

УДК 004.89
DOI: 10.15827/0236-235X.132.573-581

Дата подачи статьи: 19.04.20
2020. Т. 33. № 4. С. 573–581

Применение трансформаций таблиц решений при создании интеллектуального программного модуля «Детектор» для веб-приложений

А.Ю. Юрин¹, к.т.н., доцент, зав. лабораторией, iskander@icc.ru

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

Создание модулей принятия решений для веб-приложений, использующих методы искусственного интеллекта, в частности, в форме баз знаний, требует разработки специализированного методического и программного обеспечения. Автоматизировать данный процесс можно с помощью модельно-управляемых подходов, реализующих принципы порождающего и визуального программирования, а также модельных трансформаций.

В данной работе описываются новая специализация одного из подобных подходов и ее применение для создания интеллектуального программного модуля «Детектор». Особенностью специализации является использование концептуальных моделей в форме диаграмм классов UML и таблиц решений в качестве исходных формализмов для представления знаний, авторской нотации Rule Visual Modeling Language (RVML) в качестве инструмента для визуализации логических правил, языка PHP (Hypertext Preprocessor) в качестве целевой платформы, PKBD (Personal Knowledge Base Designer) в качестве инструментария, реализующего подход.

Преимуществом подхода является возможность автоматизированного создания веб-модулей принятия решений на основе преобразований концептуальных моделей и таблиц решений без прямого программирования (манипулирования конструкциями языка программирования). Ограничения подхода связаны с определенным классом создаваемых систем (веб-модули на PHP), а также с глубиной реализуемого логического вывода: структура запрограммированных знаний такова, что решение в модулях принимается за один шаг и не предполагает цепочки рассуждений.

Приведены описание подхода и пример его применения при разработке модуля «Детектор», решающего задачу принятия решений при обнаружении нежелательных сообщений и выявлении клиентов, нарушающих правила использования сервиса СМС-информирования. Показана применимость разработанного модуля, а также дана оценка подхода по временному критерию при решении учебных (тестовых) задач.

Ключевые слова: трансформация, база знаний, правила, таблица решений, веб-приложения, UML.

При разработке модулей интеллектуальной поддержки принятия решений, в том числе для веб-приложений, наиболее трудоемким процессом (этапом) остается разработка *баз знаний* (БЗ).

Одним из способов повышения эффективности решения задач этого этапа, связанных с концептуализацией, формализацией и кодогенерацией, является использование принципов порождающего и визуального программирования, а также модельных трансформаций. Каждый из этих принципов по отдельности реализуется в различных методиках и инструментариях, например [1–3]. Однако все они имеют ограничения по поддерживаемым платформам (языкам) и требуют от разработчиков минимальных навыков программирования. Совместно рассмотренные принципы используются в рамках направлений, известных как мо-

дельно-ориентированный подход [4]: MDE (Model-Driven Engineering), MDD (Model-Driven Development), MDA (Model-Driven Architecture) и др.

В данной работе предлагается новая специализация модельно-ориентированного подхода, обеспечивающая автоматизированное создание встраиваемых модулей принятия решений для веб-приложений на основе модельных трансформаций, порождающего и визуального программирования. Создаваемые модули принятия решений содержат БЗ, реализующие логические правила. Отличием предлагаемой специализации от существующих является использование:

– концептуальных моделей в форме диаграмм классов UML и таблиц решений в качестве исходных формализмов для представления знаний в форме логических правил;

– авторского проблемно-ориентированного языка RVML (Rule Visual Modeling Language) [5] в качестве инструмента для визуализации и редактирования разрабатываемых БЗ;

– языка PHP (Hypertext Preprocessor) в качестве целевой платформы.

Подход апробирован при разработке веб-модуля принятия решений «Детектор» [6] для платформы «СМС-Органайзер». Модуль предназначен для обнаружения сообщений, нарушающих положения 38-ФЗ «О рекламе».

Вклад данной работы заключается в следующем:

– оригинальная методика автоматизированного создания веб-ориентированных модулей принятия решений, реализующих логический вывод на основе правил;

– программный модуль трансформации таблиц решений, осуществляющий предлагаемую методику и расширяющий функциональность авторской системы прототипирования продукционных БЗ PKBD (Personal Knowledge Base Designer) [5];

– веб-модуль «Детектор» [6], решающий прикладную задачу, а также результаты экспериментальной оценки подхода, демонстрируемые на учебных примерах.

Состояние вопроса

Модельно-ориентированная разработка и модельные трансформации. Принципы порождающего и визуального программирования, а также модельных трансформаций широко используются в рамках направлений, известных как модельно-ориентированная или модельно-управляемая разработка [4]. Под этим термином объединяются подходы в области программной инженерии, которые предлагают создавать ПО на основе преобразования (трансформации) и интерпретации информационных (в частном случае концептуальных) моделей. При этом наиболее стандартизированной версией MDE является архитектура, управляемая моделью (MDA), основанная на комплексном использовании стандартов OMG (Object Management Group). В соответствии с ее принципами предлагается не только использовать UML в качестве основного средства формализации и визуализации моделей, но и придерживаться определенной последовательности (цепочки) преобразований моделей: от моделей с большей абстракцией к моделям меньшей абстракции и конкретным програм-

мным кодам. При этом цепочка преобразования в классическом виде выглядит следующим образом: CIM → PIM → PSM → {software, source codes, specifications}, где CIM (Computation Independent Model) – вычислительно-независимая модель, PIM (Platform Independent Model) – платформо-независимая модель, PSM (Platform Specific Model) – платформо-зависимая модель [5].

Помимо цепочек преобразования, MDA как разновидность модельно-ориентированного подхода реализует четырехуровневую схему метамоделирования [4]. В соответствии с этой схемой выделяют следующие виды трансформаций: горизонтальные и вертикальные, эндогенные и экзогенные, однонаправленные и двунаправленные [7]. В качестве средств реализации этих трансформаций могут использоваться как специализированные языки трансформации моделей (ATL, QVT и др.), так и языки программирования общего назначения.

В данной работе предлагается специализировать и применить основные принципы MDA в контексте создания модуля принятия решений «Детектор», содержащего БЗ.

Разработка БЗ на основе модельных трансформаций. Существует ряд примеров создания БЗ и интеллектуальных систем на основе использования принципов MDD/MDA. При этом условно их можно разделить по следующим критериям [5]:

– использованные модели и цепочки их преобразований; по этому критерию можно выделить работы, использующие собственные методологические принципы без четкого выделения моделей и цепочек, реализующие цепочки преобразований и определяющие эти модели;

– целевые платформы; можно выделить работы, использующие языки общего назначения, например Perl или C#, .NET [8, 9], специализированные языки создания БЗ, например JESS, Drools, CLIPS [5, 10, 11], собственные оригинальные разработки, в частности PRISMA [9];

– источники информации; по этому критерию можно выделить работы, использующие онтологии [10–12], UML-модели [5], деревья [13] или XML-подобные структуры [8];

– ориентация на программирующего [8, 10, 11] или непрограммирующего пользователя [5, 9, 12, 14].

На основе анализа был сделан вывод о необходимости более полного вовлечения непро-

граммирующего специалиста-предметника в процесс разработки интеллектуальных систем. Подобное вовлечение может быть достигнуто за счет использования в качестве источников информации различных концептуальных моделей, а также таблиц решений и автоматизированной генерации программных кодов в результате последовательных преобразований используемых моделей.

Таким образом, в данной работе предлагается расширить специализацию MDA [5], использующую схему последовательных горизонтальных экзогенных трансформаций и реализованную при помощи языков общего назначения в форме авторского инструментария – РКВД.

Способы описания логических правил. Несмотря на популярность семантических технологий, в частности, онтологий как одного из основных формализмов представления знаний, языки представления знаний (ЯПЗ), основанные на правилах, до сих пор остаются востребованными при разработке интеллектуальных систем. Привлекательность данной модели представления знаний обусловлена ее простотой и наглядностью для предметников, высокой модульностью, легкостью внесения дополнений и изменений и прозрачностью механизма логического вывода.

При наличии множества ЯПЗ продукционного типа (CLIPS, JESS, Drools, RuleML, SWRL и др.) остается актуальной задача автоматизации создания БЗ при помощи специализированного ПО, автоматизирующего этапы получения, концептуализации и формализации предметных знаний.

На основе классификации [15] и ее последующей модификации можно выделить следующие основные группы подходов, реализуемых в ПО, решающем данную задачу:

- текстовый, обеспечивающий прямое манипулирование конструкциями ЯПЗ; подход ориентирован на программистов и реализуется в форме специализированных редакторов;
- графический, обеспечивающий создание визуальных абстракций, соответствующих элементам логических правил, с их последующей трансляцией в программные коды на ЯПЗ;
- табличный, основанный на построении таблиц решений и их трансляции в программные коды на ЯПЗ.

В данной работе в рамках новой специализации модельно-управляемого подхода предлагается применить табличный и графический

подходы совместно, при этом в рамках последнего будет использована оригинальная проблемно-ориентированная нотация RVML [5].

Предлагаемый подход

Предлагаемый подход расширяет метод, описанный в [5], и состоит из следующих основных этапов.

1. Построение моделей предметной области. Полученные модели рассматриваются как вычислительно-независимые и могут быть представлены в виде OWL-онтологии, UML-моделей (в частности, в виде UML-диаграмм классов) или карт памяти. С точки зрения классического процесса разработки БЗ этот этап соответствует этапу концептуализации.

Особенностью предлагаемого подхода является не только возможность использования на данном этапе уже ранее созданных концептуальных моделей, разработанных в таких системах, как Protégé, CmapTools, Xmind, IBM Rational Rose Enterprise и StarUML [5], но и осуществление процесса обратной инженерии канонических таблиц TabbyXL [16] с целью извлечения предметных понятий и отношений, представленных ранее в электронных таблицах различных компоновки и стилей.

2. Построение платформо-независимых моделей. Модели являются результатом трансформации вычислительно-независимой модели с возможностью их отображения в RVML для визуализации и последующего уточнения. Алгоритмы трансформации реализованы в РКВД [5]. С точки зрения классического процесса разработки БЗ этот этап соответствует этапу формализации.

Особенностью предлагаемого подхода является совместное использование графических концептуальных моделей и формализма таблиц решений. При этом для формирования таблиц решений может использоваться любой табличный редактор, в частности Microsoft Excel.

3. Построение платформо-зависимых моделей. Количество этих моделей определяется количеством платформ, для которых создается интеллектуальная система, они являются результатом автоматических преобразований платформо-независимой модели с помощью специальных инструментов с последующей модификацией конечным пользователем. В соответствии с подходом разработчик должен уточнить RVML модели правил с учетом особенностей целевой платформы.

Особенностью предлагаемой методологии является расширение списка платформ, поддерживаемых инструментарием за счет языка общего назначения РНР. В частности, для обеспечения кодогенерации на данном этапе уточняются элементы RVML-схем, такие как приоритеты правил и значения слотов по умолчанию.

4. Генерация программных кодов. На этом этапе автоматически с помощью РКВД выполняется интерпретация RVML-схем с получением программных кодов. В рамках данной работы кодогенерация расширена поддержкой языка РНР.

5. Тестирование. Полученные программные коды тестируются как в специальном ПО (в интерпретаторе РКВД), так и в готовом приложении после их интеграции.

Интеграция является единственным действием, требующим навыков программирования при использовании предлагаемого подхода, все остальные этапы могут осуществляться непрограммирующими пользователями.

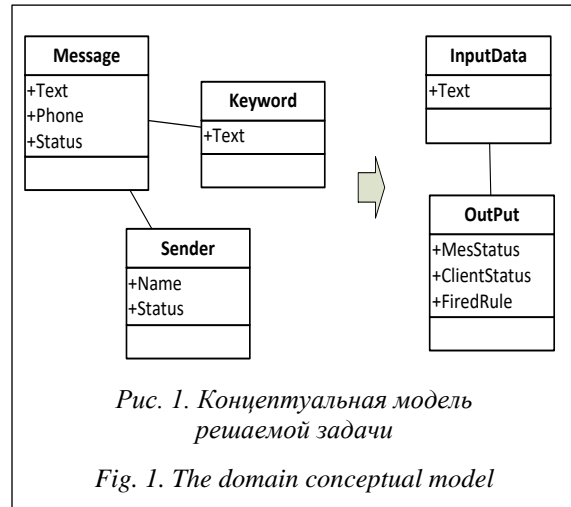
С точки зрения модельных трансформаций цепочка преобразований новой специализации модельно-ориентированного подхода имеет следующий вид: UML → DecisionTables → RVML → РНР.

Разработка веб-модуля «Детектор»

Рассмотрим применение предлагаемого подхода при разработке программного модуля «Детектор», содержащего продукционную БЗ. Модуль предназначен для обнаружения сообщений, нарушающих положения 38-ФЗ «О рекламе», а также отправляющих их абонентов.

1. На этапе построения модели предметной области был выполнен ее анализ и выделены

ключевые абстракции (рис. 1). На концептуальном уровне основными элементами в решаемой задаче являются СМС-сообщение (message), содержащее ключевые слова (keywords), так называемые маркеры СПАМ-сообщений, и отправитель сообщений (sender). В дальнейшем при формализации модель была упрощена для более полного соответствия структуре логических правил.



Для выделения комбинаций маркеров (ключевых слов) был произведен анализ БД из 1 366 490 сообщений и отобраны 829 сообщений клиентов, уличенных ранее в рассылке СПАМ-сообщений.

Сообщения сформировали обучающую выборку (тестовый набор), в которой были выделены пять основных групп (табл. 1): пропаганда запрещенных веществ, недобросовестная реклама, мошенничество, угрозы и оскорбления, сообщения, не несущие признаков нежелательных. По каждой группе определена возможная рекомендация модуля поддержки при-

Таблица 1

Основные группы нежелательных сообщений

Table 1

Main groups of banned messages

№ группы	Наименование группы	Всего	Отобрано	Реакция системы	
				Новый статус сообщения	Новый статус пользователя (индекс)
1	Пропаганда запрещенных веществ	346	253	Ошибка	Заблокирован (1)
2	Недобросовестная реклама	12	8	На модерации	Активен (0)
3	Мошенничество	236	80	Ошибка	Заблокирован (1)
4	Угрозы и оскорбления	185	146	На модерации	Заблокирован (1)
5	Сообщения, не несущие признаков нежелательных	50	0	-	-
Итого:		829	487		

нения решения, в частности, блокировка или нет отправки сообщения (изменения его статуса) и блокировка или нет отправителя.

2. Выделенные концепты были использованы при построении платформо-независимых моделей, в том числе в форме таблицы решений, содержащей информацию о 487 уникальных наборах (табл. 2). Таблица была разработана путем автоматизированного анализа БД сообщений, структура таблицы сформирована путем перечисления свойств всех концептов, где каждое свойство – это столбец таблицы.

3. Построение платформо-зависимых моделей осуществлялось с использованием PKBD путем импорта разработанной таблицы решений (см. <http://www.swsys.ru/uploaded/image/2020-4/2020-4-dop/15.jpg>) с последующим уточнением в виде RVML (см. <http://www.swsys.ru/uploaded/image/2020-4/2020-4-dop/16.jpg>), включая приоритет правил и значения по умолчанию.

4. Посредством генератора PKBD синтезирована 6 871 строчка PHP-кода с описанием 487 правил. При этом каждое логическое правило было представлено в форме условного оператора if, в условии которого анализируется вхождение определенных ключевых слов в отправляемое сообщение. Приведем фрагмент полученного кода:

```
<?php
//***** exported from PKBD
*****
// version: 4.2018.0201.6
// knowledge base:
// info:
//***** classes
*****
class InputData_1{
    var $Keyword;
    function Init(){
        $this->Keyword = "";
    }
}
class OutPut_1{
    var $MesStatus;
    var $ClientStatus;
    var $FiredRule;
    function Init(){
        $this->MesStatus = "";
        $this->ClientStatus = "";
        $this->FiredRule = "";
    }
}
//***** Initialization (facts)
*****
$InputData_1_ = new InputData_1;
$InputData_1_->Init();
$OutPut_1_ = new OutPut_1;
$OutPut_1_->Init();
//***** rules
*****
//rule_1-2
if (
    ((strpos($InputData_1_->Keyword,
        "1п+1п=2п") != false))
){
    $OutPut_1_->MesStatus = "ошибка";
    $OutPut_1_->ClientStatus = "1";
    $OutPut_1_->FiredRule = "rule_1-2";
}
...

```

Таблица 2

Фрагмент таблицы решений

Table 2

A fragment of a decision table

RuleName	InputData:: Keyword	InputData:: Keyword	InputData:: Keyword	#OutPut:: MesStatus	#OutPut:: ClientStatus	#OutPut:: FiredRule
rule 1-10	pharmbar.com	pharmbar.net		ошибка	1	rule 1-10
rule 1-100	левые	железного	зеленого	ошибка	1	rule 1-100
rule 1-101	видно	внизу	даже	ошибка	1	rule 1-101
rule 1-102	киви	лежит	мегафон	ошибка	1	rule 1-102
rule 1-103	валлет	сумма	виза	ошибка	1	rule 1-103
rule 1-104	давай	киви	мегафононский	ошибка	1	rule 1-104
rule 1-105	готовый	киви	клад	ошибка	1	rule 1-105
rule 1-106	изза	меня	проблеммы	ошибка	1	rule 1-106
rule 1-107	дублируются	сверить	клады	ошибка	1	rule 1-107
rule 1-108	городом	клады	остались	ошибка	1	rule 1-108
rule 1-109	березой	кичигино	остановка	ошибка	1	rule 1-109
rule 1-111	клады	курительные	легал	ошибка	1	rule 1-111
rule 1-110	легал	наличии		ошибка	1	rule 1-110
rule 1-111	меня	пока	тихо	ошибка	1	rule 1-111

Приоритеты правил были учтены путем сортировки атомарных условных операторов в рамках вычислительного блока (функции), таким образом, в случае активации правила происходили остановка вывода и выход из блока, при этом возвращался результат последнего активированного правила.

В дальнейшем сгенерированный код был интегрирован в платформу «СМС-Органайзер».

5. Тестирование разработанного модуля «Детектор» осуществлялось с использованием выделенной ранее базы сообщений. В частности, после модификации БЗ модуля составляла 498 правил, которые обеспечили обнаружение 653 нежелательных сообщений из тестового набора, таким образом, точность (ассурасу) составила 0,83. Среднее время выполнения (быстродействие) – 0,00026 сек. В результате проверки подсистемой «Детектор» набора из 1 366 490 сообщений были выявлены 1 145 СПАМ-сообщений и 25 клиентов (отправителей), нарушивших положения 38-ФЗ «О рекламе», которые ранее не были обнаружены.

Оценка подхода

Оценка эффективности подхода осуществлялась по временному критерию, при этом использовались полученные ранее результаты и тестовые задачи [5]. Основная цель выполнения тестовых задач – создание БЗ экспертных систем статического типа и определение затрат времени на выполнение отдельных этапов разработки.

Задачи решались тремя способами.

Способ 1 (PKBD) основан на использовании PKBD в части импорта концептуальных моделей и автоматизированного ввода элементов БЗ, при этом решались задачи импорта модели, ввода правил, автоматической кодогенерации.

Способ 2 (Decision Tables) базируется на использовании Microsoft Excel для создания таблиц решений и их последующей обработки в PKBD; решались задачи ручного создания таблиц решений, импорта таблиц, корректировки правил, автоматической кодогенерации.

Способ 3 (Hand coding) основан на ручном программировании (кодировании) элементов БЗ в специализированном редакторе.

Результаты показали, что использование таблиц решений позволяет значительно сократить затраты времени по сравнению с ручным кодированием, однако в среднем на 36 % менее эффективно по сравнению с использованием

специальных программ-мастеров PKBD или манипулирования специализированными графическими примитивами (например RVML) (рис. 2).

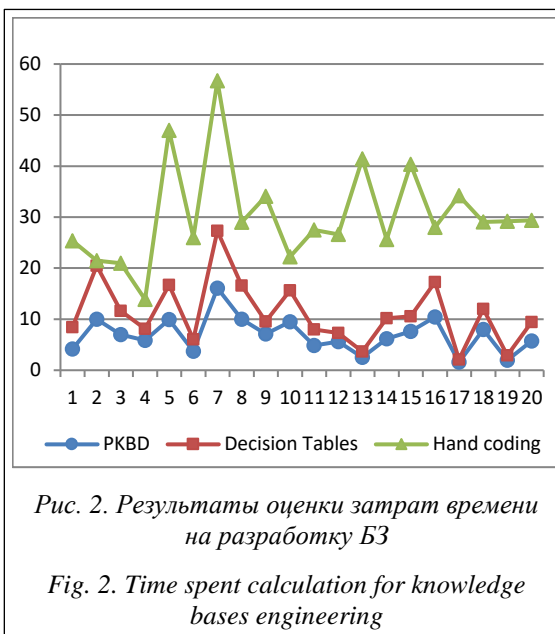


Рис. 2. Результаты оценки затрат времени на разработку БЗ

Fig. 2. Time spent calculation for knowledge bases engineering

Необходимо отметить, что в данном эксперименте таблицы решений создавались вручную и объем тестовых БЗ не превышал 10–12 правил. Однако при автоматизированном формировании таблиц решений, содержащих большее количество записей (487, как в рассмотренном примере), ручной ввод правил посредством программ-мастеров PKBD будет не так эффективен.

Заключение

Использование таблиц решений при разработке БЗ является хорошо зарекомендовавшей себя практикой [17], особенно при большом количестве правил. Применение же модельных трансформаций и специализированного инструментария позволяет снизить трудоемкость кодификации правил при использовании таблиц решений.

В данной работе описана новая специализация подхода к разработке БЗ продукционного типа, основанная на принципах модельных трансформаций. Особенности подхода являются:

- представление исходных данных в форме таблиц решений;
- использование авторского проблемно-ориентированного языка описания логических правил RVML [5] как инструмента для визуа-

лизации и редактирования разрабатываемой БЗ;

– использование РНР как целевого языка (платформы) для генерации программных кодов.

Среди ограничений подхода можно выделить следующие: класс создаваемых систем – веб-модули принятия решений; платформа – РНР; глубина логического вывода создаваемых веб-модулей – решение в модулях принимается

за один шаг и не предполагает цепочки рассуждений.

Преодоление данных ограничений может быть рассмотрено как направление дальнейших исследований, а также как расширение области применения за счет описанного в [18] решения задач генерации экспертных систем и БЗ для решения междисциплинарных задач обеспечения техногенной безопасности на основе самоорганизации.

Литература

1. Akkermans H., Schreiber G., Ahjewierden A., de Hoog R., Shadbolt N.R., de Velde W.V., Wielinga B. Knowledge Engineering and Management: The CommonKADS Methodology. UK, Cambridge, MIT Press Publ., 2000, 447 p.
2. Stokes M. Managing Engineering Knowledge: МОКА: Methodology for Knowledge Based Engineering Applications. NY, ASME Press Publ., 2001, 298 p.
3. Рыбина Г.В. Инструментальные средства построения динамических интегрированных экспертных систем: развитие комплекса АТ-ТЕХНОЛОГИЯ // Искусственный интеллект и принятие решений. 2010. № 1. С. 41–48.
4. De Silva A.R. Model-driven engineering: a survey supported by the unified conceptual model. Computer Languages, Systems & Structures, 2015, no. 43, pp. 139–155. DOI: 10.1016/j.cl.2015.06.001.
5. Грищенко М.А., Дородных Н.О., Николайчук О.А., Юрин А.Ю. Применение модельно-управляемого подхода для создания производственных экспертных систем и баз знаний // Искусственный интеллект и принятие решений. 2016. № 2. С. 16–29.
6. Юрин А.Ю. Детектор: Свид. регистр. ПрЭВМ. № 2020614257. Рос. Федерация, 2020.
7. Mens T., Gorp P.V. A taxonomy of model transformations. Electronic Notes in Theoretical Computer Science, 2006, no. 152, pp. 125–142. DOI: 10.1016/j.entcs.2005.10.021.
8. Dunstan N. Generating domain-specific web-based expert systems. Expert Systems with Applications, 2008, vol. 35, pp. 686–690. DOI: 10.1016/j.eswa.2007.07.048.
9. Cabello M.E., Ramos I., Gomez A., Limon R. Baseline-oriented modeling: An MDA approach based on software product lines for the expert systems development. Proc. 1st Asian Conf. Intelligent Information and Database Systems, 2009, pp. 208–213.
10. Shue L., Chen C., Shiue W. The development of an ontology-based expert system for corporate financial rating. Expert Systems with Applications, 2009, no. 36, pp. 2130–2142. DOI: 10.1016/j.eswa.2007.12.044.
11. Canadas J., Palma J., Tunes S. InSCo-Gen: A MDD tool for web rule-based applications. ICWE, 2009, no. 5648, pp. 523–526. DOI: 10.1007/978-3-642-02818-2_53.
12. Nofal M.A., Fouad K.M. Developing web-based semantic and fuzzy expert systems using proposed tool. IJCA, 2015, no. 112, pp. 38–45.
13. Ruiz-Mezcua B., Garcia-Crespo A., Lopez-Cuadrado J., Gonzalez-Carrasco I. An expert system development tool for non AI experts. Expert Systems with Applications, 2011, no. 38, pp. 597–609. DOI: 10.1016/j.eswa.2010.07.009.
14. Kadhim M.A., Alam M.A., Kaur H. Design and implementation of intelligent agent and diagnosis domain tool for rule-based expert system. Proc. ICMIRA, 2013, pp. 619–622. DOI: 10.1109/ICMIRA.2013.129.
15. Гаврилова Т.А., Гулякина Н.А. Визуальные методы работы со знаниями: попытка обзора // Искусственный интеллект и принятие решений. 2008. № 1. С. 15–21.
16. Shigarov A.O., Mikhailov A.A. Rule-based spreadsheet data transformation from arbitrary to relational tables. Information Systems, 2017, no. 71, pp. 123–136. DOI: 10.1016/j.is.2017.08.004.
17. Еремеев А.П. Развитие аппарата таблиц решений в контексте создания гибридных интеллектуальных систем // Гибридные и синергетические интеллектуальные системы: матер. III Всерос. Поспеловской конф. с междунар. участием. 2016. С. 121–132.
18. Берман А.Ф., Николайчук О.А., Юрин А.Ю., Павлов А.И. Принципы информационной технологии решения междисциплинарных задач обеспечения техногенной безопасности на основе самоорганизации // Информационные и математические технологии в науке и управлении. 2019. № 2. С. 5–15.

Using decision tables transformations when creating the "Detector" intelligent software module for web applications

A.Yu. Yurin¹, Ph.D. (Engineering), Associate Professor, Head of Laboratory, iskander@icc.ru

¹Matrosov Institute of System Dynamics and Control Theory SB RAS, Irkutsk, 664033, Russian Federation

Abstract. Creating decision-making modules for web applications (which are including knowledge bases) requires the development of specialized methods and tools. In this connection, the use of model-driven approaches that implement the principles of transformations, generative and visual programming is promising.

This paper describes a new specialization of one of these approaches and its application for creating the "Detector" intelligent software module. This specialization involves the use of decision tables and conceptual models in the form of UML class diagrams for knowledge formalization and representation; a domain-specific language, namely, Rule Visual Modeling Language for designing logical rules; the Hypertext Preprocessor (PHP) language as a target software platform; Personal Knowledge Base Designer as software that implements the approach.

The advantage of this approach is the automated creation of web-based decision-making modules based on transformations of conceptual models and decision tables without direct programming (direct manipulation of programming language constructs). The limitations of the approach are related to a certain class of created systems (PHP web modules), as well as the depth of the implemented logical inference: the decision in the modules is made in one step and does not involve a chain of reasoning.

The description of the proposed approach and an example of its application for developing the "Detector" module are presented. The "Detector" is intended for decision making when detecting banned messages and clients violating rules of using a Short Message Service. The applicability of the developed module is shown, as well as the evaluation of the approach based on a time criterion for solving educational (test) problems.

Keywords: transformations, knowledgebase, rules, decision table, web applications, UML.

References

1. Akkermans H., Schreiber G., Ahjewierden A., de Hoog R., Shadbolt N.R., de Velde W.V., Wielinga B. *Knowledge Engineering and Management: The CommonKADS Methodology*. UK, Cambridge, MIT Press Publ., 2000, 447 p.
2. Stokes M. *Managing Engineering Knowledge: MOKA: Methodology for Knowledge Based Engineering Applications*. NY, ASME Press Publ., 2001, 298 p.
3. Rybina G.V. Tools for constructing dynamic integrated expert systems: the AT-TECHNOLOGY complex development. *Artificial Intelligence and Decision Making*, 2010, no. 1, pp. 41–48 (in Russ.).
4. De Silva A.R. Model-driven engineering: a survey supported by the unified conceptual model. *Computer Languages, Systems & Structures*, 2015, no. 43, pp. 139–155. DOI: 10.1016/j.cl.2015.06.001.
5. Grishchenko M.A., Dorodnykh N.O., Nikolaychuk O.A., Yurin A.Yu. Application of a model-driven approach to create production expert systems and knowledge bases. *Artificial Intelligence and Decision Making*, 2016, no. 2, pp. 16–29 (in Russ.).
6. Yurin A.Yu. *Detector*. Patent RF, no. 2020614257, 2020.
7. Mens T., Gorp P.V. A taxonomy of model transformations. *Electronic Notes in Theoretical Computer Science*, 2006, no. 152, pp. 125–142. DOI: 10.1016/j.entcs.2005.10.021.
8. Dunstan N. Generating domain-specific web-based expert systems. *Expert Systems with Applications*, 2008, vol. 35, pp. 686–690. DOI: 10.1016/j.eswa.2007.07.048.
9. Cabello M.E., Ramos I., Gomez A., Limon R. Baseline-oriented modeling: An MDA approach based on software product lines for the expert systems development. *Proc. 1st Asian Conf. Intelligent Information and Database Systems*, 2009, pp. 208–213.
10. Shue L., Chen C., Shiue W. The development of an ontology-based expert system for corporate financial rating. *Expert Systems with Applications*, 2009, no. 36, pp. 2130–2142. DOI: 10.1016/j.eswa.2007.12.044.
11. Canadas J., Palma J., Tunes S. InSCo-Gen: A MDD tool for web rule-based applications. *ICWE*, 2009, no. 5648, pp. 523–526. DOI: 10.1007/978-3-642-02818-2_53.
12. Nofal M.A., Fouad K.M. Developing web-based semantic and fuzzy expert systems using proposed tool. *IJCA*, 2015, no. 112, pp. 38–45.

13. Ruiz-Mezcua B., Garcia-Crespo A., Lopez-Cuadrado J., Gonzalez-Carrasco I. An expert system development tool for non AI experts. *Expert Systems with Applications*, 2011, no. 38, pp. 597–609. DOI: 10.1016/j.eswa.2010.07.009.
14. Kadhim M.A., Alam M.A., Kaur H. Design and implementation of intelligent agent and diagnosis domain tool for rule-based expert system. *Proc. ICMIRA*, 2013, pp. 619–622. DOI: 10.1109/ICMIRA.2013.129.
15. Gavrilova T.A., Gulyakina N.A. Visual knowledge processing techniques: A brief review. *Artificial Intelligence and Decision Making*, 2008, no. 1, pp. 15–21 (in Russ.).
16. Shigarov A.O., Mikhailov A.A. Rule-based spreadsheet data transformation from arbitrary to relational tables. *Information Systems*, 2017, no. 71, pp. 123–136. DOI: 10.1016/j.is.2017.08.004.
17. Ereemeev A.P. The development of decision tables apparatus in the context of creating hybrid intelligent systems. *Proc. 3rd All-Russ. Pospelov's Conf. on Hybrid and Synergistic Intelligent Systems*, 2016, pp. 121–132 (in Russ.).
18. Berman A.F., Nikolaychuk O.A., Yurin A.Yu., Pavlov A.I. Principles of information technology for solving interdisciplinary problems of technogenic safety provision on the basis of self-organization. *Information and Mathematical Technologies in Science and Management*, 2019, no. 2, pp. 5–12 (in Russ.).

Для цитирования

Юрин А.Ю. Применение трансформаций таблиц решений при создании интеллектуального программного модуля «Детектор» для веб-приложений // Программные продукты и системы. 2020. Т. 33. № 4. С. 573–581. DOI: 10.15827/0236-235X.132.573-581.

For citation

Yurin A.Yu. Using decision tables transformations when creating the "Detector" intelligent software module for web applications. *Software & Systems*, 2020, vol. 33, no. 4, pp. 573–581 (in Russ.). DOI: 10.15827/0236-235X.132.573-581.