

О подходе к построению баз знаний для задач авиационной диагностики

Н.О. ДОРОДНЫХ,
канд. техн. наук,
О.А. НИКОЛАЙЧУК,
д-р техн. наук

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

Ю.В. КОТЛОВ,
канд. техн. наук
(МГТУ ГА,
Иркутский филиал,
Иркутск),

А.Ю. ЮРИН,
д-р техн. наук
(ИРНИТУ, Иркутск)
iskander@icc.ru

Рассмотрено развитие подхода к построению прототипов декларативных баз знаний классических экспертных систем на основе модельных трансформаций использованием диаграмм переходов состояний и таблиц решений. Данные способы представления информации позволяют описать в виде логических правил взаимосвязи внешних проявлений неисправностей и причин, а также операций по поиску и устранению неисправностей. Подход апробирован при адаптации (настройке) приложения «АвиаТехПом.Терминал» для системы кондиционирования воздуха самолета Cessna 182T и систем электроснабжения, управления и кондиционирования воздуха самолета Sukhoi Superjet 100 (RRJ-95). Показана целесообразность применения описанного подхода для моделирования ветвящихся сценариев работ, верификации таблиц решений и технических руководств.

Диагностирование воздушных судов, автоматизация, разработка баз знаний, трансформация моделей, визуальные модели, диаграммы переходов состояний, таблицы решений, генерация кода

Одним из факторов, снижающих эффективность и масштаб применения методов искусственного интеллекта для решения задач диагностирования, ремонта и технического обслуживания воздушных судов, является в некоторой степени уникальность (незначительное количество) рассматриваемых объектов, что связано с ограниченным количеством данных, статистических показателей и узкой специализацией соответствующих программных систем. Интеллектуальные решения данной проблемы осуществляются в области создания классических экспертных систем [1–4] и систем распознавания неисправностей на основе нейросетевых моделей [5–7]. Необходимо отметить, что существует ограниченное количество готовых подобных решений, характеризующихся сложностью их настройки и адаптации к особенностям решаемых задач и свойств рассматриваемых объектов. В случае классических экспертных систем необходимы трудоемкие процессы, связанные с концептуализацией, формализацией и кодификацией предметных знаний [8], для нейросетевых моделей нужно осуществлять подготовку представительных наборов (корпусов) данных и обучение моделей [9, 10].

Определенным способом решения указанных проблем является создание программных решений в рамках парадигмы «разработка, ориентированная на конечных пользователей» (End-User Development (EUD)) [11]. Основная идея этой парадигмы состоит в обеспечении возможности адаптации приложений под особенности предметной области силами непрограммирующих пользователей их перенастройкой, перекомпиляцией или повторной сборкой без непосредственного взаимодействия с языками, средами или технологиями программирования. В этом контексте идеи EUD могут реализовываться посредством методов визуального программирования [12], проблемно-ориентированных языков, модельно-ориентированного подхода [13, 14] и другими способами, обеспечивающими сокрытие от конечного пользователя языковых конструкций или деталей программной реализации [15].

Одна из реализаций идей данного подхода [16, 17] использовалась при создании пилотной версии программы «АвиаТехПом» для обучения и интеллектуальной поддержки принятия оперативных решений техником при диагностировании и обслуживании воздушного судна [18]. В частности, при формировании специализированных продукционных баз знаний применялись деревья событий и программно-алгоритмическое обеспечение для их преобразования в программные коды и спецификации [15], что позволило выявить ряд ограничений и определить направления дальнейшего совершенствования подхода. В данной работе описывается развитие модельно-ориентированного подхода, основанное на применении диаграмм переходов состояний и таблиц решений.

«АвиаТехПом» является пилотной программной платформой для построения диагностических интеллектуальных систем поддержки принятия решений, использующих декларативные базы знаний [19]. Построенные на платформе программные системы, предназначенные для оперативной информационной поддержки техников аэродромных служб, позволяют повысить эффективность процессов за счет снижения затрат времени на выполнение работ по поиску и устранению неисправностей и отказов, обеспечить накопление информации по инцидентам для их дальнейшего использования.

Основные подсистемы платформы «АвиаТехПом» представлены на рис. 1.

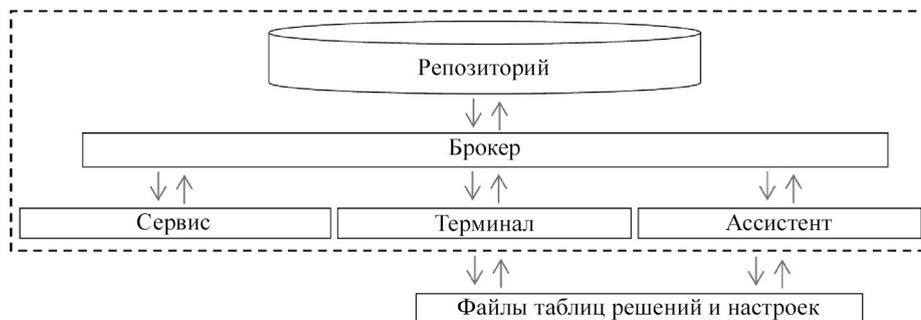


Рис. 1

Репозиторий – централизованное хранилище данных и знаний на основе СУБД PostgreSQL, содержит структуры, описывающие модель предметной области, включая такие понятия, как работа, материально-техническое обеспечение, признаки неисправности, система и др., а также отношения между ними [20].

С помощью специализированного модуля «Брокер» осуществляется обращение к хранилищу данных в рамках концепции микросервисной архитектуры посредством программных интерфейсов.

Веб-приложение «Сервис» реализует функции по поиску и устранению неисправностей и отказов на основе классических технологий искусственного интеллекта – производственных и прецедентных экспертных систем.

«Терминал» – настольное приложение аналогичной функциональности с веб-приложением, дополненное возможностью использования данных в форме локальных текстовых файлов формата CSV.

«Ассистент» – приложение в форме чат-виджета, взаимодействующее как с централизованным хранилищем, так и локальными файлами. С его помощью также реализуются функции поддержки поиска и устранения неисправностей. Приложение создано с использованием конструктора чат-ботов Aimylogic. Поддерживается возможность общения с пользователем на ограниченном естественном языке (текст) благодаря применению технологий генеративного искусственного интеллекта, в частности методов машинного обучения и большой языковой модели JustGPT.

Таким образом, «АвиаТехПом» представляет собой клиент-серверное приложение, состоящее из серверного «ядра» – репозитория и разных видов клиентских приложений, взаимодействие которых организовано согласно принципам REST API. При этом подсистемы платформы, напрямую взаимодействующие с пользователем («Сервис», «Терминал», «Ассистент»), имеют определенную степень адаптивности, т. е. отдельные элементы их интерфейса подстраиваются под информацию в структурах баз данных и знаний. Фактически большинство полей для ввода и отображения данных (метки с наименованием полей, поля для ввода данных строкового или числового вида, поля ввода в форме выпадающих списков) не определены заранее на этапе программирования, но реализованы механизмы динамического их создания или генерации во время исполнения приложения. Наименования отдельных элементов интерфейса (группы параметров, закладки, кнопки) могут быть заданы пользователем в специальных конфигурационных файлах.

Подобная связь элементов графического интерфейса приложений и структур данных обуславливает важность задачи их наполнения и поддержания в актуальном состоянии.

Для решения данной задачи использовались следующие системы.

1. Табличный редактор Microsoft Excel.

Широкодоступный, но в связи с большим количеством ошибок, связанных с ручным вводом данных, возникла необходимость разработки дополнительных программ-конверторов для загрузки полученных данных в репозиторий.

2. Универсальный редактор баз знаний Personal Knowledge Base Designer (PKBD) [21].

Реализует модельно-ориентированный подход через набор программ-мастеров (последовательностей диалоговых форм) и специализированных генераторов кодов. Более трудоемкий в сравнении с внешними табличными редакторами, особенно при необходимости описания разветвленных сценариев работ (операций).

3. Встроенный редактор таблиц решений подсистемы «Терминал».

Высокоэффективен за счет набора специализированных функций (унификация представления (коррекция данных), поиск ошибок, копирование строк (записей), ввод данных для продолжения цепочки правил, отображение цепочки правил, отображение следующего правила). Применение ограничено редактированием локальных файлов и др.

4. Веб-ориентированная программная система Knowledge Modeling System (KMS) [17].

В частности, активно применялся редактор деревьев событий при описании сценариев [15].

Использование KMS перспективно при необходимости моделирования сценариев (цепочек правил), описывающих последовательности работ, с использованием принципов визуального моделирования знаний и общепринятых нотаций (деревьев событий и отказов). Однако необходимо усовершенствование метода в контексте обеспечения возможности описания разветвленных сценариев со сложными условиями переходов. Подобная возможность была реализована, что позволило уточнить подход к созданию баз знаний на основе модельных трансформаций в контексте решения задач авиационной диагностики.

С методологической точки зрения предлагается воспользоваться специализированной модельно-ориентированной технологией [16] создания прототипов декларативных баз знаний интеллектуальных систем, использующих для поддержки принятия решений модели логических правил и прецедентов. Согласно технологии процесс разработки представляется в виде цепочки последовательных трансформаций концептуальных моделей различной степени абстракции.

Приведем основные этапы процесса разработки.

1. Построение модели предметной области, включая анализ и идентификацию основных понятий и отношений.

Модели строятся с использованием ряда популярных нотаций и средств, в частности UML-диаграмм классов (средства StarUML и XMind), онтологий (Protégé) или концепт-карт (CMapTools), и сохраняются (сериализуются) в XML-подобных форматах.

2. Построение моделей, описывающих последовательность и динамику процессов, на основе модели предметной области.

Модели строятся с использованием специализированных нотаций, в частности деревьев событий и отказов. При этом отношения между узлами моделей интерпретируются как единицы знаний, соответствующие простейшим логическим правилам, описывающим предметные знания. Для использования информации из моделей предметной области создаются трансформационные операторы (или операторы преобразования), которые преобразовывают элементы модели предметной области в элементы модели динамики на основе анализа XML-подобных форматов.

3. Создание базы знаний на основе уточненных сущностей и взаимосвязей, содержащихся в моделях динамики.

База знаний включает структуры, описывающие факты и правила в соответствии с продукционной моделью. При создании структур используются трансформационные операторы, с помощью которых интерпретируются отношения между узлами модели динамики.

4. Создание программных кодов (спецификаций) прототипов баз знаний.

В данном случае трансформационные операторы представляют собой генераторы, которые для структур базы знаний синтезируют конструкции на определенном языке представления знаний, например CLIPS.

5. Тестирование и интеграция.

Данные операции осуществляются либо в условиях целевой программной платформы, либо в рамках инструментария технологии, например PKBD.

В данной работе предлагается расширить технологию за счет использования диаграмм переходов состояний [22], предназначенных для моделирования поведения систем в зависимости от их состояний и внешних воздействий. Каждая диаграмма – это граф, в котором узлы представляют собой состояние системы, связанное с длительным выполнением какой-либо деятельности. Дуги (переходы) – события или действия, приводящие к изменению состояния, могут описываться набором свойств. Диаграмма переходов состояний используется при проектировании и анализе сложных систем, для которых необходимо построить некоторую последовательность событий или работ.

С учетом предлагаемого расширения изменяются подвергаются этапы 2–4, в частности, построение моделей, описывающих последовательность и динамику процессов, осуществляется при помощи диаграмм переходов состояний. Создание базы знаний (фактов и правил) происходит интерпретацией отношений между узлами с учетом свойств дуг (переходов). Создаваемые программные коды представляют собой таблицы решений в формате CSV. Тестирование осуществляется в рамках целевой программной платформы iDSS.Desktop.

Для каждого из этапов предлагается использование определенного набора инструментальных средств (рис. 2), которые были апробированы при анализе документации по отдельным системам самолетов Sukhoi Superjet 100 (RRJ-95) и Cessna 182T (системы онтологического моделирования Protegé, системы визуального моделирования знаний KMS, универсального редактора декларативных баз знаний со встроенным интерпретатором и набором модулей-конвертеров концептуальных моделей PKBD [21], проблемно-ориентированной программы-оболочки iDSS.Desktop).



Рис. 2

В работе [15] была рассмотрена цепочка Protegé – KMS – PKBD, предполагающая использование онтологических моделей, деревьев событий и встроенного интерпретатора PKBD. В данной работе подробнее рассматривается цепочка Protegé – KMS – iDSS.Desktop, что предполагает использование диаграмм переходов состояний и таблиц решений.

Согласно рассмотренной последовательности (см. рис. 2), на этапе 1 выполнялось построение модели предметной области, включающее определение основных понятий и отношений. В качестве исходных данных использовались онтологические модели в области диагностики [23–25], технические руководства по эксплуатации, поиску и устранению неисправностей, а также информация об отказах в форме карт учета неисправностей авиационной техники. В рамках данной работы использовалась уже готовая модель предметной области, построенная ранее [15].

На рис. 3 представлен фрагмент диаграммы классов UML, описывающий основные классы.

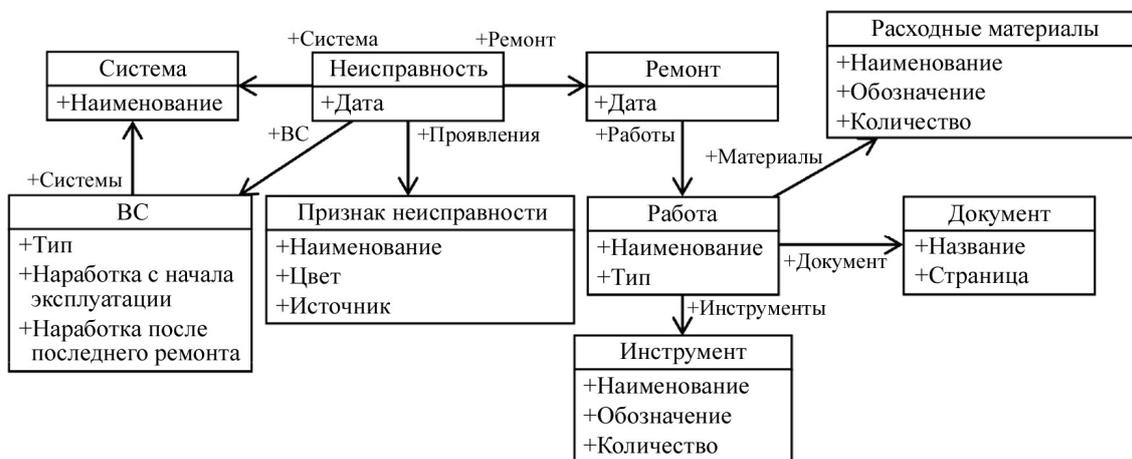


Рис. 3

Решение задач этапа 2 осуществлялось в системе моделирования знаний KMS, которая интегрирована с Protégé «по данным» в части импорта базовых понятий. Таким образом, после переноса информации о предметной области в KMS было осуществлено создание диаграмм переходов состояний, соответствующих определенным цепочкам операций (работ) по поиску и устранению неисправностей. При этом дополнительно задавались параметры и ограничения на их значения как для состояний, так и для переходов между ними.

Диаграммы создавались для системы кондиционирования воздуха самолета Cessna 182T и систем электроснабжения, управления и кондиционирования воздуха самолета Sukhoi Superjet 100 (RRJ-95) (рис. 4).

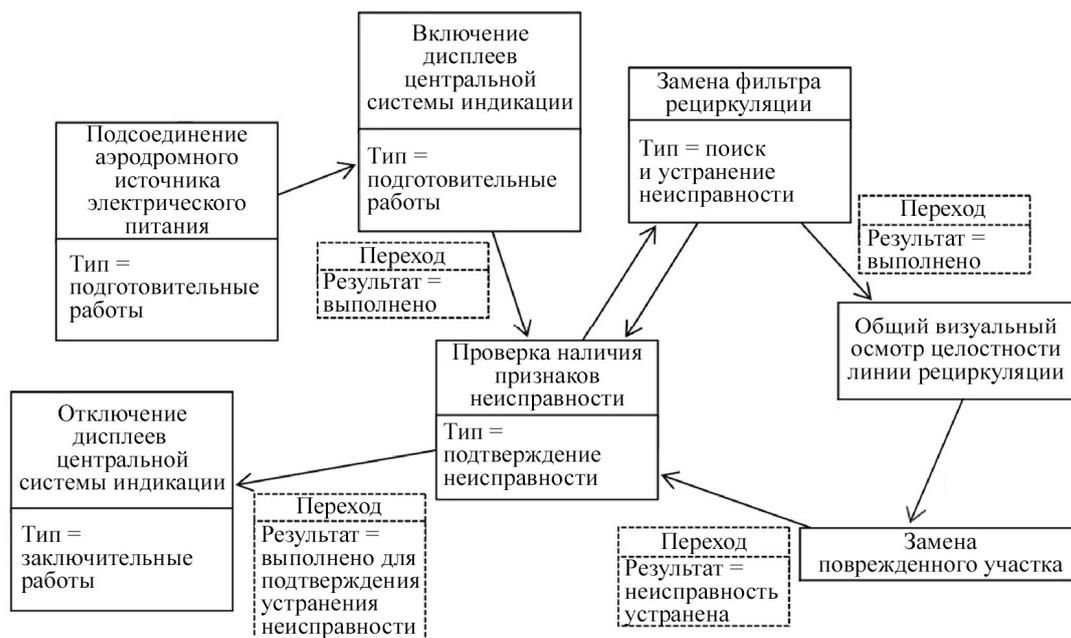


Рис. 4

Для каждой комбинации проявлений неисправностей была разработана отдельная диаграмма (табл. 1).

Таблица 1

Самолет	Система	Количество диаграмм	Количество правил определения последовательности операций (работа – работа)	Отношение количества правил к количеству диаграмм	Количество правил определения причин (проявление – система)
Sukhoi Superjet 100 (RRJ-95)	Электроснабжение	22	828	37,6	173
	Кондиционирование	24	687	28,6	178
Cessna 182T	Кондиционирование	11	52	4,7	23

Решение задач этапа 3, связанных с генерацией структур баз знаний, а также кодов и спецификаций, максимально автоматизировано в KMS. На этом этапе построенные диаграммы экспортируются в один из доступных форматов (XML или CSV), что подразумевает различные способы использования результатов моделирования. Второй способ, который и предлагается в данной работе, позволяет экспортировать данные в виде таблиц решений [26] для их дальнейшей отладки в iDSS.Desktop / «АвиаТех-Пом.Терминал». В табл. 2 приведен фрагмент полученной таблицы решений для системы кондиционирования самолета Cessna 182T. Необходимо отметить, что визуальное моделирование применялось только для построения правил, описывающих зависимости между операциями или работами (работа – работа), поскольку именно в этом случае требовалось построение разветвленных диаграмм переходов состояний. Количественные результаты генерации правил приведены в табл. 1.

Таблица 2

Признак неисправности-1	Признак неисправности-2	Работа			Следующая работа
		Документация	Текущая работа	Результат выполнения работы	
Недостаточный расход хладагента	Недостаточное охлаждение. На смотровом стекле появляются пузырьки	21-50-00-103	Начальная	Выполнено	Проверка герметичности
			Проверка герметичности		Устранение утечки
			Устранение утечки		Зарядка системы

Решение задач тестирования (отладки) и интеграции (этап 4) осуществлялось в среде настольного приложения «Терминал», благодаря которой возможно не только их редактирование при помощи специализированного редактора таблиц решений, но и отладки посредством оригинальной реализации алгоритма сопоставления с образцом.

Была использована следующая последовательность действий при работе с приложением «Терминал» при выполнении действий этапа 4.

1. Предобработка экспортированных по результатам этапа 3 данных, включающая объединение всех результирующих таблиц в один файл с формированием единой таблицы для определенной системы самолета и согласование наименований (переименование) заголовков столбцов.

2. Размещение полученных таблиц решений в служебных директориях приложения «Терминал».

3. Унификация, проверка и редактирование таблиц решений в специализированном редакторе.

4. Отладка отредактированных данных и знаний выполнением различных поисковых запросов на основном экране приложения, в том числе указанием различных комбинаций внешних проявлений неисправностей, а также пошаговой проверкой разных последовательностей (цепочек) операций/работ. Критерием корректности данных являлось полное соответствие полученных результатов нормативным документам.

Применение подходов, предполагающих использование визуальных общепринятых моделей, в частности диаграмм переходов состояний, а также автоматической кодогенерации позволяет значи-

тельно расширить круг пользователей программных систем и более полно вовлекать непрограммируемых специалистов в процесс адаптации/настройки программного обеспечения. В работе было продемонстрирована принципиальная возможность решения задачи построения прототипов баз знаний для задач авиационной диагностики на основе предложенного подхода.

Созданные базы знаний (см. табл. 1) включают логические правила, описывающие зависимости двух типов:

1) между проявлениями неисправностей и причинами, где проявления – это либо сигналы Crew Alerting System, либо внешние признаки, а причины – возможные системы-претенденты на отказ (правила типа проявление – система);

2) между операциями по поиску и устранению неисправностей, где каждой операции соответствует определенный тип и/или раздел документации (правила типа работа – работа).

Если правила первого типа не предполагают многошаговых рассуждений и являются «атомарными», то правила второго типа предполагают построение цепочек (для самолета Cessna 182T) или разветвленных ориентированных графов (для самолета Sukhoi Superjet 100 (RRJ-95)). Именно в случае построения разветвленных ориентированных графов применялось визуальное моделирование (см. рис. 4) для корректного формирования логических связей между работами. В зависимости от системы самолета по одной диаграмме в среднем формировалось до 37 правил.

Основываясь на результатах использования диаграмм переходов состояний для создания прототипов баз знаний, можно сделать следующие выводы.

Эффективность использования диаграмм по критерию времени (а также большинства систем графических обозначений) определяется как сложностью описываемых структур, так и объемом базы знаний. Обычно она ниже, чем при использовании текстовых и табличных форм представления, в частности таблиц решений.

В сравнении с деревьями событий [15] использование диаграмм переходов состояний более предпочтительно, поскольку благодаря циклическим связям уменьшается количество визуальных объектов на диаграммах. В дальнейшем необходимо обеспечить автоматическое взаимное преобразование данных моделей.

Использование диаграмм переходов состояний позволяет не только верифицировать таблицы решений, визуально представляя цепочки операций, но и технические руководства, в которых присутствуют ошибки, обусловленные человеческим фактором.

Таким образом, результаты, полученные в данной работе, позволяют говорить о перспективности подхода к построению прототипов декларативных баз знаний для диагностирования неисправностей технических систем. Для отработки подхода использовалась система кондиционирования воздуха самолета Cessna 182T и системы электроснабжения, управления и кондиционирования воздуха самолета Sukhoi Superjet 100 (RRJ-95). В дальнейшем планируется рассмотреть другие системы и типы воздушных судов, а также возможность перенесения отдельных принципов модельно-ориентированной разработки на область виртуальных ассистентов в контексте решения задач авиационной диагностики и технического обслуживания.

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации по проекту № 121030500071-2.

СПИСОК ЛИТЕРАТУРЫ

1. Зрячев С.А., Ларин С.Н. Разработка базы знаний послепродажного обслуживания авиационной техники // Изв. СамНЦ РАН. 2022. № 5. С. 48–53.
2. Перфильев О.В., Рыжиков С.Г., Должиков В.А. Интеллектуальная система поиска неисправности на самолете // Изв. СамНЦ РАН. 2018. № 4. С. 326–331.
3. Сухих Н.Н., Рукавишников В.Л. Экспертные системы – средство информационной поддержки принятия решений экипажем самолета // Изв. вузов. Авиационная техника. 2022. № 2. С. 19–25.

4. *Kotlov Y. et al.* Towards an Intelligent Decision Support System for Aircraft Troubleshooting // Lecture Notes in Mechanical Engineering. Proceedings of 10th International Conference on Recent Advances in Civil Aviation / Ed. O.A. Gorbachev, Xiaoguang Gao, Bo Li. Springer, 2023. P. 77–91.
5. *Бельский А.Б., Есипович О.А., Сафоклов Б.Б.* Прогнозирование технического состояния агрегатов винтокрылых летательных аппаратов на основе нейросетевого моделирования // Изв. вузов. Авиационная техника. 2023. № 4. С. 4–14.
6. *Баутин А.А., Свирский Ю.А.* Применение нейросетевых технологий в задачах мониторинга состояния критических мест конструкции транспортных самолетов // Вестник МАИ. 2020. № 4. С. 81–91.
7. *Долгов О.С., Сафоклов Б.Б., Смагин А.А.* Диагностика и прогнозирование ресурса взлетно-посадочных устройств с использованием искусственных нейронных сетей // Изв. вузов. Авиационная техника. 2022. № 2. С. 3–10.
8. *Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И.* Инженерия знаний. Модели и методы. СПб.: Лань, 2016. 324 с.
9. *Schmidhuber J.* Deep Learning in Neural Networks: An Overview // Neural Networks. 2015. Vol. 61. P. 85–117.
10. *Matsuo Y. et al.* Deep Learning, Reinforcement Learning, and World Models // Neural Networks. 2022. Vol. 152. P. 267–275.
11. *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.
12. *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.
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. *Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice / Eds. L.G. Cretu, D. Florin.* Apple Academic Press, 2014. 350 p.
15. *Котлов Ю.В., Николайчук О.А., Юрин А.Ю.* Использование деревьев событий при автоматизации и интеллектуализации диагностирования и ремонта авиационной техники // Изв. вузов. Авиационная техника. 2023. № 4. С. 222–228.
16. *Дородных Н.О., Юрин А.Ю.* Технология создания продукционных экспертных систем на основе модельных трансформаций. Новосибирск: СО РАН, 2019. 144 с.
17. *Дородных Н.О., Юрин А.Ю.* Программная реализация алгоритмов для создания прототипов баз знаний на основе визуального моделирования и трансформаций // Программные продукты и системы. 2024. № 3. С. 324–333.
18. *Юрин А.Ю., Котлов Ю.В.* АвиаТехПом: Состояние и перспективы // Crede Experto: транспорт, общество, образование, язык. 2024. № 1. С. 146–156.
19. *Юрин А.Ю., Утехин С.Ю.* Применение программно-алгоритмического обеспечения «АвиаТехПом» при решении задач ремонта и технического обслуживания воздушных судов // Crede Experto: транспорт, общество, образование, язык. 2024. № 4. С. 116–126.
20. *Yurin A.Yu., Kotlov Y.V., Popov V.M.* The Conception of an Intelligent System for Troubleshooting an Aircraft [Electronical Resource]. URL: <https://ceur-ws.org/Vol-2984/paper6.pdf> (дата обращения: 5.02.2025).
21. *Yurin A.Yu., Dorodnykh N.O.* Personal Knowledge Base Designer: Software for Expert Systems Prototyping // SoftwareX. 2020. Vol. 11. Article № 100411.
22. *Хопкрофт Дж., Мотвани Р., Ульман Дж.* Введение в теорию автоматов, языков и вычислений. М.: Вильямс, 2008. 528 с.
23. *Берман А.Ф.* Деградация механических систем. Новосибирск: Наука, 1998. 320.
24. *Грибова В.В., Шалфеева Е.А.* Онтология диагностики процессов // Онтология проектирования. 2019. № 4. С. 449–461.
25. *Дородных Н.О., Николайчук О.А., Юрин А.Ю.* Использование онтологических шаблонов содержания при построении баз знаний для технического обслуживания и ремонта авиационной техники // Онтология проектирования. 2022. № 2. С. 158–171.
26. *Seagle J.P., Duchessi P.* Acquiring Expert Rules with the Aid of Decision Tables // European Journal of Operational Research. 1995. Vol. 84. Iss. 1. P. 150–162.

Поступила в редакцию 5.03.25
 После доработки 14.03.25
 Принята к публикации 14.03.25

An Approach to Building Knowledge Bases for Aviation Diagnostics Tasks

N.O. DORODNYKH¹, O.A. NIKOLAICHUK¹, YU.V. KOTLOV², AND A.YU. YURIN³

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

² Moscow State Technical University of Civil Aviation, Irkutsk Branch, Irkutsk

³ INRTU, Irkutsk

The paper describes the extension of the author's approach to building prototypes of declarative knowledge bases of classical expert systems based on model transformations through the use of state transition diagrams and decision tables. These methods describe in the form of logical rules the information about relationships between the external signs of malfunctions and the causes, as well as between troubleshooting operations. The approach has been tested during the adaptation of the AviaTekhPom.Terminal application for the Cessna 182T air conditioning system and Sukhoi Superjet 100 (RRJ-95) power supply, control, and air conditioning systems. The estimation of results showed the expediency of using the described approach for modeling the branching scenarios of operations, verifying decision tables, and technical manuals.

Aircraft diagnostics, automation, knowledge base development, model transformation, visual models, state transition diagrams, decision tables, code generation