

Использование деревьев событий при автоматизации и интеллектуализации диагностирования и ремонта авиационной техники

Ю.В. КОТЛОВ,

канд. техн. наук

(МГТУ ГА, Иркутский
филиал, Иркутск),

О.А. НИКОЛАЙЧУК,

д-р техн. наук,

А.Ю. ЮРИН,

д-р техн. наук

(Институт динамики систем
и теории управления

им. В.М. Матросова СО РАН,
Иркутск)

iskander@icc.ru

Рассмотрено использование графического формализма деревьев событий при создании прототипов баз знаний интеллектуальных систем, при этом знания представлены в декларативной форме в виде логических правил, описывающих взаимосвязи между внешними признаками неисправности, возможными причинами и операциями по их локализации и устранению. В качестве методологической основы используется разработанная ранее технология прототипирования производственных экспертных систем, реализующая принципы визуального программирования и модельных трансформаций. Описано ее применение при создании баз знаний для авиационной диагностики, включая основные модели и этапы, приведен пример. Разработаны алгоритмы и программное обеспечение в форме специализированного редактора деревьев событий и модуля расширения для Personal Knowledge Base Designer.

Диагностирование воздушных судов, интеллектуализация, автоматизация, разработка баз знаний, преобразования, деревья событий

Решение научно-практических задач в области диагностирования, ремонта и технического обслуживания воздушных судов требует автоматизации и интеллектуализации [1–3], под которыми подразумевается использование методов и систем искусственного интеллекта в контексте оперативной поддержки принятия решений [4–6]. Особенно это важно для новых типов самолетов – «Сухой Суперджет», МС-21, – для которых в настоящее время отсутствует репрезентативная статистика отказов. Хотя текущий уровень исследований в области искусственного интеллекта достаточно высок, проблема повышения эффективности и качества разработки программного обеспечения систем искусственного интеллекта, включая базы знаний, по-прежнему актуальна и характеризуется высокими временными затратами и необходимостью привлечения и взаимодействия специалистов различной квалификации, в том числе профессиональных программистов. Для решения этой проблемы предлагаются новые языки, инструментальные средства, методы и технологии. В то же время наблюдается тенденция к более полному включению конечных пользователей в процесс создания систем искусственного интеллекта с передачей им отдельных функций, которые исторически реализовывались программистами. Подобные подходы объединены в направление разработки, ориентированной на конечных пользователей, – End-User Development (EUD) [7]. Основная идея EUD заключается в том, чтобы дать возможность конечным пользователям самим адаптировать и настраивать приложения под особенности их задач. В этом контексте идеи EUD реализуются посредством методов визуального программирования, проблемно-ориентированных языков, модельно-ориентированного подхода и др. [8].

В рамках парадигмы EUD была разработана оригинальная модельно-ориентированная методология прототипирования экспертных систем на основе трансформаций [9], а также реализующие ее программные средства [10], которые применялись при решении ряда практических задач [9, 11]. В данной работе описывается еще одно применение разработанной методологии при создании прототипов баз знаний для поиска и устранения неисправностей самолета «Сухой Суперджет», в частности, его системы электрооборудования. Решение данной задачи потребовало расширения методологии и средств с точки зрения поддержки формализма деревьев событий и их преобразования в программные коды и спецификации. В частности, были созданы алгоритмы и программные средства – специализированный редактор для построения деревьев событий Extended Event Tree Editor (EETE) [12] и плагин PKBD.ExTree для системы прототипирования баз знаний Personal Knowledge Base Designer (PKBD) [10].

Далее рассмотрим основные принципы модельно-ориентированного подхода и методологии прототипирования экспертных систем на основе трансформаций, а также приведем подробное описание использования деревьев событий при разработке прототипов баз знаний для диагностики воздушных судов.

Модельно-ориентированный подход (Model-Driven Development (MDD)) [13, 14] представляет собой методологию разработки программного обеспечения, в которой информационные (или концептуальные) модели являются основными артефактами и могут быть использованы для получения других моделей, исходных кодов и спецификаций. Данная методология позволяет конечным пользователям, которые могут быть программистами или непрограммистами (в зависимости от реализации методологии), создавать и настраивать программное обеспечение.

Приведем основные принципы.

Модель – абстрактное описание системы (или процесса) на формальном языке (в частности, UML (Unified Modeling Language)). Все модели могут быть сериализованы (представлены) в форматах, подобных XML (Extensible Markup Language). Модели трансформируются (преобразовываются) от более абстрактных к менее абстрактным. Преобразование (трансформация) [15] модели – это автоматическая генерация целевой модели из исходной модели в соответствии с набором правил преобразования. В этом случае каждое правило преобразования описывает соответствие между элементами исходной и целевой метамodelей.

Методология MDD имеет несколько реализаций: Model Driven Architecture (MDA), Eclipse Modeling Framework (EMF), Model Integrated Computing (MIC), Microsoft Software Factories и JetBrains MPS. Различные варианты MDD применяются для разработки приложений баз данных, агентно-ориентированных приложений, систем поддержки принятия решений и интеллектуальных систем [16–18].

В работе [8] была предложена специализация MDD/MDA для разработки баз знаний и экспертных систем, которая включает следующие элементы:

- оригинальный метод проектирования декларативных (продукционных и прецедентных) баз знаний для интеллектуальных систем, расширяющий принципы MDD за счет новых моделей, языков и платформ в контексте разработки баз знаний;
- язык визуального проектирования логических правил RVML (Rule Visual Modeling Language), основанный на UML и расширяющий его выразительные возможности в контексте моделирования декларативных баз знаний;
- программное обеспечение, поддерживающее разработанные язык и метод.

Подробное описание методологии, программных средств и примеров ее применения приведено в работе [9]. На рис. 1 показаны основные элементы прототипирования экспертных систем на основе трансформаций, используемые в сравнении с MDD.

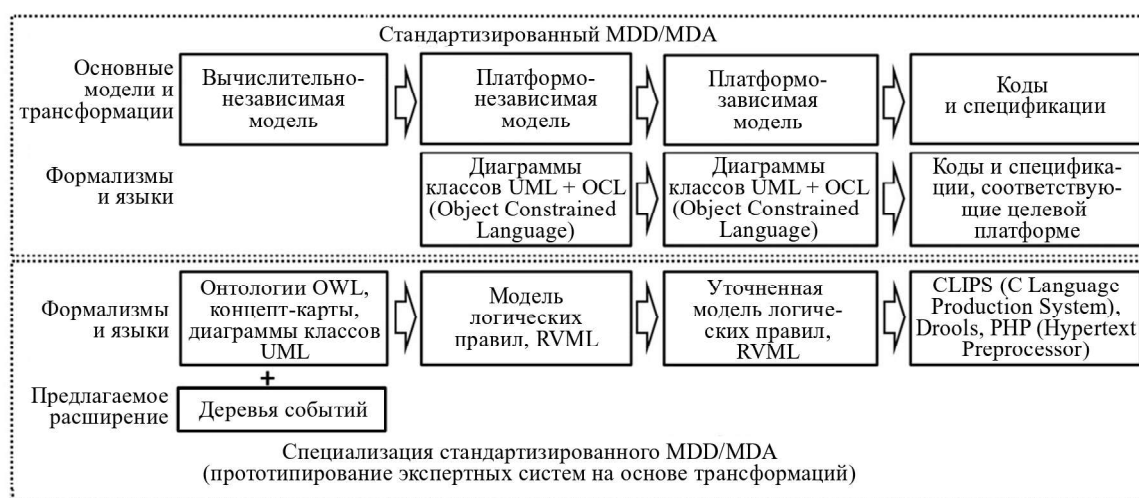


Рис. 1

Актуальность задач диагностирования, ремонта и технического обслуживания самолетов связана с проектированием новых самолетов, эксплуатацией и совершенствованием существующих [1, 5], при

этом активно используется ряд систем поддержки принятия решений типа электронных технических руководств, автоматизирующих процесс диагностирования и технического обслуживания, в частности Airbus AirNav [19] и MyBoeing Fleet [20]. Однако подобное программное обеспечение является закрытым и узкоспециализированным, поэтому перспективно создание аналогичного программного обеспечения для самолета «Сухой Суперджет» и других воздушных судов российского производства, но с дополнительными возможностями, обеспечивающими не только логический вывод в части формирования последовательности работ (операций) по поиску, подтверждению и устранению неисправностей на основе нормативных документов (например, руководств по поиску и устранению неисправностей), но и фиксацию тех неисправностей, которые не учтены в текущих редакциях документации, но фиксируются в форме карт учета неисправностей авиационной техники.

В данной работе решается задача, связанная с разработкой подобной программной системы [21], в частности, рассмотрен подход к созданию прототипов баз знаний диагностической интеллектуальной системы поддержки принятия решений непрограммирующими пользователями посредством визуального программирования и применения адаптированной методологии прототипирования экспертных систем на основе трансформаций [9].

Ранее в результате решения практических задач с помощью этой методологии [9, 10] были определены основные этапы предлагаемого подхода и рекомендуемые программные средства (указаны в скобках на рис. 2).

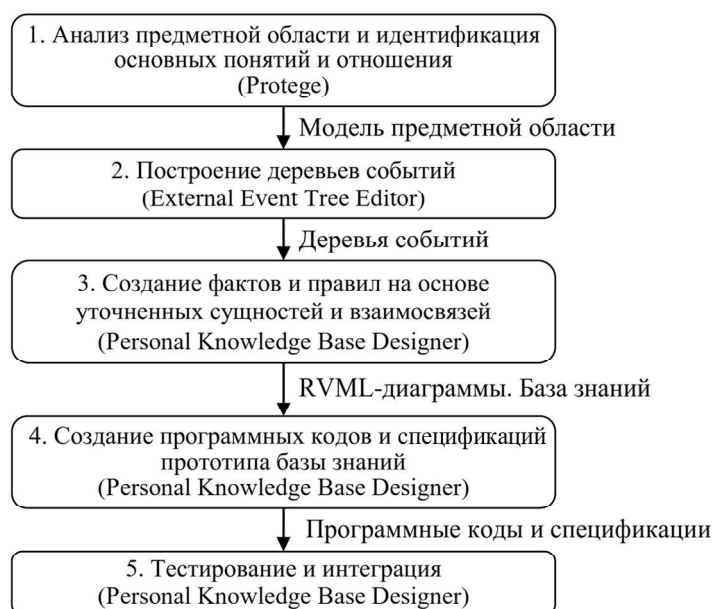


Рис. 2

На первом этапе выполняется анализ предметной области для определения основных понятий и взаимосвязей предметной области. Для решения этой задачи было использовано программное средство Protege, обеспечивающее построение онтологических моделей. В качестве источников информации использовались руководства по технической эксплуатации, а также по поиску и устранению неисправностей системы электроснабжения самолета «Сухой Суперджет».

В результате была разработана онтология диагностирования системы электроснабжения самолета «Сухой Суперджет» [22], включающая 355 понятий и 1066 аксиом, описывающих основные системы, подсистемы и операции в соответствии с их различными классификациями. Основная цель этого этапа состояла в определении списка основных операций (работ) и их взаимосвязей в контексте проявления определенных признаков неисправностей или отказов.

В качестве формализма для визуализации предметных знаний на следующем этапе было предложено использовать деревья событий, являющиеся сценарным методом системного анализа для описания закономерностей функционирования сложных объектов [23]. Для обеспечения использования деревьев

было разработано программное обеспечение для создания деревьев событий Extended Event Tree Editor (EETE), которое является частью платформы системы моделирования знаний [12]. Одной из особенностей EETE является интеграция с онтологиями, сериализованными в формате OWL (Web Ontology Language) в части импорта понятий и отношений.

На втором этапе был осуществлен импорт разработанной онтологии и формирование сценариев (в виде деревьев событий), описывающих цепочки операций по выявлению и устранению неисправностей и отказов. На рис. 3 показан фрагмент одной из построенных моделей.

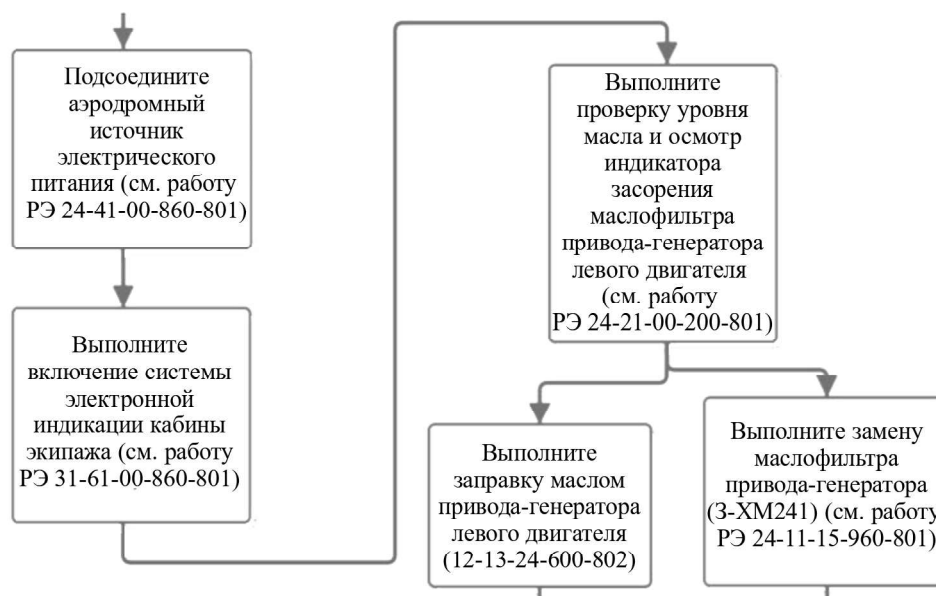


Рис. 3

На третьем этапе построенные деревья событий были импортированы в PKBD для получения структур базы знаний. Для поддержки импорта был разработан модуль PKBD.ExTree, обеспечивший сопоставление элементов дерева событий с элементами базы знаний на уровне форматов хранения данных. Соответствия между форматами EETE и PKBD в упрощенном виде представлены следующими образом: Diagram – KnowledgeBase, Event – Template, Operator (lhs) – Condition, Operator (rhs) – Action, Parameter (class) – Slot (name).

При импорте взаимосвязи между событиями интерпретировались как причинно-следственные отношения. В результате были сформированы наборы правил со следующей структурой:

ЕСЛИ (Operation) ТО (Operation).

Далее в структуру правила был добавлен еще один элемент, описывающий внешние проявления неисправности:

ЕСЛИ (Operation) И (Malfunction) ТО (Operation).

На рис. 4 показан пример одного из полученных правил в виде диаграммы RVML (Rule Visual Modeling Language) и соответствующего ей кода на PHP (Hypertext Preprocessor) и CLIPS (C Language Integrated Production System), который может быть сгенерирован на следующем этапе.

В рамках подхода этапы 4 и 5 могут меняться местами, так как для выполнения тестирования и отладки нет необходимости явно генерировать программный код, соответствующий структурам базы знаний. Тестирование базы знаний подразумевает повторные ее запуски (прогоны) машиной вывода (интерпретатором) с разными наборами исходных фактов. При этом с точки зрения пользователя PKBD генерация кода не происходит: пользователь видит только результаты работы машины вывода в упрощенной табличной форме.

После успешной отладки база знаний может быть сохранена в форматах CLIPS, Drools или PHP для дальнейшей интеграции и использования при создании настольных и веб-приложений.

В работе рассмотрен способ создания баз знаний для поддержки авиационного диагностирования на примере системы электроснабжения самолета «Сухой Суперджет». Содержательная оценка эксперта-

ми полученных баз знаний показала их недостаточную семантическую выразительность из-за отсутствия информации, содержащейся в деревьях событий. В частности, при формировании цепочки правил также должна была учитываться информация о системе/подсистеме, последствиях и типе ремонта. Позже эта информация была добавлена, итоговая база знаний содержала 836 правил, описывающих последовательности операций при разных неисправностях.

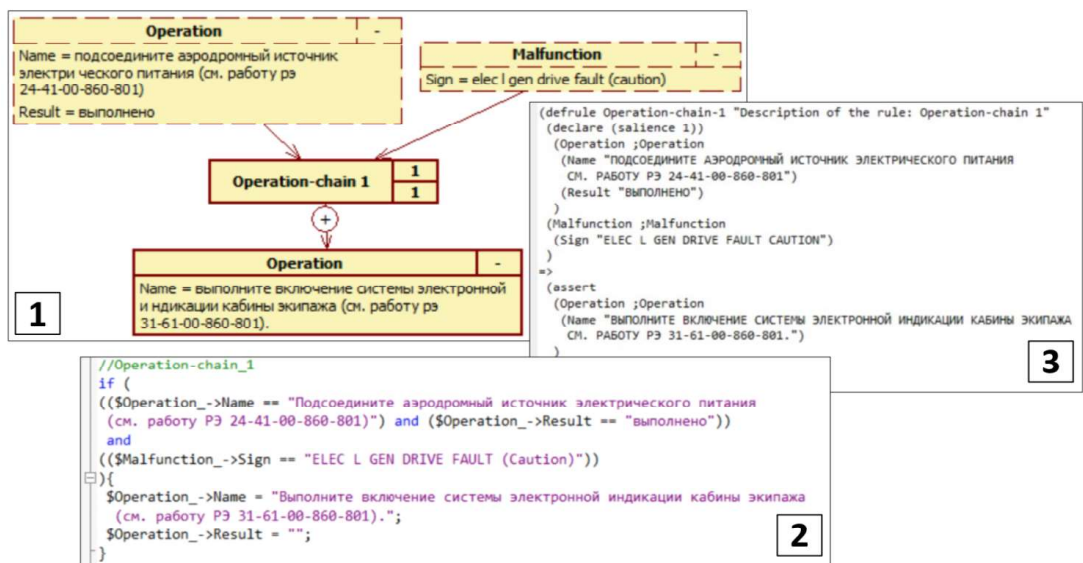


Рис. 4

Кроме того, было проведено качественное сравнение с другими методиками аналогичной направленности (OSTIS [24], IACPaaS [25] и др.), реализующими отдельные принципы EUD, которое выявило (таблица) их направленность на концептуализацию и формализацию знаний, а также крупноблочное моделирование; отсутствие доступного программного обеспечения, готового к использованию без модификации (настройки) программистами; слабую способность к интеграции существующих решений, в частности «по данным», с другим программным обеспечением и отсутствие возможности создания отчуждаемых кодов. Использованный подход обеспечил устранение этих недостатков, а также продемонстрировал преимущества перед прямым кодированием знаний благодаря ориентации на конечных пользователей, что позволило значительно расширить группу разработчиков прототипа базы знаний.

Критерий/ методология	Цели методологии	Реализованные EUD-направления	Поддерживающие инструментальные средства	Целевые языки программирования	Поддерживаемые языки моделирования	Интеграционные возможности
OSTIS (Open Semantic Technologies for Intelligent Systems) [24]	Проектирование, генерация кода, разработка программного обеспечения	Компонентная сборка, визуальное программирование, модельные трансформации	IMS.OSTIS METASYSTEM	SC (Semantic Code)	SC	Средние
IACPaaS (Intelligence Application Control Platform as a Service) [25]	Проектирование, интерпретация спецификаций, разработка программного обеспечения	Компонентная сборка, модельные трансформации	Облачная платформа IACPaaS	-	Язык описания модели, бинарный граф	-
CAKE (Computer Aided Knowledge Engineering Technique) [26]	Крупноблочное моделирование, описание логики	Визуальное программирование, модельные трансформации	-	-	Расширенные диаграммы пакетов UML	-
MOKA (Methodology and Tools Oriented to Knowledge-Based Engineering Applications) [27]	Крупноблочное моделирование, описание логики, концептуализация, формализация	Визуальное программирование, модельные трансформации, мастера, шаблоны	-	-	MML (MOKA Modelling Language)	-

Критерий/ методология	Цели методологии	Реализованные EUD-направления	Поддерживаю- щие инстру- ментальные средства	Целевые языки про- граммиро- вания	Поддержи- ваемые язы- ки модели- рования	Интегра- ционные возмож- ности
Прототипирование экспертных систем на основе трансфор- маций [9]	Проектирование, генерация кода, интерпретация спе- цификаций, разра- ботка программно- го обеспечения	Визуальное про- граммирование, модельные трансформации, мастера	PKBD, EETE	CLIPS, DROOLS, PHP	UML, RVML, концепт- карты, онтологии	Высокие

Диагностирование неисправностей гражданских воздушных судов требует применения методов и систем искусственного интеллекта для повышения эффективности поддержки принятия решений, при этом разработка и применение новых методов создания программного обеспечения для систем искусственного интеллекта, включая базы знаний, остается актуальной задачей, особенно в контексте вовлечения конечных пользователей в этот процесс и делегирования им определенных функций, которые исторически реализовывались программистами.

В данной работе представлен подход к прототипированию баз знаний для диагностирования неисправностей воздушных судов на основе использования формализма деревьев событий и его трансформации в декларативную форму знаний в виде логических правил, описывающих взаимосвязи между внешними проявлениями, возможными причинами и операциями (работами) по их локализации и устранению.

Полученные результаты позволяют говорить о перспективности предлагаемого подхода. В дальнейшем планируется применение других методов визуальной формализации знаний, в частности диаграмм переходов состояний, которые позволили бы создавать циклические графы и не клонировать узлы (операции) при описании сценариев.

СПИСОК ЛИТЕРАТУРЫ

1. Кирпичев И.Г., Петров Д.В., Чинючин Ю.М. Многофункциональная интегрированная платформа сопровождения технической эксплуатации воздушных судов // Научный Вестник МГТУ ГА. 2020. № 6. С. 28–37.
2. Макаров Н.Н. Синтез алгоритма функционирования информационно-управляющей системы контроля и диагностики состояния общесамолетного оборудования // Изв. вузов. Авиационная техника. 2008. № 1. С. 46–50.
3. Перфильев О.В., Рыжиков С.Г., Должиков В.А. Интеллектуальная система поиска неисправности на самолете // Изв. СамНЦ РАН. 2018. № 4 (3). С. 326–331.
4. Сухих Н.Н., Рукавишников В.Л. Экспертные системы – средство информационной поддержки принятия решений экипажем самолета // Изв. вузов. Авиационная техника. 2022. № 2. С. 19–25.
5. Зрячев С.А., Ларин С.Н. Разработка базы знаний послепродажного обслуживания авиационной техники // Изв. СамНЦ РАН. 2020. № 5. С. 48–53.
6. Chiu C., Chiu N.-H., Hsu C.-I. Intelligent Aircraft Maintenance Support System Using Genetic Algorithms and Case-Based Reasoning // International Journal of Advanced Manufacturing Technology. 2004. Vol. 24. P. 440–446.
7. Barricelli B.R. et al. End-User Development, End-User Programming and End-User Software Engineering: A Systematic Mapping Study // Journal of Systems and Software. 2019. Vol. 149. P. 101–137.
8. Coronado E. et al. Visual Programming Environments for End-User Development of Intelligent and Social Robots, a Systematic Review // Journal of Computer Languages. 2020. Vol. 58. Article № 100970.
9. Дородных Н.О., Юрин А.Ю. Технология создания продукционных экспертных систем на основе модельных трансформаций. Новосибирск: Сибирское отделение РАН, 2019. 144 с.
10. Yurin A.Yu., Dorodnykh N.O. Personal Knowledge Base Designer: Software for Expert Systems Prototyping // SoftwareX. 2020. Vol. 11. Article № 100411.
11. Берман А.Ф. и др. Поддержка принятия решений на основе продукционного подхода при проведении экспертизы промышленной безопасности // Химическое и нефтегазовое машиностроение. 2014. № 11. С. 28–34.
12. Knowledge Modeling System [Electronical Resource]. URL: <http://kms.knowledge-core.ru/> (дата обращения: 3.07.2023).
13. Da Silva A.R. Model-Driven Engineering: A Survey Supported by the Unified Conceptual Model // Computer Languages, Systems & Structures. 2015. Vol. 43. P. 139–155.
14. Cretu L.G., Dumitriu F. Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice. Apple Academic Press, 2014. 374 p.

15. Mens T., Van Gorp P. A Taxonomy of Model Transformations // Electronic Notes in Theoretical Computer Science. 2006. Vol. 152. P. 125–142.
16. Nofal M.A., Fouad K.M. Developing Web-Based Semantic and Fuzzy Expert Systems Using Proposed Tool // International Journal of Computer Applications. 2015. Vol. 112. № 7. P. 38–45.
17. Cabello M.E. et al. Baseline-Oriented Modeling: An MDA Approach Based on Software Product Lines for the Expert Systems Development // Proc. of the First Asian Conference on Intelligent Information and Database Systems, April 1–3, 2009. Dong Hoi, Quang Binh, Vietnam. P. 208–213.
18. Canadas J., Palma J., Tunes S. Defining the Semantic of Rule-Based Web Applications Through Model-Driven Development // International Journal of Applied Mathematics and Computer Science. 2011. Vol. 21. № 1. P. 41–55.
19. AirNav-Maintenance [Electronical Resource]. URL: <https://www.airnav.com/maintenance/index.html> (дата обращения: 3.07.2023).
20. MyBoeing Fleet [Electronical Resource]. URL: <https://www.myboeingfleet.com> (дата обращения: 3.07.2023).
21. Yurin A. et al. Towards an Intelligent Decision Support System for Aircraft Troubleshooting // Proc. of the 10th International Conference on Recent Advances in Civil Aviation / Eds. O.A. Gorbachev, Xiaoguang Gao, Bo Li. Springer, 2023. P. 77–91.
22. Котлов Ю.В., Попов В.М., Юрин А.Ю. Онтология поиска и устранения неисправностей системы электроснабжения воздушного судна // Знания – онтологии – теории: Материалы 8-й Междунар. конф., г. Новосибирск, 8–12 нояб. 2021. Новосибирск: Институт математики им. С.Л. Соболева СО РАН, 2021. С. 141–148.
23. ГОСТ Р 54142-2010 Менеджмент рисков. Руководство по применению организационных мер безопасности и оценки рисков. Методология построения универсального дерева событий. М.: Стандартинформ, 2020. 33 с.
24. Голенков В.В., Гулякина Н.А. Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: принципы создания // Онтология проектирования. 2014. № 1. С. 42–64.
25. Грибова В.В. и др. Методы и средства платформы IACPAAS для семантического представления знаний и разработки декларативных компонентов интеллектуальных систем: Материалы междунар. науч.-техн. конф., г. Минск, 21 – 23 фев. 2019. Минск: БГУИР, 2019. С. 21–24.
26. Doherty P., Łukaszewicz W., Szalas A. CAKE – Computer Aided Knowledge Engineering Technique [Electronical Resource]. URL: <https://www.ida.liu.se/divisions/aiics/publications/ECAI-2002-CAKE-Computer-Aided.pdf> (дата обращения: 3.07.2023).
27. Managing Engineering Knowledge: MOKA: Methodology for Knowledge Based Engineering Applications / Ed. M. Stokes. Wiley, 2001. 310 p.

Поступила в редакцию 3.08.23

После доработки 11.11.23

Принята к публикации 14.12.23

Using the Event Trees in the Automation and Intellectualization of Aircraft Diagnostics and Repair

YU.V. KOTLOV¹, O.A. NIKOLAICHUK², AND A.YU. YURIN²

¹ Moscow State Technical University of Civil Aviation, Irkutsk Branch, Irkutsk

² V.M. Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of Russian Academy of Sciences, Irkutsk

The paper considers the use of the graphical formalism of event trees when creating prototypes of knowledge bases for intelligent systems. Here, the knowledge is represented in a declarative form as logical rules describing the relationship between external signs of a malfunction, possible causes, and operations for their localization and elimination. The previously developed technology of prototyping the rule-based expert systems, implementing the principles of visual programming and model transformations, is used as a methodological basis. Its application is described in the creation of knowledge bases for aviation diagnostics, including the main models and stages, an example is given. Algorithms and software were developed in the form of a specialized event tree editor and an extension module for Personal Knowledge Base Designer.

Aircraft diagnostics, intellectualization, automation, knowledge base development, transformations, event trees