

KNOWLEDGE BASES ENGINEERING ON THE BASIS OF FAULT TREES ANALYSIS

Alexandr F. Berman⁽¹⁾, Nikita O. Dorodnykh⁽¹⁾, Olga A. Nikolaychuk⁽¹⁾, Alexandr Yu. Yurin⁽¹⁾

⁽¹⁾Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russia

The paper describes an approach for the computer-aided development of rule-based knowledge bases on the basis of analysis of fault trees represented with the aid of the author's notation. The approach is based on the extraction of structural elements from the fault trees specification and converting them to the target constructions of knowledge representation language, in particular, CLIPS. The description of main approach stages, the analyzed structures of fault trees in the XML-format are presented.

Keywords: knowledge acquisition, knowledge base, fault trees, model transformation, code generation, CLIPS

РАЗРАБОТКА БАЗ ЗНАНИЙ НА ОСНОВЕ АНАЛИЗА ДЕРЕВЬЕВ ОТКАЗОВ

Берман А.Ф.⁽¹⁾, Дородных Н.О.⁽¹⁾, Николайчук О.А.⁽¹⁾, Юрин А.Ю.⁽¹⁾

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

Рассмотрен подход к автоматизированному созданию продукционных баз знаний на основе анализа деревьев отказов, представленных при помощи авторской нотации. Подход основан на выделении структурных элементов деревьев отказов и их преобразования в конструкции целевого языка представления знаний, в частности, CLIPS. Приведено описание основных этапов подхода, анализируемых конструкций рассматриваемого формата деревьев отказов.

Ключевые слова: получение знаний, база знаний, деревья отказов, трансформация моделей, генерация кода, CLIPS

Введение. В настоящее время разработка новых методов и подходов к созданию интеллектуальных систем и, в частности, проблемно-ориентированных систем анализа риска опасных технических систем остается перспективной областью научных исследований [1, 8-9]. Базовым компонентом (ядром) таких систем является база знаний (БЗ), представленная на определенном языке представления знаний (ЯПЗ). Основную сложность при разработке БЗ представляет этап формализации [6], эффективность которого может быть повышена путем автоматической генерации кода БЗ на основе трансформации различных концептуальных моделей (например, диаграмм классов UML, концепт-карт и т.д.) [7].

На сегодняшний день существует множество различных языков для представления концептуальных моделей предметной области (например, DFD, IDEF0, IDEF5, UML и т.д.), а также разрабатываются нотации для отражения специфических особенностей мо-

делей представления знаний (например, Rule Visual Modeling Language (RVML) [10]). Подобные нотации ориентированы, прежде всего, на системных аналитиков, программистов, инженеров по знаниям и т.д. При этом специалисты-предметники предпочитают модели общесистемной направленности и ориентированные на систематизацию знаний или поддержку принятия решений (например, концептуальные карты, диаграммы Венна и Ишикавы, и т.д.). К таким моделям можно отнести «деревья отказов» и «деревья событий», применяемые в области анализа отказов и риска технических систем (fault tree analysis, event tree analysis). Результаты моделирования в нотациях подобного типа не только содержат специальные знания, используемые при решении предметных задач, но и являются удобным и понятным для специалиста-предметника способом их представления.

Для построения деревьев событий и отказов существуют программные системы в форме редакторов с возможностью выполнения специализированных расчетов. Анализ подобных систем показал, что они не предполагают дальнейшего использования знаний, заключенных в созданных специалистами деревьях, и направлены на решение частных задач.

Целью данной работы является создание подхода к автоматизированной разработке производственных БЗ на основе анализа деревьев отказов. В качестве целевого ЯПЗ выбран – CLIPS (C Language Integrated Production System).

Задача анализа деревьев отказов и их преобразования в БЗ может быть сведена к задаче трансформации моделей, представляющей, в общем случае, процесс автоматической генерации целевой модели по исходной модели, в соответствии с набором правил трансформации (преобразования). При этом под правилом преобразования подразумевается описание того, как одна или более конструкций на исходном языке моделирования может быть преобразована (mapping) в одну или более конструкций на целевом языке моделирования [3].

Трансформация моделей является одной из основных составляющих модельно-ориентированного (-управляемого) подхода к разработке программного обеспечения (Model-Driven Engineering, MDE) [3]. В рамках данного подхода, процесс разработки программного обеспечения представляет собой последовательное преобразование моделей с разным уровнем детализации, где на завершающем этапе генерируется исходный код программы.

Формализуем постановку задачи, в которой необходимо определить оператор преобразования концептуальной модели T :

$$T : CM^{FT} \rightarrow KB, \quad (1)$$

где CM^{FT} – исходная концептуальная модель (дерево отказов); KB – целевая БЗ, при этом $KB = \langle Code^{CLIPS} \rangle$.

Структура деревьев отказов. Дерево отказов, ДО (дерево неисправностей) – упорядоченное графическое представление логико-вероятностной связи случайных событий (нарушений, отказов, ошибок и т.д.), приводящих к реализации нежелательного конечного события. Анализ дерева отказов заключается в определении и анализе условий и факторов (комбинации отказов (неполадок) оборудования, инцидентов, ошибок персонала и нерасчетных внешних (техногенных и природных) воздействий), которые приводят или могут привести к возникновению негативных завершающих событий – полной или частичной

утрате функций, деградации рабочих характеристик изделия, ухудшению безопасности или других важных рабочих свойств. При построении дерева отказов используется обратная логика.

Авторами предложено расширить существующую модель ДО и визуальную нотацию их представления для получения более полной информации об исследуемых процессах возникновения отказов [4]. В частности, на основании результатов системного анализа проблемы исследования динамики технического состояния механической системы выделены стадии развития обозначенных процессов (субмикроуровень, микроуровень, мезоуровень, макроуровень) и элементы их описания (механизм и кинетика). В свою очередь кинетика, рассматриваемая как последовательность событий, должна быть детализирована описанием параметров (характеристик) событий. В результате в обобщенном виде получен шаблон дерева, описывающий стадии, последовательность событий (кинетика) и механизмы их возникновения (рис. 1).

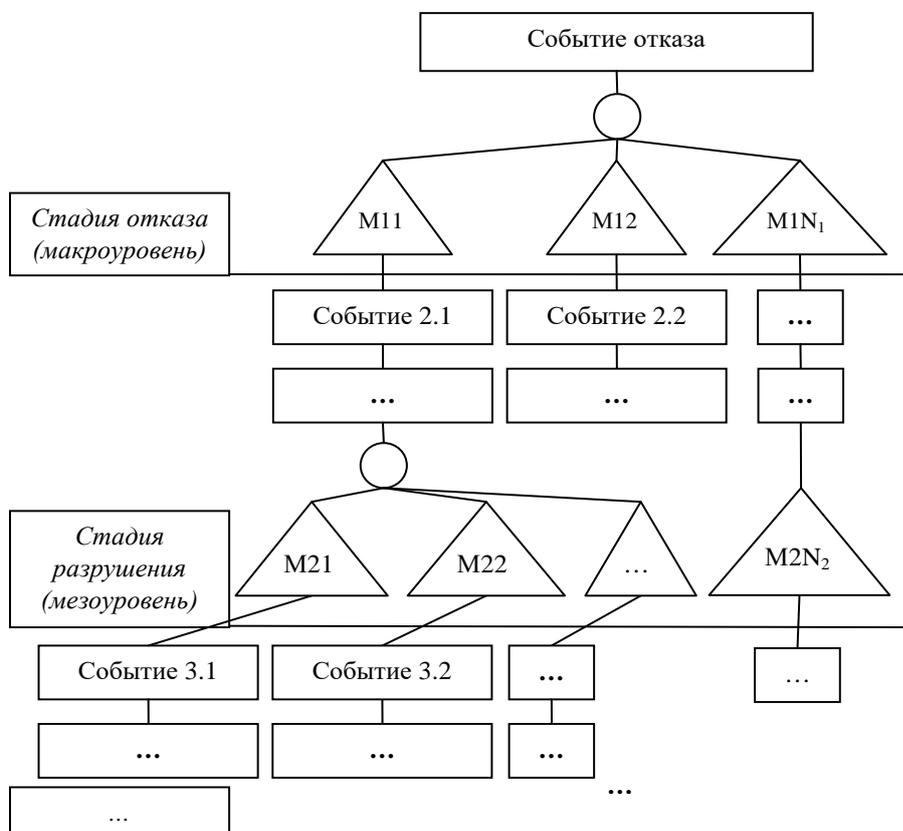


Рис. 1. Шаблон расширенной модели ДО, ○ – логический оператор, И, ИЛИ, исключаящее ИЛИ

При отсутствии описаний элементов дерева, расширяющих классическую нотацию, мы получим общепринятое на данный момент дерево.

Особо необходимо сказать о механизме процесса (в предлагаемой визуальной нотации – это треугольник). Механизм процесса – это совокупность свойств объекта и факторов, воздействующих на него. Механизм может быть описан в виде продукции (правила) следующей структуры (шаблона правила) [4]:

ЕСЛИ (Свойство объекта₁ **И**... **И** Свойство объекта_n) **И**
 (Воздействующий фактор₁ **И**... **И** Воздействующий фактор_n)

ТО Механизм j -стадии развития i -исследуемого процесса **И** (Событие₁ о...оСобытие _{n}), где о – некоторая логическая операция, $o \in \{\wedge, \vee, \oplus\}$.

Метамодель (abstract syntax) расширенной модели ДО, определяющая в абстрактной форме основные концепты, из которых состоит ДО, представлена на рисунке 2. Построенная метамодель соответствует мета-метамодели Esore и в дальнейшем используется в качестве исходной метамодели при разработке правил трансформации (модели трансформации), описывающих соответствия между элементами данной метамодели и целевой метамодели БЗ.

На настоящее время не существует единого общепринятого формата (стандарта текстовой нотации) для представления ДО, пригодного для машинной обработки. Учитывая этот факт, а также исходя из специфики расширения модели ДО дополнительными элементами (механизм и кинетика), принято решение о разработке собственного формата (спецификации) сериализации ДО на XML. Формат XML является универсальным и наиболее распространенным способом интеграции программных систем и обеспечения обмена информацией между приложениями.

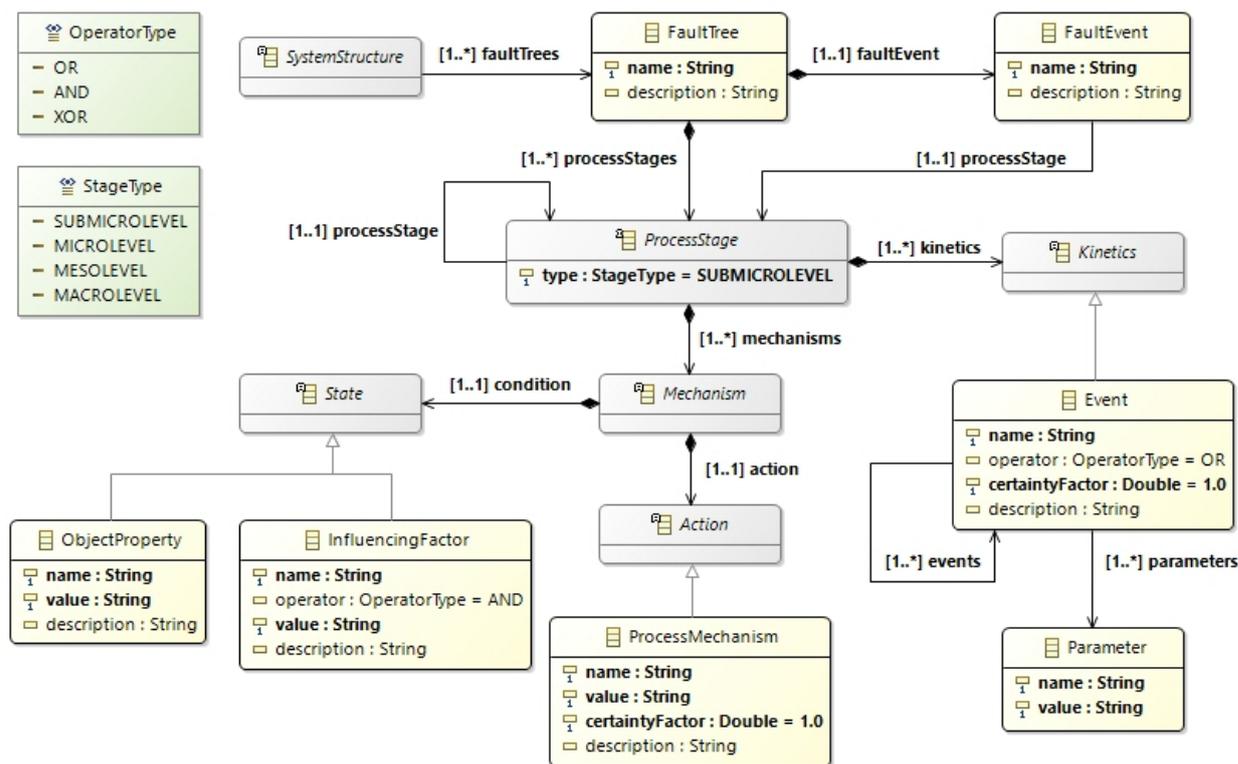


Рис. 2. Метамодель расширенной модели ДО

Описание разработанного XML-формата (concrete syntax) расширенной модели ДО, определяющее основные XML-конструкции, представлено в таблице 1.

Алгоритм преобразования деревьев событий. Для решения поставленной задачи (1), специализируем обобщенный алгоритм трансформации концептуальных моделей в код БЗ, описанный в работе [5]. Основной задачей разработанного алгоритмического обеспечения является преобразование (mapping) элементов расширенной модели ДО,

представленной в формате XML, в конструкции ЯПЗ CLIPS, что может быть представлено в виде последовательности действий.

На этапах 1 и 2 такого преобразования, **средствами** внешних программ (редакторов), пользователь (эксперт) строит ДО, описывающее последовательности событий, приводящие к рассматриваемому отказу, и представляет (сохраняет) его в формате XML. На этапе 3 процесса анализа XML-структуры ДО, выделяются элементы дерева и их отношения. Далее (этап 4) на их основе автоматически формируется (генерируется) модель продукций, как универсальное представление знаний в виде продукционных правил, независимое от используемого ЯПЗ (например, CLIPS, JESS, Drools, RuleML, SWRL и т.д.). Данная модель может содержать правила с несколькими условиями и разными типами операторов (например, «and», «or», «not»). Описание модели продукции приводится в [7]. При помощи специальной графической нотации RVML [10] предоставляется возможность визуализации, модификации (проверки) полученных продукций (этап 5). В алгоритме преобразования учитывается, что при построении ДО используется обратная логика, поэтому продукция создается на основе анализа дерева «снизу вверх» в отличие от построения дерева событий. На этапе 6 происходит генерация кода БЗ в формате CLIPS на основе сформированной модели продукций.

Таблица 1. XML-формат представления расширенной модели ДО

Элементы XML структуры ДО	Описание
System	Информация об исследуемом объекте, например, завод, линия, отделение, механическая система.
Component	Информация о структурной иерархии исследуемого объекта
Element	Информация о структурном элементе механической системы
FaultTree	Общая информация о ДО
ProcessStage	Общая информация о стадии процесса
FaultEvent	Информация об событии отказа
Mechanism	Общая информация о механизме процесса
ObjectProperty	Информация о свойстве объекта (элемента системы)
InfluencingFactor	Информация о воздействующем факторе
ProcessMechanism	Информация о механизме стадии развития исследуемого процесса
Kinetics	Общая информация о кинетике
Event	Информация о конкретном событии
Parameter	Информация о параметрах события
Operator	Логический оператор: И, ИЛИ, исключающее ИЛИ

Таким образом, уточним оператор преобразования концептуальной модели из (1):

$$T = \langle T_{CM-RM}, T_{RM-KB} \rangle, T_{CM-RM} : M_{XML}^{FT} \rightarrow M^{PR}, T_{RM-KB} : M^{PR} \rightarrow Code^{CLIPS} \quad (2)$$

где T_{CM-RM} – оператор преобразования ДО в модель продукций; T_{RM-KB} – оператор модели продукций в код БЗ на ЯПЗ CLIPS; M_{XML}^{FT} – представление расширенной модели ДО в XML-формате; M^{PR} – представление полученных знаний в виде модели продукций; $Code^{CLIPS}$ – код БЗ на ЯПЗ CLIPS.

Реализация алгоритма. Реализация разработанного алгоритмического обеспечения осуществлялась с использованием специализированной технологии автоматизации разработки программных компонентов систем, основанных на знаниях, обеспечивающих генерацию кода БЗ на основе трансформации концептуальных моделей [5] и соответствующего веб-ориентированного программного средства – Knowledge Base Development System (KBDS) [2]. В частности, разработан экспериментальный программный компонент для генерации модели продукции на основе трансформации ДО. Процесс разработки соответствует этапам создания программного компонента, описанным в [5].

Заключение. В настоящей работе предлагается подход к автоматизированной разработке продукционных БЗ для проблемно-ориентированных систем анализа риска опасных технических объектов. Подход основан на анализе структурных элементов ДО, расширенных конструкциями механизма процесса и кинетики, и их преобразования в конструкции целевого ЯПЗ CLIPS. Такой подход позволяет, с одной стороны, избежать ошибок программирования на этапе формализации знаний за счет автоматической кодогенерации, а также сократить время, затрачиваемое на разработку БЗ, с другой – приблизить специалистов-предметников к непосредственной разработке интеллектуальных компонентов проблемно-ориентированных программных систем, позволяя им создавать программный код, оперируя понятными предметно-ориентированными моделями.

Работа выполнена при частичной финансовой поддержке РФФИ (гранты № № 18-07-01164, 18-08-00560).

Результаты получены при использовании сетевой инфраструктуры ЦКП «Интегрированная информационно-вычислительная сеть Иркутского научно-образовательного комплекса» (<http://net.icc.ru>).

ЛИТЕРАТУРА

- [1] Berman A.F., Nikolaychuk O.A., Yurin A.Yu., Pavlov A.I. A methodology for the investigation of the reliability and safety of unique technical systems. Part O: Journal of Risk and Reliability. 2014. Vol. 228, No. 1. Pp. 29–38.
- [2] Dorodnykh N.O. Web-based software for automating development of knowledge bases on the basis of transformation of conceptual models // Open Semantic Technologies for Intelligent Systems. 2017. Vol.7. P. 145-150.
- [3] Kleppe A., Warmer J., Bast W. MDA Explained: The Model-Driven Architecture: Practice and Promise. 1st ed. Addison-Wesley Professional, 2003. 192 p.
- [4] Берман А.Ф., Николайчук О.А., Павлов Н.Ю., Юрин А.Ю. Методы и средства автоматизированного построения деревьев событий и отказов // Автоматизация и современные технологии. 2013. №9. С. 8–16.
- [5] 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.
- [6] Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И. Инженерия знаний. Модели и методы. СПб.: Лань, 2016. 324 с.

- [7] 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.
- [8] Массель Л.В., Массель А.Г. Интеллектуальные вычисления в исследованиях направлений развития энергетики // Известия Томского политехнического университета. 2012. Т. 321. № 5. Управление, вычислительная техника и информатика. С. 135–141.
- [9] Ноженкова Л.Ф., Ничепорчук В.В., Ноженков А.И. Создание комплексной системы безопасности региона на основе системной интеграции технологий // Информатизация и связь. 2013. №2. С. 122–124.
- [10] Юрин А.Ю. Нотация для проектирования баз знаний производственных экспертных систем // Объектные системы. 2016. №12. С. 48–54.