баз знаний как производственного, так и онтологического типа. Данные особенности можно выделить в качестве преимуществ системы, так как они способствуют более полному вовлечению в процесс разработки непрограммирующих пользователей и минимизируют ошибки программирования.

В качестве примера использования KMS рассмотрено создание прототипов баз знаний при решении задач диагностирования и прогнозирования технического состояния объектов и систем в нефтехимии.

Список литературы

1. Saibene A., Assale M., Giltri M. Expert systems: Definitions, advantages and issues in medical field applications. Expert Systems with Applications, 2021, vol. 177, art. 114900. doi: 10.1016/j.eswa.2021.114900.

2. Wagner W.P. Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies. *Expert Systems with Applications*, 2017, vol. 76, pp. 85–96. doi: 10.1016/j.eswa.2017.01.028.
3. Сухих Н.Н., Рукавишников В.Л. Экспертные системы – средство информационной поддержки принятия решений экипажем самолета // *Изв. вузов*. 2022. № 2. С. 19–25.
4. Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И. Инженерия знаний. Модели и методы. СПб: Лань, 2023. 324 с.
5. Karabulut E., Pileggi S.F., Groth P., Degeler V. Ontologies in digital twins: A systematic literature review. *Future Generation Computer Systems*, 2024, vol. 153, pp. 442–456. doi: 10.1016/j.future.2023.12.013.
6. Ji S., Pan S., Cambria E., Martinen P., Yu P.S. A survey on knowledge graphs: Representation, acquisition and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 2022, vol. 33, no. 2, pp. 494–514. doi: 10.1109/TNNLS.2021.3070843.
7. Nowak-Brzezińska A., Wakulicz-Deja A. Exploration of rule-based knowledge bases: A knowledge engineer's support. *Information Sciences*, 2019, vol. 485, pp. 301–318. doi: 10.1016/j.ins.2019.02.019.
8. Martinez-Rodriguez J.L., Hogan A., Lopez-Arevalo I. Information extraction meets the semantic web: A survey. *Semantic Web*, 2020, vol. 11, pp. 255–335. doi: 10.3233/SW-180333.
9. Берман А.Ф., Николайчук О.А., Павлов А.И., Юрин А.Ю., Кузнецов К.А. Информационно-аналитическая поддержка экспертизы промышленной безопасности объектов химии, нефтехимии и нефтепереработки // *ХНГМ*. 2018. № 8. С. 30–36.
10. Димитров В.П., Борисова Л.В., Хубиян К.Л. Моделирование знаний в задаче поиска причин неисправностей // *Инженерные технологии и системы*. 2021. Т. 31. № 3. С. 364–379. doi: 10.15507/2658-4123.031.202103.364-379.
11. Nalepa G.J., Kluza K. UML representation for rule-based application models with XTT2-based business rules. *Int. J. of Software Eng. and Knowledge Eng.*, vol. 22, no. 4, pp. 485–524. doi: 10.1142/S021819401250012X.
12. Голенков В.В., Шункевич Д.В., Давыденко И.Т., Гракова Н.В. Принципы организации и автоматизации процесса разработки семантических компьютерных систем // *OSTIS*. 2019. № 3. С. 53–90 (на англ.).
13. Дородных Н.О., Юрин А.Ю., Коршунов С.А. Средства поддержки моделирования логических правил в нотации RVML // *Программные продукты и системы*. 2018. Т. 31. № 4. С. 667–672. doi: 10.15827/0236-235X.124.667-672.
14. Yurin A.Yu. Technology for prototyping expert systems based on transformations (PESoT): A method. *CEUR Workshop Proc. Proc. III Sci.-Pract. Workshop Inform. Tech.: Algorithms, Models, Systems*, 2020, vol. 2677, pp. 36–50.
15. Brambilla M., Cabot J., Wimmer M. *Model-driven software engineering in practice*, Springer Publ., 2017, 280 p. doi: 10.2200/s00441ed1v01y201208swe001.
16. Ogunyomi B., Rose L.M., Kolovos D.S. Incremental execution of model-to-text transformations using property access traces. *Software & System Modeling*, 2019, vol. 18, pp. 367–383. doi: 10.1007/s10270-018-0666-5.
17. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений; [пер. с англ.]. М.: Вильямс, 2008. 528 с.
18. Дородных Н.О., Юрин А.Ю. Язык для описания моделей трансформаций. *CEUR Workshop Proc. Proc. I Sci.-Pract. Workshop Inform. Tech.: Algorithms, Models, Systems*, 2018, vol. 2221, pp. 70–75 (на русс.).

Software & Systems

doi: 10.15827/0236-235X.142.324-333

2024, 37(3), pp. 324–333

Software implementation of algorithms for knowledge base prototyping with visual modelling and transformations

Nikita O. Dorodnykh¹, Aleksandr Yu. Yurin¹✉¹Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk, 664033, Russian Federation

For citation

Dorodnykh, N.O., Yurin A.Yu. (2024) 'Software implementation of algorithms for knowledge base prototyping with visual modelling and transformations', *Software & Systems*, 37(3), pp. 324–333 (in Russ.). doi: 10.15827/0236-235X.142.324-333

Article info

Received: 16.02.2024

After revision: 07.05.2024

Accepted: 14.05.2024

Abstract. The paper describes a special purpose web-oriented software system called Knowledge Modeling System (KMS) designed for experts and system analysts. The system enables building visual conceptual models in the form of state transition diagrams, event and failure trees, as well as automatic generation of knowledge base code based on their transformation. The obtained knowledge bases can become prototypes in further development of intelligent systems. The created system bases on the principles of visual modelling and model transformations. The latter principle enables describing the correspondence between the elements of different models (notations) and their transformation. Forming transformation operators for conceptual models involves describing their models (metamodels) that include basic elements and relations. The paper shows transformations as a set of correspondences between model elements, they are implemented in the general-purpose language PHP. CLIPS and OWL2 DL are the target languages. The paper describes the developed system, including the method of creating knowledge bases using the transformation of conceptual models, main functions and architecture. The implemented method includes the stages of building a subject area conceptual model, XML representation of

conceptual models, analysis of XML-structure of the model, formation of ontology model or products, modification of the obtained knowledge, generation of knowledge base code in the target language. The authors of the paper present examples of using KMS in terms of creating knowledge bases in the field of technological safety. The examples show how to solve the problems of diagnosing and forecasting the technical state of objects and systems for describing accident dynamics when oil flows out of the reservoir, as well as planning the failure analysis algorithm.

Keywords: software, visual modeling, knowledge base, prototype, rules, ontology, event tree, state transition diagram, model transformation, code generation

Acknowledgements. The work was carried out within the framework of the state assignment of the Ministry of Education and Science of the Russian Federation under the project No. 121030500071-2

References

1. Saibene, A., Assale, M., Giltri, M. (2021) 'Expert systems: Definitions, advantages and issues in medical field applications', *Expert Systems with Applications*, 177, art. 114900. doi: 10.1016/j.eswa.2021.114900.
2. Wagner, W.P. (2017) 'Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies', *Expert Systems with Applications*, 76, pp. 85–96. doi: 10.1016/j.eswa.2017.01.028.
3. Sukhikh, N.N., Rukavishnikov, V.L. (2022) 'Expert systems: Means of information support for aircraft crew decision making', *Russ. Aeronautics*, 65, pp. 243–250.
4. Gavrilova, T.A., Kudryavtsev, D.V., Muromtsev, D.I. (2023) *Knowledge Engineering. Models and Methods*. St. Petersburg, 324 p. (in Russ.).
5. Karabulut, E., Pileggi, S.F., Groth, P., Degeler, V. (2024) 'Ontologies in digital twins: A systematic literature review', *Future Generation Computer Systems*, 153, pp. 442–456. doi: 10.1016/j.future.2023.12.013.
6. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S. (2022) 'A survey on knowledge graphs: Representation, acquisition and applications', *IEEE Transactions on Neural Networks and Learning Systems*, 33(2), pp. 494–514. doi: 10.1109/TNNLS.2021.3070843.
7. Nowak-Brzezińska, A., Wakulicz-Deja, A. (2019) 'Exploration of rule-based knowledge bases: A knowledge engineer's support', *Information Sciences*, 485, pp. 301–318. doi: 10.1016/j.ins.2019.02.019.
8. Martinez-Rodriguez, J.L., Hogan, A., Lopez-Arevalo, I. (2020) 'Information extraction meets the semantic web: A survey', *Semantic Web*, 11, pp. 255–335. doi: 10.3233/SW-180333.
9. Berman, A.F., Nikolaychuk, O.A., Pavlov, A.I., Yurin, A.Yu., Kuznetsov, K.A. (2018) 'Informational and analytical support of industrial safety expert review of chemical and petrochemical objects', *Chemical and Petroleum Engineering*, (8), pp. 30–36 (in Russ.).
10. Dimitrov, V.P., Borisova, L.V., Khubiyan, K.L. (2021) 'Knowledge modeling in troubleshooting', *Engineering Technologies and Systems*, 31(3), pp. 364–379 (in Russ.). doi: 10.15507/2658-4123.031.202103.364-379.
11. Nalepa, G.J., Kluza, K. (2012) 'UML representation for rule-based application models with XTT2-based business rules', *Int. J. of Software Eng. and Knowledge Eng.*, 22(4), pp. 485–524. doi: 10.1142/S021819401250012X.
12. Golenkov, V., Shunkevich, D., Davydenko, I., Grakova, N. (2019) 'Principles of organization and automation of the semantic computer systems development', *OSTIS*, (3), pp. 53–90.
13. Dorodnykh, N.O., Yurin, A.Yu., Korshunov, S.A. (2018) 'Support tools for modeling logical rules in the RVML notation', *Software & Systems*, 31(4), pp. 667–672 (in Russ.). doi: 10.15827/0236-235X.124.667-672.
14. Yurin, A.Yu. (2020) 'Technology for prototyping expert systems based on transformations (PESoT): A method', *CEUR Workshop Proc. Proc. III Sci.-Pract. Workshop Inform. Tech.: Algorithms, Models, Systems*, 2677, pp. 36–50.
15. Brambilla, M., Cabot, J., Wimmer, M. (2017) *Model-driven Software Engineering in Practice*. Springer Publ., 280 p. doi: 10.2200/s00441ed1v01y201208swe001.
16. Ogunyomi, B., Rose, L.M., Kolovos, D.S. (2019) 'Incremental execution of model-to-text transformations using property access traces', *Software & System Modeling*, 18, pp. 367–383. doi: 10.1007/s10270-018-0666-5.
17. Hopcroft, J.E., Motwani, R., Ullman, J.D. (2001) *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publ., 521 p. (Russ. ed.: (2018) Moscow, 528 p.).
18. Dorodnykh, N.O., Yurin, A.Yu. (2018) 'A domain-specific language for transformation models', *CEUR Workshop Proc. Proc. I Sci.-Pract. Workshop Inform. Tech.: Algorithms, Models, Systems*, 2221, pp. 70–75 (in Russ.).

Авторы

Дородных Никита Олегович¹, к.т.н.,
старший научный сотрудник, nikidorny@icc.ru
Юрин Александр Юрьевич¹,
д.т.н., доцент, ведущий научный сотрудник,
зав. лабораторией, iskander@icc.ru

¹ Институт динамики систем
и теории управления
имени В.М. Матросова СО РАН,
г. Иркутск, 664033, Россия

Authors

Nikita O. Dorodnykh¹, Cand. of Sci. (Engineering),
Senior Researcher, nikidorny@icc.ru
Aleksandr Yu. Yurin¹, Dr.Sci. (Engineering),
Associate Professor, Leading Researcher,
Head of Laboratory, iskander@icc.ru

¹ Matrosov Institute for System Dynamics
and Control Theory of Siberian Branch
of Russian Academy of Sciences,
Irkutsk, 664033, Russian Federation