

УДК 004.89

СОЗДАНИЕ НЕЧЕТКИХ БАЗ ЗНАНИЙ НА ОСНОВЕ ПРЕОБРАЗОВАНИЯ  
НЕЧЕТКИХ КОНЦЕПТУАЛЬНЫХ МОДЕЛЕЙ

Дородных Никита Олегович\*, Николайчук Ольга Анатольевна\*\*,  
Юрин Александр Юрьевич\*\*\*

\* к.т.н., н.с., e-mail: [tualatin32@mail.ru](mailto:tualatin32@mail.ru),

\*\* д.т.н., в.н.с., e-mail: [nikoly@icc.ru](mailto:nikoly@icc.ru),

\*\*\* к.т.н., зав. лабораторией «Информационных технологий исследования  
природной и техногенной безопасности», e-mail: [iskander@icc.ru](mailto:iskander@icc.ru),

Институт динамики систем и теории управления

им. В.М. Матросова СО РАН,

664033, г. Иркутск, ул. Лермонтова, 134,

Коршунов Сергей Андреевич

Программист «ООО ЦентраСиб»,

e-mail: [info@centrasib.ru](mailto:info@centrasib.ru),

664003, г. Иркутск, ул. Лермонтова, 277/5, 26.

**Аннотация.** Разработка баз знаний остается важной областью научных исследований. Эффективность этого процесса может быть повышена за счет автоматизированного анализа существующих моделей предметной области в виде концептуальных диаграмм различных типов. В данной работе предлагается подход, который может быть использован для прототипирования нечетких продукционных баз знаний путем преобразования концептуальных моделей с нечеткими факторами. Предлагаемый подход включает в себя: расширенный предметно-ориентированный декларативный язык описания моделей трансформаций (TMRL); методику автоматизированного анализа и преобразования нечетких концептуальных моделей, реализованных в XML-подобных форматах; программные средства в форме модулей-конвертеров для системы разработки баз знаний (KBDS), поддерживающие предложенный подход. Апробация подхода осуществлена при создании баз знаний для решения задач автоматизации проведения экспертизы промышленной безопасности.

**Ключевые слова:** получение знаний, нечеткость, нечеткая база знаний, трансформация моделей, генерация кода, нечеткая концептуальная модель, нечеткие деревья событий.

**Цитирование:** Дородных Н. О., Николайчук О. А., Юрин А. Ю., Коршунов С. А. Создание нечетких баз знаний на основе преобразования нечетких концептуальных моделей. // Информационные и математические технологии в науке и управлении. 2020. № 2 (18). С. 19 – 35. DOI: 10.38028/ESI.2020.18.2.002

**Введение.** Одним из способов повышения эффективности создания программного обеспечения, в частности, интеллектуальных систем и баз знаний (БЗ), является использование принципов модельно-ориентированного подхода (Model-Driven Engineering)

[16] и трансформаций концептуальных моделей, представленных в различных форматах (например, UML, IDEF1X и др.). Под концептуальными понимаются модели, представленные множеством понятий и связей между ними, которые в совокупности определяют смысловую структуру рассматриваемой предметной области вместе со свойствами и характеристиками, классификацией этих понятий по типам, ситуациям, признакам в данной области и законов протекания процессов в ней [1].

Однако, при решении реальных практических задач используемая информация часто является неточной или неопределенной, например, в случае больших объемов данных о состоянии уникальных механических и технических систем или результатах каких-либо измерений. При этом классические концептуальные модели часто не позволяют представлять и интерпретировать подобную неточность. Чтобы удовлетворить потребность в моделировании сложных объектов и процессов с учетом факторов неточности и неопределенности активно развивается область нечеткого концептуального моделирования, в частности, теория нечетких логики и множеств [15, 19, 26]. Данная теория широко применяется для расширения существующих концептуальных моделей, в части учета факторов неопределенности вербальных характеристик, неточности, разной степени уверенности, отсутствия некоторых данных и т.п. Результатом ее применения стало появление новых форматов нечетких концептуальных моделей, являющихся модификацией популярных подходов и языков концептуального моделирования, например: нечетких когнитивных карт (fuzzy cognitive maps) [24], нечетких моделей «сущность-связь» (fuzzy ER models) [33], нечетких UML моделей [29], а также нечетких древовидных семантических структур в виде нечетких деревьев событий (fuzzy event trees) или отказов (fuzzy fault trees), применяемых в области нечеткого анализа отказов и риска технических систем (fuzzy fault tree analysis и fuzzy event tree analysis) [28, 30] и др.

Данная работа является продолжением исследований авторов на тему автоматизации разработки БЗ интеллектуальных систем на основе трансформации концептуальных моделей. Ранее авторами был предложен подход [32] к автоматизированному созданию продукционных БЗ на основе анализа и преобразования различных концептуальных моделей, в частности: диаграмм классов UML [7], концепт-карт ХТМ [8], деревьев событий [4], диаграмм Исикавы [5] и др. Данные модели используются для комплексного исследования сценариев развития нежелательных (деградационных) процессов, включающих сложные взаимодействия между техническими системами и их компонентами, персоналом, внешними и внутренними воздействующими факторами.

В данной статье предлагается обобщить полученный опыт по данной тематике и применить его к обработке концептуальных моделей, содержащих нечеткую информацию. Таким образом, далее представлен подход, направленный на разработку нечетких БЗ продукционного типа путем анализа и трансформации нечетких концептуальных моделей. В частности, рассматривается:

- формальное описание процесса трансформации нечетких концептуальных моделей в код нечетких БЗ;
- расширение проблемно-ориентированного языка – Transformation Model Representation Language (TMRL) [18], обеспечивающее поддержку описания «нефакторов», таких как: неточность, неопределенность, нечеткость и др. [11];

- методика автоматизированного создания нечетких БЗ на основе трансформации нечетких концептуальных моделей с использованием TMRL.

Апробация представленного подхода осуществлена на примере трансформации нечетких диаграмм деревьев событий в код нечеткой БЗ для решения задачи прогнозирования развития деградационных процессов на нефтехимическом оборудовании, в частности, на трубчатом реакторе.

## 1. Состояние вопроса в области трансформации моделей

**1.1. Основные положения.** В общем случае трансформация моделей представляет собой процесс автоматической генерации целевой модели по исходной модели в соответствии с некоторым набором правил преобразования [17]. По существу, трансформация моделей является логическим продолжением (развитием) области преобразования программ. В специальной литературе по трансформации моделей рассматривается широкий спектр артефактов в качестве потенциальных преобразуемых объектов для процесса разработки программного обеспечения. К ним относятся различные UML-модели, характеристические модели (feature models), спецификации интерфейсов, схемы данных, компонентов и программный код. При этом сами трансформации рассматриваются с различных точек зрения [25]: по типу результата, по типу используемого языка реализации, по уровню абстракции моделей и направлению трансформаций.

Одним из основных понятий трансформации моделей является метамодель, представляющая собой модель языка, используемого для описания трансформаций. В целом для представления метамodelей и трансформаций используется четырехуровневая иерархия метамоделирования (M0-M3 схема) [16]. Более детальное описание данной схемы трансформации, применительно к инженерии знаний приводятся в [3].

На сегодняшний день существует ряд примеров подходов к автоматизированному созданию БЗ и интеллектуальных систем, основанных на принципах модельно-ориентированного подхода (Model-Driven Engineering) [16] и трансформации моделей [10, 14, 32]. Следует отметить, что эти подходы предназначены, главным образом, для четко определенных задач и не позволяют обрабатывать нечеткую и неопределенную информацию. Таким образом, обработка такой информации требует новых специальных методов и программного обеспечения.

**1.2. Средства реализации модельных трансформаций.** В настоящее время методы преобразования моделей базируются на графовых грамматиках (переписывании графов), теории категорий и гибридном (декларативно-императивном) подходе. Наиболее популярными языками преобразования моделей являются: QVT (Query/View/Transformation) [27] – спецификация консорциума OMG, которая определяет три языка преобразования моделей: QVTc, QVTo и QVTr; ATL (ATLAS Transformation Language) [23] – язык описания преобразований моделей, основанный на стандарте QVT, и стандартизированный язык описания ограничений, называемый OCL (Object Constraint Language); GReAT (Graph REwriting And Transformation) – язык описания модельных преобразований, основанный на подходе тройных преобразований графов [13]; Henshin – язык преобразования моделей, основанный на переписывании графов и использовании шаблонных правил, которые могут быть структурированы во вложенные единицы преобразования с четко определенной семантикой [12]; Epsilon – семейство языков и инструментов для преобразования моделей,

генерации кода, проверки моделей, миграции и рефакторинга [20]; XSLT (eXtensible Stylesheet Language Transformations) – язык преобразования XML-документов [31].

Основным недостатком этих языков преобразования моделей являются высокие требования к квалификации конечных пользователей при разработке правил преобразования. В частности, пользователю необходимо знать синтаксис и семантику определенного языка преобразования моделей, который может быть довольно сложным. Пользователь также должен знать языки метамоделирования (например, MOF, Ecore, KM3 и т. д.), которые используются для описания входных и выходных моделей, а также различные расширения и дополнения (например, OCL). Существенным недостатком почти всех языков преобразования моделей является жесткая привязка к определенным инструментам, в частности, к платформе Eclipse. Сочетание этих факторов усложняет практическое использование этих языков и инструментов в разработке БЗ и интеллектуальных систем, особенно когда преобразование концептуальных моделей предметной области производится непрограммирующими пользователями (например, экспертами предметной области, инженерами по знаниям, аналитиками и т. д.).

Таким образом, в данной статье предлагается подход к разработке нечетких БЗ на основе трансформации концептуальных моделей, содержащих нечеткие факторы. Подход ориентирован на непрограммирующих пользователей и позволяет с помощью специализированных методических, языковых и программных средств повысить эффективность создания нечетких БЗ продукционного типа.

## **2. Подход к разработке нечетких баз знаний на основе трансформации нечетких концептуальных моделей**

**2.1. Формальное описание модельных трансформаций.** Предлагаемый подход позволяет автоматизировать этапы формализации и кодификации знаний за счет анализа нечетких концептуальных моделей. В свою очередь анализируемые модели – это результат этапов получения и концептуализации экспертных знаний об объекте или процессе.

Формализовать постановку задачи можно следующим образом: необходимо определить оператор преобразования нечеткой концептуальной модели:

$$T : FCM \rightarrow FKB, \quad (1)$$

где  $FCM$  – исходная нечеткая концептуальная модель;  $FKB$  – целевая БЗ на определенном языке представления нечетких знаний.

В зависимости от языка моделирования и формата представления, нечеткая концептуальная модель может обладать различной внутренней структурой, иметь различное количество элементов и отношений между ними. Однако, каждая подобная нечеткая концептуальная модель, расширенная аппаратом *нечеткой логики* (fuzzy logic) [15], на абстрактном уровне (abstract syntax) обладает единообразным определением факторов *нечеткости* и *неопределенности*.

Фактор нечеткости концептуальной модели включает определение понятия – *лингвистической (нечеткой) переменной* (fuzzy variable) для учета нечеткости в форме отдельных конструкций модели (например, нечетких классов, нечетких свойств и т.п.). Значения данной лингвистической переменной определяются через так называемые *нечеткие множества* (fuzzy sets) [11, 19], а именно: имеется базовая шкала  $X$ , каждое значение на которой определяет некоторое числовое значение, принимаемое определяемой

нечеткой переменной; функция принадлежности  $\mu(x)$ , принимающая значения на интервале  $[0; 1]$  и отражающая субъективную степень уверенности одного эксперта в том, что данное конкретное значение базовой шкалы соответствует определяемому нечеткому множеству и принимающую значение на интервале. Таким образом, нечеткое множество  $A$  – это совокупность пар вида  $x; \mu(x)$ , т.е.:  $A = \{(x_1; \mu(x_1)), \dots, (x_n; \mu(x_n))\}$ . Данный вариант задания функции принадлежности для определения лингвистической переменной называется *табличным*. Другим вариантом задания функции принадлежности для определения лингвистической переменной является аналитический способ, для которого могут быть описаны различные виды функций принадлежности (например, треугольная, трапецевидная,  $S$ -образная и  $Z$ -образная сплайн-функция, линейная  $S$ -образная и  $Z$ -образная функция,  $\Pi$ -образная функция и др.).

Таким образом, на уровне абстрактного синтаксиса исходную нечеткую концептуальную модель можно разделить на две части:

$$FCM = \{FV, E, R\}, \quad (2)$$

где  $FV$  – множество нечетких сущностей концептуальной модели, определенных как лингвистические переменные с использованием нечетких множеств;  $E$  – множество конкретных четких (crisp) сущностей концептуальной модели (классов, объектов, свойств, отношений и т.п.);  $R$  – множество отношений между  $FV$  и  $E$ .

Вторым базовым понятием нечетких концептуальных моделей является фактор *неопределенности*, который возникает, когда нет полной уверенности в некоторой части информации, представленной в модели (например, у пользователя нет полной уверенности в существовании определенных связей между элементами и т.д.). Степень неопределенности, как правило, представляется *коэффициентом уверенности* (certainty factor) принимающее числовое значение по шкале от 0 до 1. Данный коэффициент может относиться к любым элементам концептуальной модели, при этом если он равен 1, то считается, что данный элемент определен пользователем как верный (т.е. у пользователя есть полная уверенность в истинности данного элемента), и наоборот – если коэффициент уверенности равен 0, то данный элемент определен неверно (т.е. у пользователя есть полная уверенность в ложности данного элемента).

Используя (2) формализуем фактор неопределенности для нечеткой концептуальной модели на уровне абстрактного синтаксиса:

$$E = \{E^{cf}, E^c\},$$

где  $E^{cf}$  – множество сущностей концептуальной модели с определенными коэффициентами уверенности;  $E^c$  – множество четких сущностей концептуальной модели без коэффициентов уверенности.

Таким образом,  $FV$  и  $E^{cf}$  являются основными элементами в исходной концептуальной модели, которые задают нечеткость информации представленной в данной модели.

Уточним оператор трансформации, определенный в (1), для представления полной последовательности преобразования нечетких концептуальных моделей:

$$T = \langle T_{FCM-FRM}, T_{FRM-FKB} \rangle,$$

$$T_{FCM-FRM} : FCM^{XML} \rightarrow FRM, T_{FRM-FKB} : FRM \rightarrow FKB^{FuzzyCLIPS},$$

где  $T_{FCM-FRM}$  – оператор преобразования исходной нечеткой концептуальной модели в модель нечетких продукций;  $T_{FRM-FKB}$  – оператор преобразования модели нечетких продукций в код нечеткой БЗ;  $FCM^{XML}$  – исходная нечеткая концептуальная модель, представленная (сериализованная) в XML-подобном формате;  $FRM$  – модель нечетких продукций, являющиеся унифицированным средством представления извлекаемых элементов нечеткой концептуальной модели, поддерживающая описание для  $FV$  и  $E^{cf}$ ;  $FKB^{FuzzyCLIPS}$  – код нечеткой БЗ на языке представления нечетких знаний – FuzzyCLIPS [22]. FuzzyCLIPS выбран как один из наиболее широко используемых языков и инструментальных сред для разработки нечетких продукционных экспертных систем.

Для решения поставленной задачи и обеспечения поддержки всех этапов трансформации, определенных в (2), предлагается использовать ранее разработанную технологию [2] и ее реализацию в виде веб-ориентированной программной системы (Knowledge Base Development System, KBDS) [9], а также разработанные метамоделей  $FRM$  и  $FKB^{FuzzyCLIPS}$  и метод создания метамоделей для  $FCM^{XML}$  в качестве методологической основы для поддержки синтеза нечетких БЗ [6]. Технология обеспечивает поддержку создания программных компонентов (модулей-конвертеров), предназначенных для трансформации концептуальных моделей, представленных в XML-подобных форматах, в код БЗ на целевом языке представления знаний. Для описания трансформаций в рамках данной технологии используется проблемно-ориентированный язык TMRL, предназначенный для описания и обработки четких концептуальных моделей.

Далее определим расширение TMRL, а также приведем специальную методику для поддержки синтеза нечетких БЗ на основе рассмотренной технологии и с учетом этапов трансформации, определенных в (2).

**2.2. Расширение декларативного языка описания моделей трансформаций.** Для описания сценариев трансформации моделей, содержащих элементы нечеткости, предлагается расширить разработанный ранее язык описания моделей трансформаций – Transformation Model Representation Language (TMRL) [18]. Грамматика TMRL относится к классу контекстно-свободных грамматик (КС-грамматик, в частности – LL (1)). TMRL описывает элементы моделей в декларативной форме, в частности, в виде правил, представляющих соответствия элементов метамоделей. Спецификации TMRL отвечают требованиям точности, ясности и полноты: представлена вся необходимая информация для рассматриваемых преобразований; все объекты моделей хорошо формализованы; спецификации достаточно компактны и понятны (читабельны).

Сценарии трансформации на TMRL состоят из трех основных блоков: элементы и взаимосвязи исходной метамоделей; элементы и взаимосвязи целевой метамоделей; правила преобразования, описывающие соответствие между элементами исходной и целевой метамоделей. В создаваемых правилах могут использоваться логические операторы («and», «or») и оператор условного выбора («if»), в частности, с целью задания определенного значения (константы) для целевого элемента метамоделей в соответствии с выполнением условия, определенного в блоке «if» (например, исходя из определенного значения

исходного элемента). Основные элементы TMRL описаны в Таблице 1. Подробное описание TMRL представлено в [18].

Таблица 1. Основные элементы TMRL

Элемент	Описание
Source Meta-Model	Описание исходной метамодели для языка моделирования (исходные концептуальные модели представлены на этом языке).
Target Meta-Model	Описание целевой метамодели для языка представления знаний (на этом языке представлена целевая нечеткая модель на основе правил).
Elements	Описание понятий исходной и целевой метамodelей.
Attributes	Описание концептуальных свойств исходной и целевой метамodelей.
Relationships	Описание взаимосвязей исходной и целевой метамodelей.
Transformation	Основная конструкция, описывающая набор правил преобразования.
Rule	Описание правила преобразования, которое определяет соответствие между понятиями или свойствами исходной и целевой метамodelей.
Call	Дополнительный блок для вызова существующего модуля преобразования, который обрабатывает эту модель.

В данной работе мы расширяем TMRL, добавляя возможность представления и обработки нечетких факторов  $FV$  и  $E^{cf}$ . В частности, добавлены следующие термы (элементы): лингвистическая (нечеткая) переменная, функция принадлежности (как в табличном, так и в аналитическом виде), нечеткое множество. Таким образом, нечеткое множество в TMRL описывается следующим образом:

```
<fuzzy set> = <description type> [<membership function type> (<membership function parameters>)]
<description type> = table | analytical
<membership function type> = triangular | trapezoidal | ...
<membership function parameters> = (a, b, c) | (a, b, c, d) | ...
```

**2.3. Методика автоматизированной разработки нечетких базы знаний** представляет собой систематизированную совокупность действий, которые нацелены на решение задачи автоматического создания кода нечеткой БЗ на FuzzyCLIPS путем трансформации исходных нечетких концептуальных моделей с использованием программных компонентов в виде модулей-конвертеров и возможностью промежуточного хранения и редактирования (уточнения) полученных нечетких знаний.

Основным принципом созданной методики является возможность использования двухступенчатой трансформации ( $T_{FCM-FRM}$  и  $T_{FRM-FKB}$ ), что повышает технологичность процесса разработки нечетких БЗ: сначала преобразование нечеткой концептуальной модели в модель нечетких продукций с возможностью модификации и уточнения полученных нечетких знаний, затем преобразование этой модели в код нечеткой БЗ на целевом ЯПЗ FuzzyCLIPS.

В общем виде автоматизированная разработка нечетких БЗ может быть представлена в виде следующей последовательности этапов (Рис. 1):

Этап 1: Анализ XML-структуры нечеткой концептуальной модели, в ходе которого выделяются понятия предметной области и их отношения, в том числе конструкции,

описывающие  $FV$  и  $E^{cf}$ . Анализ происходит на основе разработанной модели трансформации на языке TMRL.

Этап 2: Формирование модели нечетких продукций, как универсального представления нечетких знаний, независящего от исходной концептуальной модели или языка представления знаний. Подробное описание данной модели представлено в [6]. Следует отметить, что на данном этапе возможна модификация полученных нечетких продукций и добавление недостающих правил, которые не удалось извлечь на предыдущем этапе.

Этап 3: Генерация кода нечеткой БЗ в формате FuzzyCLIPS на основе модели нечетких продукций. Мета модель языка FuzzyCLIPS с описанием основных конструкций приводится в работах [22].



Рис. 1. Методика автоматизированной разработки нечетких БЗ на основе трансформации нечетких концептуальных моделей

Модули-конвертеры для осуществления двухступенчатой трансформации ( $T_{FCM-FRM}$  и  $T_{FRM-FKB}$ ) могут быть сформированы на основе технологии автоматической сборки программных компонентов [2, 3]. Ядром таких модулей является сценарий трансформации, определяющий набор правил преобразования на расширенном TMRL. Несмотря на простоту синтаксиса TMRL, прямое манипулирование конструкциями языка предполагается только в особых случаях. В обычной ситуации используется методика автоматизированного создания моделей трансформации и реализующий ее инструментарий, который является частью инструментальной системы KBDS, в частности:

- анализатор (XML-парсер) исходного формата нечеткой концептуальной модели;
- визуальный редактор метамodelей;
- визуальный редактор моделей трансформаций;
- генератор кода моделей трансформаций на TMRL.



С использованием этого инструментария процесс создания моделей трансформации на TMRL состоит из следующих этапов:

- создание метамодели для исходной нечеткой концептуальной модели на основе XML-схемы ее описания или путем прямого анализа структуры исходной нечеткой концептуальной модели (процедура обратной инженерии);
- выбор метамодели целевой БЗ (по умолчанию реализована метамодель нечетких продукций и FuzzyCLIPS, представленные в [6]);
- разработка модели трансформации, включая: анализ структуры исходной и целевой метамодели; построение правил трансформации в визуальном редакторе в виде соответствий между элементами исходной и целевой метамодели; генерация кода модели трансформации на TMRL; модификация полученного кода TMRL при необходимости.

На основе полученного TMRL-сценария автоматически может быть сгенерирован программный код модуля-конвертера. При этом используется концепция шаблонной или каркасной сборки, когда с разъемы (слоты) наполняются определенным содержимым, полученным на основе интерпретации модели трансформации. На текущий момент код модулей генерируется на PHP и требует доработки программистом для их дальнейшей интеграции в систему KBDS. Подробнее об этом в [2].

**5. Пример применения.** В качестве примера применения предлагаемого подхода рассмотрена задача трансформации концептуальных моделей в форме деревьев событий, содержащих нечеткие факторы, в код нечеткой БЗ продукционного типа. Исходные диаграммы деревьев событий в формате EETD (Extended Event Tree Diagram) были получены в процессе концептуального моделирования в прототипе веб-ориентированного программного средства Extended ETE [21]. Данные диаграммы описывают динамику развития деградационных процессов (ДП) на рассматриваемой механической системе, в частности – рассматривается коррозионное растрескивание на стадии повреждения на уникальном трубчатом реакторе типа «труба в трубе», являющийся компонентом сложного технологического комплекса «Производство полиэтилена высокого давления» и выполняющего функцию обеспечения теплообмена между рабочей средой высокого давления и контактирующей теплообменной средой в межтрубном пространстве (в рубашке) (Рис. 2). Детально данное четкое дерево событий рассмотрено в работе [4].

В приведенном примере (Рис. 2) на диаграмме дерева событий указано два нечетких события, нечеткость которых обусловлена значениями характеризующих их параметров (параметры выделены пунктиром): «глубина микротрещин» и «длина микротрещин», а также каждому событию соответствует коэффициент уверенности, задающий вероятность происхождения того или иного события (коэффициент уверенности указан в нижней части блока события). Механизмы динамики ДП могут быть также описаны в виде нечетких диаграмм Исикавы [6]. Остальные элементы дерева являются четкими. Представление детальной информации о нечетких событиях в примере имеет вид, указанный на Рисунках 3 и 4.



**Рис. 2.** Пример фрагмента нечеткой диаграммы дерева события для ДП «Коррозионное растрескивание» на стадии повреждения (треугольник – символ механизма процесса; М-КР; М-СУБ-1,2; М-МИК-1,2 – механизмы процессов коррозионного растрескивания на общем, субмикро- и микроуровнях соответственно).

Эксперта 1	Глубина : значительная
[1,0; 4,0] мм	
Треугольная (3,0; 3,5; 4,0)	

**Рис. 3.** Нечеткий фактор «глубина микротрещины».

Эксперта 1	Длина : значительная
[1,0; 5,0] мм	
Трапецевидная (3,0; 3,5; 4,5; 5,0)	

**Рис. 4.** Нечеткий фактор «длина микротрещины».

Разработка БЗ производилась согласно рассмотренной выше методике. На основе извлеченных элементов нечеткой диаграммы дерева событий сгенерирована модель нечетких продукций, как совокупность шаблонов правил (включая нечеткие). На основе полученной модели синтезирован код нечеткой БЗ на FuzzyCLIPS. Фрагмент сгенерированного кода нечеткой продукционной БЗ:

```
(deftemplate glubina
  1,0 4,0 mm
  (
    (znachitelnaia (3,0 0) (3,5 0.5) (4,5 1) )
  )
)
(deftemplate dlina
  1,0 5,0 mm
  (
    (znachitelnaia (3,0 0) (3,5 1) (4,5 1) (5,0 0) )
  )
)
...
(deffacts fuzzy-facts
  (glubina znachitelnaia)
  (dlina znachitelnaia)
)
```

**Заключение.** Эффективность разработки БЗ, в том числе с элементами нечеткости, может быть повышена за счет автоматизированного анализа существующих моделей предметной области в виде концептуальных диаграмм различных типов. В данной работе предлагается подход в виде совокупности методических, языковых и программных средства, которые могут быть использованы для преобразования концептуальных моделей в БЗ с учетом нечетких факторов. Дается формальное описание процесса трансформации нечетких концептуальных моделей с рассмотрением нечетких факторов, общих для различных типов концептуальных моделей. Рассмотренный подход включает: расширение предметно-ориентированного языка описания моделей трансформаций (TMRL); методику и программные средства, реализующие автоматизированный анализа и преобразования исходных нечетких концептуальных моделей, а также построение модулей-конверторов.

Прямое корректное сравнение с другими подходами, основанными на трансформации, затруднено из-за различий в их назначении и реализациях. В связи с этим возможно только качественное сопоставление некоторых отдельных их элементов, например, трансформационных языков. Сравнение показало, что основное различие между TMRL и существующими языками трансформации моделей (например, QVT, ATL и т.д.) заключается в его простоте использования, которая достигается за счет ограниченного набора элементов. TMRL не является расширением других языков и не использует конструкции других языков, как это очень часто делают другие языки преобразования (в частности, ATL использует язык ограничений OCL). Кроме того, TMRL имеет удобочитаемый синтаксис, а дополнительной особенностью TMRL является интеграция с ранее разработанными программными компонентами KBDS.

Предложенный подход реализован в виде двух экспериментальных программных компонентов (модулей-конвертеров), которые могут быть использованы в качестве автономных модулей в составе других интеллектуальных систем. Подход был апробирован при прототипировании БЗ, используемой при решении задач автоматизации проведения экспертизы промышленной безопасности, в частности, для решения задачи диагностирования состояний технических устройств, используемых на нефтехимических

производствах. В рамках решения данной задачи в настоящее время разрабатывается экспертная система идентификации ДП. На этапе создания нечеткой БЗ экспертам был предложен изложенный подход. Были получены показатели его эффективности, характеризующие повышение качества и сокращение временных затрат.

В дальнейшем мы планируем усовершенствовать разработанный инструментарий и использовать его для создания БЗ для решения междисциплинарных задач обеспечения технологической безопасности на основе самоорганизации.

Работа выполнена при финансовой поддержке РФФИ (проект № 19-07-00927).

#### СПИСОК ЛИТЕРАТУРЫ

1. Аверкин А.Н., Гаазе-Рапопорт М.Г., Поспелов Д.А. Толковый словарь по искусственному интеллекту. М.: Радио и связь, 1992. 256 с.
2. Бычков И.В., Дородных Н.О., Юрин А.Ю. Подход к разработке программных компонентов для формирования баз знаний на основе концептуальных моделей // Вычислительные технологии. 2016. Т. 21, №4. С. 16–36.
3. Дородных Н.О., Николайчук О.А., Юрин А.Ю., Коршунов С.А. Разработка и использование метамodelей для синтеза спецификаций и кодов баз знаний // Информационные и математические технологии в науке и управлении. 2019. №2(14). С. 26–39.
4. Дородных Н.О., Николайчук О.А., Юрин А.Ю. Автоматизированное создание продукционных баз знаний на основе деревьев событий // Информационные и математические технологии в науке и управлении. 2017. №2(6). С. 30–41.
5. Дородных Н.О., Николайчук О.А., Юрин А.Ю. Подход автоматизированной разработки баз знаний на основе трансформации диаграмм Исикавы // Вестник компьютерных и информационных технологий. 2018. №4. С. 41–51.
6. Дородных Н.О., Николайчук О.А., Юрин А.Ю. Разработка метамodelей для поддержки синтеза нечетких баз знаний // Вестник компьютерных и информационных технологий. 2020. №1. С. 34–47.
7. Дородных Н.О., Юрин А.Ю. Использование диаграмм классов UML для формирования продукционных баз знаний // Программная инженерия. 2015. №4. С. 3–9.
8. Дородных Н.О., Юрин А.Ю. Использование концепт-карт для автоматизированного создания продукционных баз знаний // Программные продукты и системы. 2017. №4. С. 658–662.
9. Дородных Н.О. Web-based software for automating development of knowledge bases on the basis of transformation of conceptual models // Открытые семантические технологии проектирования интеллектуальных систем. 2017. №7. С. 145–150.
10. Стенников В.А., Барахтенко Е.А., Соколов Д.В. Применение онтологий при реализации концепции модельно-управляемой разработки программного обеспечения для проектирования теплоснабжающих систем // Онтология проектирования. 2014. Т. 14, № 4. С. 54–68.

11. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 166 с.
12. Arendt T., Biermann E., Jurack S., Krause C., Taentzer G. Henshin: advanced concepts and tools for in-place EMF model transformations // Model Driven Engineering Languages and Systems. MODELS 2010. Lecture Notes in Computer Science. Vol. 6394. 2010. Pp. 121–135.
13. Balasubramanian D., Narayanan A., Buskirk C., Karsai G. The graph rewriting and transformation language: GReAT // Electronic Communications of the EASST. Vol. 1. 2006. Pp. 1–8.
14. Baumeister J., Striffler A. Knowledge-driven systems for episodic decision support // Knowledge-Based Systems. 2015. Vol. 88. Pp. 45–56.
15. Biacino L., Gerla G. Fuzzy logic, continuity and effectiveness // Archive for Mathematical Logic. Vol. 41, № 7. 2002. Pp. 643–667.
16. Cretu L.G., Florin D. Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice. Apple Academic Press. 2014. 350 p.
17. Czarnecki K., Helsen S. Feature-based survey of model transformation approaches // IBM Systems Journal. Vol. 45, № 3. 2006. Pp. 621–645.
18. Dorodnykh N.O., Yurin A.Y. A domain-specific language for transformation models // CEUR Workshop Proceedings. Proceedings for First Scientific-practical Workshop Information Technologies: Algorithms, Models, Systems. Vol. 2221. 2018. Pp. 70–75.
19. Dubois D., Prade H., Yager R.R. Readings in Fuzzy Sets for Intelligent Systems. 1st ed. // Elsevier. 2014.
20. Epsilon. 2017. URL: <http://www.eclipse.org/epsilon/> (дата обращения: 2020-06-20).
21. Extended Event Tree Editor (EETE). 2020. Доступ: [https://github.com/LedZeppelin/event\\_tree\\_editor](https://github.com/LedZeppelin/event_tree_editor) (дата обращения: 2020-06-20).
22. Orchard R.A. FuzzyCLIPS Version 6.04A – User’s Guide. Institute for Information Technology, National Research Council Canada. 1995.
23. Jouault F., Allilaire F., Bézivin J., Kurtev I. ATL: A model transformation tool // Science of Computer Programming. Vol. 72, № 1. 2008. Pp. 31–39.
24. Kosko B. Fuzzy Cognitive Maps // International Journal of Man-Machine Studies. Vol. 24. 1986. Pp. 65–75.
25. Mens T., Gorp P.V. A Taxonomy of Model Transformations // Electronic Notes in Theoretical Computer Science. Vol. 152. 2006. Pp. 125–142.
26. Novak V. On fuzzy type theory // Fuzzy Sets and Systems. Vol. 149, № 2. 2005. Pp. 235–273.
27. Query/View/Transformation (QVT) Version 1.3 // OMG Document formal/2016-06-03. 2016. URL: <http://www.omg.org/spec/QVT/1.3/> (дата обращения: 2020-06-20).
28. Senol Y.E., Aydogdu Y.V., Sahin B., Kilic I. Fault Tree Analysis of chemical cargo contamination by using fuzzy approach // Expert Systems with Applications. Vol. 42, № 12. 2015. Pp. 5232–5244.

29. Sicilia M.A., Garcia E., Gutierrez J.A. Integrating fuzziness in object oriented modeling language: towards a fuzzy-UML // Proceedings of International Conference on Fuzzy Sets Theory and its Applications. 2002. Pp. 66–67.
  30. Wang D., Liu E., Huang L. Event Tree Analysis of Xinda Oil Pipeline Leakage Based on Fuzzy Set Theory // The Open Civil Engineering Journal. Vol. 11. 2017. Pp. 101–108.
  31. XSL Transformations (XSLT) Version 2.0. 2007. Доступ: <http://www.w3.org/TR/xslt20> (дата обращения: 2020-06-20).
  32. Yurin A.Yu., Dorodnykh N.O., Nikolaychuk O.A., Grishenko M.A. Designing rule-based expert systems with the aid of the model-driven development approach // Expert Systems. Vol. 35, № 5. 2018. Pp. 1–23.
  33. Zhang F., Ma Z. M., Yan L. Representation and reasoning of fuzzy ER model with description logic // Proceedings of IEEE International Conference on Fuzzy Systems. 2008. Pp. 1358–1365.
- 

**UDK 004.89**

## **FUZZY KNOWLEDGE BASE ENGINEERING BASED ON TRANSFORMATION OF FUZZY CONCEPTUAL MODELS**

**Nikita O. Dorodnykh\***, **Olga A. Nikolaychuk\*\***, **Alexander Yu. Yurin\*\*\***

\* Ph.D., e-mail: [tualatin32@mail.ru](mailto:tualatin32@mail.ru)

\*\* D.Sc, e-mail: [grey.for@gmail.com](mailto:grey.for@gmail.com)

\*\*\* Ph.D., Head. Laboratory of Information and telecommunication technologies for investigation of technogenic safety, e-mail: [iskander@icc.ru](mailto:iskander@icc.ru)

Matrosov Institute for System Dynamics and Control Theory of Siberian Branch  
of Russian Academy of Sciences (ISDCT SB RAS)  
134, Lermontov Str., 664033, Irkutsk, Russia,

**Sergey A. Korshunov**

Programmer, CentrSib LLC, 11, e-mail: [info@centrasib.ru](mailto:info@centrasib.ru)  
Karl Marx Str., 664003, Irkutsk, Russia,

**Abstract.** Knowledge base engineering remains an important area of scientific research. The efficiency of this process can be enhanced by an automated analysis of existing domain models in the form of conceptual diagrams of various types. In this paper, we propose an approach that can be used to prototype fuzzy rule-based knowledge bases by transforming conceptual models with fuzzy factors. The proposed approach includes: an extended domain-specific declarative language for describing transformation models (TMRL); a technique for automated analysis and transformation of fuzzy conceptual models in XML-like formats; tools in the form of converters for the Knowledge Base Development System (KBDS) supporting the proposed approach. The approach was tested when creating knowledge bases for solving problems of automation of the industrial safety inspection.

**Keywords:** knowledge acquisition, fuzzy, fuzzy knowledge base, model transformation, code generation, fuzzy conceptual model, fuzzy event trees

### References

1. Averkin A.N., Haase-Rapoport M.G., Pospelov D.A. Explanatory Dictionary of Artificial Intelligence. Moscow: Radio and communications. 1992. (in Russian).
2. Bychkov I.V., Dorodnykh N.O., Yurin A.Yu. Podhod k razrabotke programmnykh komponentov dlja formirovaniya baz znaniy na osnove konceptual'nykh modelej [Approach to the development of software components for generation of knowledge bases based on conceptual models]. Vychislitel'nye tehnologii = Computational Technologies. Vol. 21, № 4. 2016. P. 16–36. (in Russian).
3. Dorodnykh N.O., Nikolaychuk O.A., Yurin A.Yu. Avtomatizirovannoe sozdanie produkcionnykh baz znaniy na osnove derev'ev sobytij [Automated creation of rule-based knowledge bases on the basis of event trees]. Informacionnye i matematicheskie tehnologii v nauke i upravlenii = Information and mathematical technologies in science and management. Vol. 2, № 14. 2019. P. 26–39. (in Russian).
4. Dorodnykh N.O., Nikolaychuk O.A., Yurin A.Yu. Avtomatizirovannoe sozdanie produkcionnykh baz znaniy na osnove derev'ev sobytij [Automated creation of rule-based knowledge bases on the basis of event trees]. Informacionnye i matematicheskie tehnologii v nauke i upravlenii = Information and mathematical technologies in science and management. Vol. 2, № 6. 2017. Pp. 30–41. (in Russian).
5. Dorodnykh N.O., Nikolaychuk O.A., Yurin A.Yu. Podhod avtomatizirovannoj razrabotki baz znaniy na osnove transformacii diagramm isikavy [Development of knowledge bases by means of transforming fishbone diagrams]. Vestnik komp'yuternykh i informacionnykh tehnologij = Herald of computer and information technologies. № 4. 2018. Pp. 41–51. (in Russian).
6. Dorodnykh N.O., Nikolaychuk O.A., Yurin A.Yu. Razrabotka metamodelej dlja podderzhki sinteza nechetkih baz znaniy [Metamodel engineering for supporting fuzzy knowledge base synthesis]. Vestnik komp'yuternykh i informacionnykh tehnologij = Herald of computer and information technologies. № 1. 2020. Pp. 34–47. (in Russian).
7. Dorodnykh N.O., Yurin A.Yu. Ispol'zovanie diagramm klassov UML dlja formirovaniya produkcionnykh baz znaniy [Using UML Class Diagrams for Design of Knowledge Bases of Rule-Base Expert Systems]. Programmnyaya inzheneriya = Software engineering. № 4. 2015. Pp. 3–9. (in Russian).
8. Dorodnykh N.O., Yurin A.Yu. Ispol'zovanie koncept-kart dlya avtomatizirovannogo sozdaniya produkcionnykh baz znaniy [Using concept maps for rule-based knowledge bases engineering]. Programmnye produkty i sistemy = Software & System. № 4. 2017. Pp. 658–662. (in Russian).
9. Dorodnykh N.O. Web-based software for automating development of knowledge bases on the basis of transformation of conceptual models. Otkrytye-semanticheskie-tehnologii-proektirovaniya-intellektualnykh-sistem = Open Semantic Technologies for Intelligent Systems. 2017, № 7. Pp. 145–150.
10. Stennikov V.A., Barakhtenko E.A., Sokolov D.V. Primenenie ontologij pri realizacii koncepcii modelno upravlyaemoj razrabotki programmogo obespecheniya dlya

- proektirovaniya teplosnabzhayushchih sistem [ Usage of ontologies in the implementation of the concept of model-driven engineering for the design of heat supply systems]. *Ontologiya proektirovaniya = Ontology of design*. 2014, vol. 14, № 4. Pp. 54–68. (in Russian).
11. Zadeh L. The concept of a linguistic variable and its application to approximate reasoning. Moscow: Mir. 1976. (in Russian).
  12. Arendt T., Biermann E., Jurack S., Krause C., Taentzer G. Henshin: advanced concepts and tools for in-place EMF model transformations // *Model Driven Engineering Languages and Systems. MODELS 2010. Lecture Notes in Computer Science*. Vol. 6394. 2010. Pp. 121–135.
  13. Balasubramanian D., Narayanan A., Buskirk C., Karsai G. The graph rewriting and transformation language: GReAT // *Electronic Communications of the EASST*. Vol. 1. 2006. Pp. 1–8.
  14. Baumeister J., Striffler A. Knowledge-driven systems for episodic decision support // *Knowledge-Based Systems*. 2015. Vol. 88. Pp. 45–56.
  15. Biacino L., Gerla G. Fuzzy logic, continuity and effectiveness // *Archive for Mathematical Logic*. Vol. 41, № 7. 2002. Pp. 643–667.
  16. Cretu L.G., Florin D. *Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice*. Apple Academic Press. 2014. 350 p.
  17. Czarnecki K., Helsen S. Feature-based survey of model transformation approaches // *IBM Systems Journal*. Vol. 45, № 3. 2006. Pp. 621–645.
  18. Dorodnykh N.O., Yurin A.Y. A domain-specific language for transformation models // *CEUR Workshop Proceedings. Proceedings for First Scientific-practical Workshop Information Technologies: Algorithms, Models, Systems*. Vol. 2221. 2018. Pp. 70–75.
  19. Dubois D., Prade H., Yager R.R. *Readings in Fuzzy Sets for Intelligent Systems*. 1st ed. // Elsevier. 2014.
  20. Epsilon. 2017. URL: <http://www.eclipse.org/epsilon/> (дата обращения: 2020-06-20).
  21. Extended Event Tree Editor (EETE). 2020. Доступ: [https://github.com/LedZeppelin/event\\_tree\\_editor](https://github.com/LedZeppelin/event_tree_editor) (дата обращения: 2020-06-20).
  22. Orchard R.A. *FuzzyCLIPS Version 6.04A – User’s Guide*. Institute for Information Technology, National Research Council Canada. 1995.
  23. Jouault F., Allilaire F., Bézivin J., Kurtev I. ATL: A model transformation tool // *Science of Computer Programming*. Vol. 72, № 1. 2008. Pp. 31–39.
  24. Kosko B. Fuzzy Cognitive Maps // *International Journal of Man-Machine Studies*. Vol. 24. 1986. Pp. 65–75.
  25. Mens T., Gorp P.V. A Taxonomy of Model Transformations // *Electronic Notes in Theoretical Computer Science*. Vol. 152. 2006. Pp. 125–142.
  26. Novak V. On fuzzy type theory // *Fuzzy Sets and Systems*. Vol. 149, № 2. 2005. Pp. 235–273.
  27. Query/View/Transformation (QVT) Version 1.3 // *OMG Document formal/2016-06-03*. 2016. URL: <http://www.omg.org/spec/QVT/1.3/> (дата обращения: 2020-06-20).



28. Senol Y.E., Aydogdu Y.V., Sahin B., Kilic I. Fault Tree Analysis of chemical cargo contamination by using fuzzy approach // Expert Systems with Applications. Vol. 42, № 12. 2015. Pp. 5232–5244.
29. Sicilia M.A., Garcia E., Gutierrez J.A. Integrating fuzziness in object oriented modeling language: towards a fuzzy-UML // Proceedings of International Conference on Fuzzy Sets Theory and its Applications. 2002. Pp. 66–67.
30. Wang D., Liu E., Huang L. Event Tree Analysis of Xinda Oil Pipeline Leakage Based on Fuzzy Set Theory // The Open Civil Engineering Journal. Vol. 11. 2017. Pp. 101–108.
31. XSL Transformations (XSLT) Version 2.0. 2007. Доступ: <http://www.w3.org/TR/xslt20> (дата обращения: 2020-06-20).
32. Yurin A.Yu., Dorodnykh N.O., Nikolaychuk O.A., Grishenko M.A. Designing rule-based expert systems with the aid of the model-driven development approach // Expert Systems. Vol. 35, № 5. 2018. Pp. 1–23.
33. Zhang F., Ma Z. M., Yan L. Representation and reasoning of fuzzy ER model with description logic // Proceedings of IEEE International Conference on Fuzzy Systems. 2008. Pp. 1358–1365.