

## Технология создания основанных на знаниях систем для исследования динамики состояния технических объектов

А. Ф. БЕРМАН, Н. О. ДОРОДНЫХ, О. А. НИКОЛАЙЧУК\*, А. Ю. ЮРИН  
Институт динамики систем и теории управления им. В.М. Матросова СО РАН, 664033,  
Иркутск, Россия

\*Контактный автор: Николайчук Ольга Анатольевна, e-mail: nikoly@icc.ru

Поступила 20 мая 2023 г., принята в печать 19 июня 2023 г.

Целью данной работы является описание технологии создания систем, основанных на знаниях, для решения задач исследования динамики их технического состояния с использованием модельно-ориентированного подхода и онтологического моделирования. Под исследованием динамики технического состояния авторы понимают решение задач генезиса, идентификации и прогнозирования. Предлагаемая технология реализует подход MDA (model-driven architecture). Новизной технологии является проблемно-ориентированная специализация и обеспечение в ее рамках возможности использования различных моделей представления знаний и методов их обработки в зависимости от имеющихся в распоряжении исследователя данных и знаний. Апробация подхода выполнена для решения задач экспертизы промышленной безопасности и поддержки принятия решений при диагностировании систем самолета.

*Ключевые слова:* динамика технического состояния, модельно-ориентированный подход, системы, основанные на знаниях, онтология, прецедентный подход, продукции, таблицы решений.

*Цитирование:* Берман А.Ф., Дородных Н.О., Николайчук О.А., Юрин А.Ю. Технология создания основанных на знаниях систем для исследования динамики состояния технических объектов. Вычислительные технологии. 2023; 28(4):94–108. DOI:10.25743/ICT.2023.28.4.008.

### Введение

Обоснование техногенной безопасности создаваемых и эксплуатируемых опасных промышленных объектов остается актуальной проблемой, решение которой требует обработки огромных объемов данных и знаний различных научных, технических и научно-технических дисциплин [1, 2]. Данные и знания имеют разнородную структуру и различную степень формализации, что обуславливает необходимость привлечения интеллектуальных методов обработки информации, обеспечивающих взаимодействие специалистов различных квалификаций.

Применение методов искусственного интеллекта (ИИ) позволит:

- сохранить корпоративные знания и опыт в виде баз знаний;
- использовать сохраненный опыт при обучении и экспертной поддержке менее квалифицированных специалистов;

- повысить эффективность и качество принятия решений по совершенствованию технологических процессов, систем контроля качества изготовления продукции и диагностирования ее технического состояния в процессе эксплуатации;
- обеспечить прогнозирование возможных повреждений и предельных состояний на основе особенностей технологических процессов, формирование рекомендаций по снижению скорости деградации технического состояния эксплуатируемых опасных промышленных объектов и обеспечению их технической безопасности.

В свою очередь для повышения эффективности процессов обработки знаний перспективным является более полное вовлечение в процесс создания систем ИИ конечных пользователей (end users) с передачей им отдельных функций, которые исторически реализовывались программистами. Подходы подобной направленности объединены в направление, известное как end-user development (EUD) [3], основная идея которого — предоставить возможность конечному пользователю самому создавать и настраивать приложение. В рамках этого направления многообещающими способами спецификации предметной области и бизнес-логики приложений является использование [4] принципов визуального программирования, предметно- и проблемно-ориентированных языков, а также модельных трансформаций с целью автоматизированной генерации программных кодов и спецификаций.

Модельно-ориентированный подход (model-driven architecture, MDA) [5] определяет процесс создания программных систем как последовательность этапов, включающих построение определенных моделей и их трансформации:

- создание вычислительно-независимых моделей (computation independent model, CIM), обеспечивающих формулировку ключевых абстракций решаемой задачи;
- создание проблемно-ориентированных платформо-независимых моделей (platform independent model, PIM), учитывающих специфику решаемой задачи;
- создание платформо-зависимых моделей (platform specific model, PSM), ориентированных на определенную программную платформу функционирования программного приложения или язык программирования;
- генерация кодов и спецификаций программного приложения для определенной платформы или языка.

Целью настоящей работы является описание технологии создания систем, основанных на знаниях, для решения задач исследования динамики технического состояния, на базе модельно-ориентированного подхода и онтологического моделирования. Под исследованием динамики технического состояния авторы понимают решение задач его генезиса, идентификации и прогнозирования.

## 1. Предлагаемая технология

Для создания систем, основанных на знаниях, предлагается развить авторскую специализацию модельно-ориентированного подхода [6]. Новизна этого подхода — в его проблемно-ориентированной специализации и обеспечении возможности в рамках одной технологии использовать различные модели представления знаний и методы их обработки в зависимости от имеющихся в распоряжении исследователя данных и знаний.

С методологической точки зрения предлагаемый процесс разработки систем, основанных на знаниях, представлен последовательностью этапов и в обобщенном формальном виде может быть описан выражением

$$T : Ont \rightarrow Code, \quad (1)$$

где *Ont* — онтологическая модель; *Code* — программный код; *T* — трансформация моделей.

Далее опишем более детально каждый из этапов специализированного подхода (рис. 1).

*Этап 1. Построение проблемно-ориентированной модели предметной области.* Выходная информация этапа:

- онтология методологии решения задачи;
- онтология объекта решения задачи;
- онтология задачи, включая описание методов и средств ее решения.

Теоретической основой разрабатываемых онтологических моделей является предложенная авторами причинно-следственная модель динамики состояния технического объекта [7, 8]. Модель базируется на структурной иерархии исследуемого объекта: деталь (Д) — сборочная единица (СЕ) — механическая система (МС) — техническая система (ТС) — сложный технологический комплекс (СТК). Отношение часть — целое

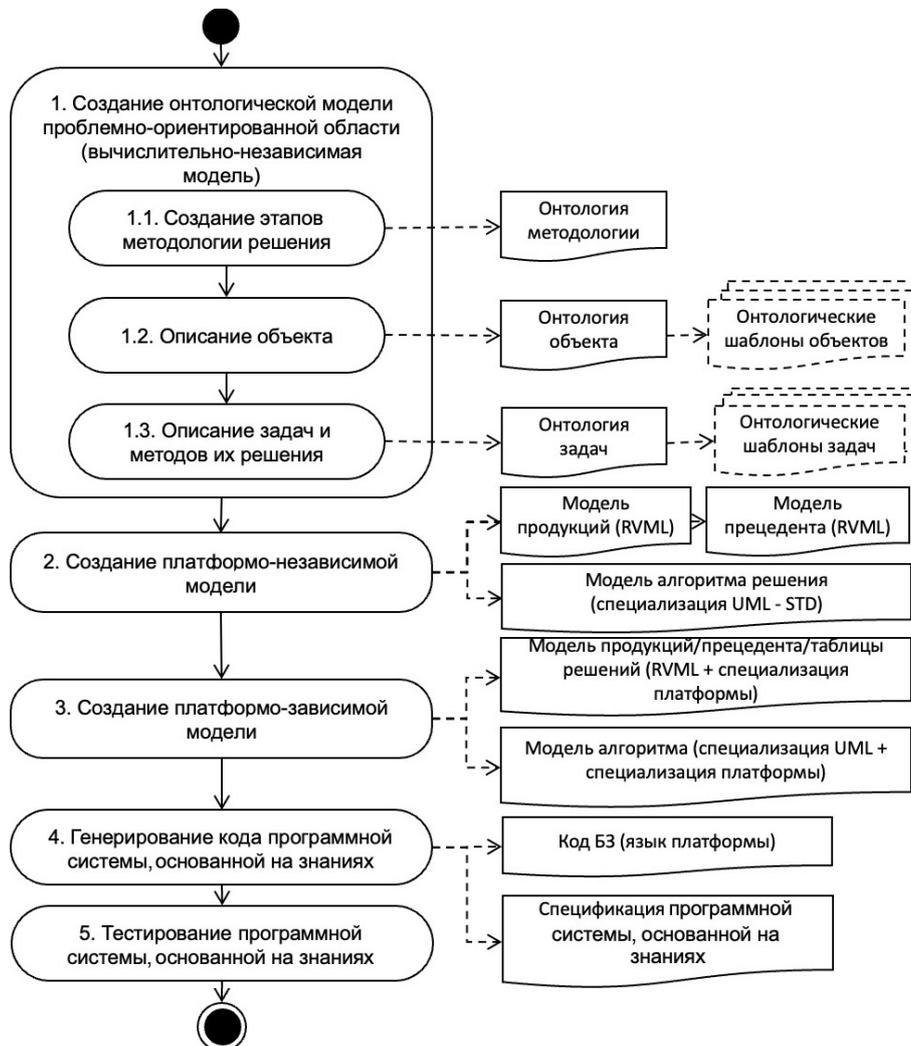


Рис. 1. Основные этапы и модели предлагаемого подхода

Fig. 1. The main stages and models of the proposed approach

между элементами структуры определяет причинно-следственные отношения между их техническими состояниями: деталь является составной частью сборочной единицы, одновременно деталь обуславливает техническое состояние сборочной единицы, а сборочная единица, в свою очередь, — техническое состояние МС и т. д.

В процессе функционирования СТК и его структурные элементы могут находиться в одном из двух типов технических состояний: безопасном и опасном. Опасное состояние связано с достижением предельных состояний и отказами структурных элементов, входящих в СТК. Динамика изменения состояний СТК непрерывна и дискретна в зависимости от вида и значения некоторых параметров, описывающих процесс. Тогда пространство состояний представлено последовательностью условно-дискретных состояний (или классов состояний): исходная дефектность (Деф), поврежденность (П), разрушение (Р), отказ (О) — для механической системы; отказ (О), аварийная ситуация (АС), авария (А), чрезвычайная ситуация (ЧС) — для сложного технологического комплекса.

Перечисленные условно-дискретные состояния соответствуют стадиям нежелательного (деградационного) процесса как на уровне технического объекта в целом, так и на уровнях его структурных элементов. Приведем причинно-следственную цепочку состояний, например, для механической системы:

$$S_{\text{Деф}}^{\text{MC}} \rightarrow S_{\text{П}}^{\text{MC}} \rightarrow S_{\text{Р}}^{\text{MC}} \rightarrow S_{\text{О}}^{\text{MC}},$$

где  $S_{\text{Деф}}^{\text{MC}}$  — дефектное состояние объекта;  $S_{\text{П}}^{\text{MC}}$  — состояние повреждения;  $S_{\text{Р}}^{\text{MC}}$  — состояние разрушения;  $S_{\text{О}}^{\text{MC}}$  — состояние отказа. Тогда с учетом иерархической структуры рассматриваемого объекта переходы состояний будут представлены в виде матрицы состояний, где каждое состояние описывается набором параметров:

$$\begin{array}{ccccccc} S_{\text{Деф}}^{\text{MC}} & \rightarrow & S_{\text{П}}^{\text{MC}} & \rightarrow & S_{\text{Р}}^{\text{MC}} & \rightarrow & S_{\text{О}}^{\text{MC}} \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ S_{\text{Деф}}^{\text{CE}} & \rightarrow & S_{\text{П}}^{\text{CE}} & \rightarrow & S_{\text{Р}}^{\text{CE}} & \rightarrow & S_{\text{О}}^{\text{CE}} \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ S_{\text{Деф}}^{\text{D}} & \rightarrow & S_{\text{П}}^{\text{D}} & \rightarrow & S_{\text{Р}}^{\text{D}} & \rightarrow & S_{\text{О}}^{\text{D}} \end{array} \quad (2)$$

Причинно-следственная цепочка состояний описывает сценарий развития некоторого нежелательного (деградационного) процесса или их совокупности.

Структуру множества параметров, описывающих процессы, также можно детализировать:

$$DP = \{DP_k\}, \quad DP_k = (Mech_k, Kin_k), \quad Mech_k \rightarrow Kin_k, \quad (3)$$

где  $DP$  — деградационный процесс;  $DP_k$  —  $k$ -я стадия этого процесса;  $Mech_k$  и  $Kin_k$  — его механизм и кинетика. При этом

$$Mech_k = (ObjPr, F, TE, L, CF, NF), \quad Kin_k = \{Ev_{ki}\}, \quad Ev_{ki} \rightarrow Ev_{ki+1}, \quad (4)$$

где  $ObjPr$  — свойства технического объекта;  $F$  — воздействующие факторы;  $TE$  — среда;  $L$  — нагрузки;  $CF$  — управляющие воздействия;  $NF$  — природные воздействия;  $Ev_{ki}$  — события кинетики;  $\rightarrow$  — причинно-следственные отношения.

Описание знаний на основе приведенных структур позволяет решать задачи и генезиса, и прогнозирования технического состояния.

Дальнейшим отражением теоретической модели (2)–(4) является онтология. Онтология в общем случае — это метамодель описания знаний, которая представляет собой описание понятий и отношений между ними:

$$M^{Ont} = (C, Pr, R, I),$$

где  $C$  — множество понятий;  $Pr$  — множество свойств понятий;  $R$  — отношения между понятиями;  $I$  — множество аксиом.

Предлагаемая модель онтологии использует понятия онтологий верхнего уровня, предметной и проблемной областей [9–11]. Для описания аспекта междисциплинарности задач предлагается использовать: онтологию дисциплин, где отражена информация, относящаяся к рассматриваемой дисциплине, ее понятия, закономерности, задачи; модель методологии, которая описывает системные аспекты решения задач; модели проблемной области, охватывающие знания о способах решения разнообразных типов задач. Для эффективного решения каждой задачи предусмотрено адекватное представление данных и знаний, а также алгоритмов, реализующих решение. Подобная модель онтологии позволяет описать иерархию задач, методы их решения и программное обеспечение (“решатели”), реализующее данные методы, разделяя информацию на предметно-зависимую и предметно-независимую, что обеспечивает требуемые свойства адаптивности программной системы, а также возможности взаимодействия специалистов и системы в терминах предметных областей.

Для единообразного представления информации из различных предметных областей необходимо разрабатывать шаблоны (паттерны) онтологического проектирования (ontology design pattern, ODP) [12], которые могли бы заполняться пользователями-экспертами для автоматизации процессов решения задач. Примерами таких шаблонов являются онтологические описания задачи диагностирования технического объекта [13].

Заполнение онтологических шаблонов необходимо автоматизировать на основе анализа концептуальных моделей, широко используемых в различных предметных областях и создаваемых непрограммирующими пользователями. Такие модели имеют общесистемную направленность, ориентированы на систематизацию знаний или поддержку принятия решений и являются удобным и понятным для специалиста-предметника способом представления знаний (например, концептуальные карты, диаграммы Венна и Исикавы, семантические модели, “деревья отказов” (fault tree analysis) и “деревья событий” (event tree analysis), применяемые в области анализа отказов и риска технических систем [14] и т. д.) (рис. 2).

Полученные модели на данном этапе рассматриваются как вычислительно-независимые и могут быть представлены в виде OWL-онтологии (рис. 3), UML-моделей (в частности, в виде UML-диаграмм классов) или карт памяти.

*Этап 2. Построение платформо-независимых моделей.* Выходная информация этого этапа:

- описание знаний (причинно-следственных зависимостей между понятиями предметной области, описывающих ее закономерности);
- описание алгоритмов решения задачи.

На основе онтологии формируются гетерогенные модели знаний. Гетерогенность может быть обусловлена различными моделями представления знаний, как хорошо структурированными, так и слабоструктурированными, также возможно их гибридное представление.

Приведем фрагменты моделей знаний, описывающих динамику технического состояния, созданные на основе прецедентного, продукционного подходов и таблиц решений.

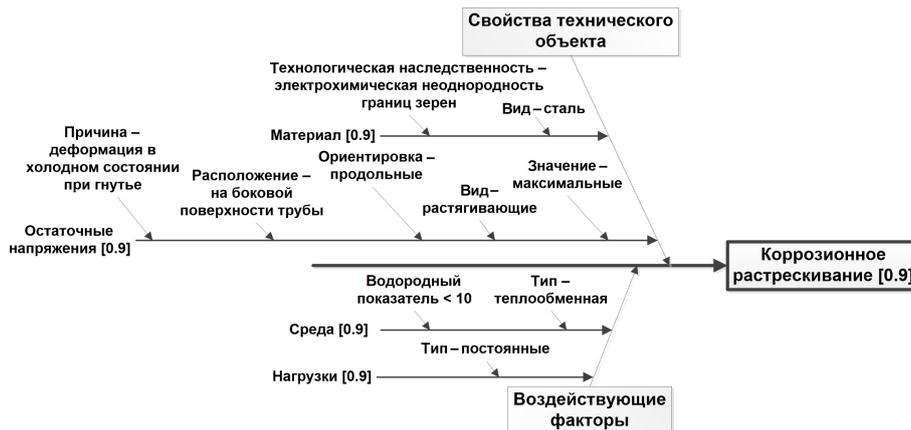


Рис. 2. Пример фрагмента диаграммы Исикавы: механизм деградационного процесса “коррозионное растрескивание”, стадия повреждения [15]

Fig. 2. An example of a fragment of the Fishbone diagram: the mechanism of the “corrosion cracking” degradation process, the stage of damaging [15]

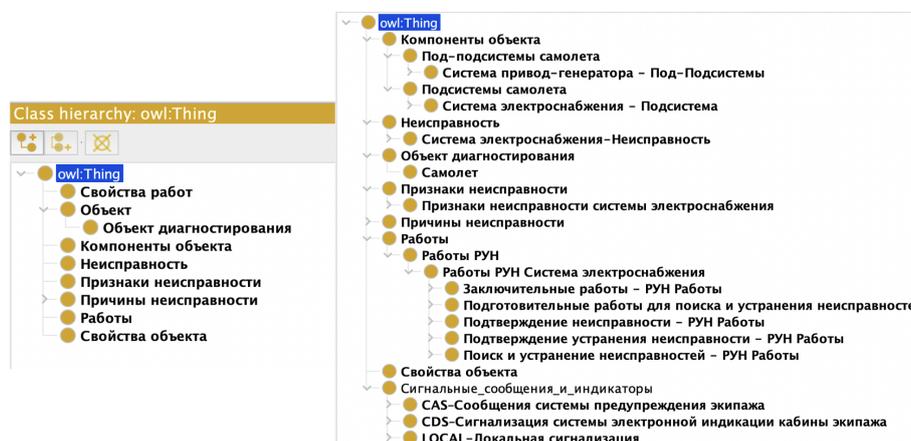


Рис. 3. Пример онтологического шаблона объекта диагностирования и фрагмент онтологии системы электроснабжения самолета

Fig. 3. An example of the ontological pattern of the diagnostic object and a fragment of the ontology of the aircraft power supply system

Прецедентный подход (case-based reasoning, CBR) [16, 17] — это методология принятия решений, которая позволяет повторно использовать и адаптировать (при необходимости) ранее полученные решения похожих проблем на основе принципа принятия решений “по аналогии”. Процесс решения проблем на основе прецедентов состоит из ряда этапов: извлечение; повторное использование, включая адаптацию; проверка полученного решения; сохранение. На каждом из этапов используются определенные методы.

Модель прецедента

$$M^{CBR} = (C^{CBR}, Pr, T, V). \tag{5}$$

Здесь  $C^{CBR} = \{c_1, \dots, c_N\}$ ,  $C^{CBR} \subset C$ , — объекты некоторой предметной области (база прецедентов);  $Pr = \{p_1, \dots, p_M\}$  — свойства этих объектов;  $T = \{t_1, \dots, t_K\}$  — типы значений, в частности количественное, качественное, интервальное;  $V$  — множество значений свойств определенного типа. Причем  $M^{CBR} : Problem^{CBR} \rightarrow Decision^{CBR}$ , где  $Problem^{CBR} = (c^*(p_1, \dots, p_M))$ ,  $c^* \notin C^{CBR}$ , — описание нового объекта или новой те-

кущей ситуации (идентифицирующая часть);  $Decision^{CBB}$  — описание решения (обучающая часть), при этом  $Decision^{CBB} = \{d_1, \dots, d_R\}$ ,  $d_r = (c_r, s_r)$ ,  $c_r \in C^{CBB}$ ,  $s_r \in [0; 1]$ , — описание решения задачи в форме извлеченных из базы прецедентов решений с оценкой их близости  $s_r$  к текущей ситуации  $c^*$ .

Существует множество методов для извлечения (поиска) прецедентов [18]: ближайшего соседа, деревьев решений и др. Наиболее популярен метод ближайшего соседа, основанный на оценке сходства (подобия), вычисляемой с помощью различных метрик, в частности евклидовой, метрики городских кварталов и т. д. Прецедентные модели описывают последовательности состояний согласно (1), что позволяет прогнозировать техническое состояние (табл. 1), получать описание отказов или аварий и принятые решения для их устранения и предотвращения, описание свойств материалов и их возможные эксплуатационные свойства для осуществления поддержки принятия решений при обосновании выбора материалов, например при изготовлении и ремонте нефтехимического оборудования.

Для моделирования динамики технического состояния и прогнозирования аварий предлагается применять оригинальный алгоритм последовательного поиска аналогов

Т а б л и ц а 1. Модель прецедента для идентификации технического состояния нефтехимического оборудования

Table 1. A case model for identifying the technical state of petrochemical equipment

Свойство	Значение
Описание проблемы	
Структурная принадлежность	<b>Завод:</b> Уфимский завод синтетического спирта, <b>Производство:</b> ПЭВД-160, <b>Линия:</b> линия-6, <b>Изделие:</b> трубопровод обвязки компрессора 2-го каскада, <b>Сб. единица:</b> фланцевое соединение 1, <b>Деталь:</b> тройник $D_y \times D_y = 24 \times 10$
Наработка	<b>В часах</b> — 47 232, <b>в месяцах</b> — 65.6, <b>в годах</b> — 5.5
Источник информации	Отчет по теме № 81-79 от 06.04.1974 ИркутскНИИхиммаш
Повреждения	Трещина (трещины). <b>Параметры:</b> место зарождения — на наружной боковой поверхности. Ориентация — перпендикулярно напряжениям. Особенности места зарождения — резьбовая. Тип образования — одиночное
Разрушение	Сквозная трещина. <b>Параметры:</b> не определены
Внешние проявления отказа	Утечка среды в окружающее пространство
Решение	
Деградационные процессы	Многоцикловая усталость
Причина	Конструкционная, эксплуатационная
Происхождение	Несовершенство правил и норм расчета и конструирования (не учтено влияние вибрации). Несовершенство правил эксплуатации, технического обслуживания и ремонта (отсутствие эффективных методов неразрушающего контроля технического состояния элементов, в том числе без разборки оборудования; недопустимая вибрация)
Рекомендуемые мероприятия	Модернизация конструкции с целью снижения уровня вибрации. Модернизация конструкции с целью обеспечения эффективного контроля технического состояния. Уточнение периодичности контроля технического состояния

в пространствах прецедентов, отражающих классы состояний (2) [19]. Результатами такого похода является дерево отказов или дерево событий (рис. 4).

Продукционная модель основана на представлении знаний в виде логических правил:

$$\text{ЕСЛИ } c_{i1} \circ c_{i2} \circ \dots \circ c_{iK} \text{ ТО } c_{iK+1}, c_{ik} \in C,$$

где  $\circ$  — логическая операция. Машина вывода (интерпретатор) осуществляет логический вывод на основе правил, в результате которого выявляются новые факты о текущей ситуации.

Теоретическая модель (2)–(4) позволяет сформировать шаблоны правил, описывающих классы состояний и причины их возникновения, на основе которых эксперты могут формировать конкретные правила для конкретных технических объектов. Например, представим шаблон правила для определения механизма деградиционного (нежелательного) процесса.

**ЕСЛИ** Свойство объекта **И** Среда **И** Нагрузки **И** Управляющее воздействие **И** Ошибки оператора **И** Природные воздействия **ТО** Механизм  $j$ -й стадии развития  $i$ -го нежелательного (деградационного) процесса. На основе этого шаблона получено следующее конкретное правило:

**ЕСЛИ** Свойство объекта (материал детали — низколегированная сталь, остаточные макронапряжения — да) **И** Среда (вид — активная) **И** (Нагрузки (тип — механические напряжения, вид — растягивающие, величина — выше нормы) **ИЛИ** Нагрузки (тип — термические напряжения)) **ТО** Механизм деградиционного процесса (стадия — повреждение, наименование — механизм коррозионного растрескивания).

Система продукций, описывающих механизмы и кинетику процессов (3), (4), позволяет сгенерировать дерево, представляющее возможную динамику технического состояния (рис. 5).

Для формирования множества прецедентов и продукций следует эффективно использовать интуитивно понятный способ описания действий, выполняемых при раз-

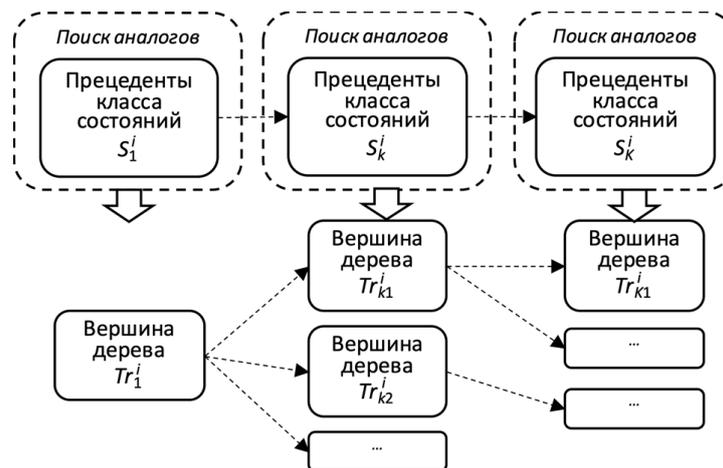


Рис. 4. Пример схемы генерирования дерева событий на основе последовательного поиска аналогов в пространствах прецедентов каждого класса состояний ( $S_k^i$  —  $k$ -й класс состояний  $i$ -го элемента,  $Tr_k^i$  — вершина дерева событий, соответствующая  $k$ -му классу состояний  $i$ -го элемента)

Fig. 4. Example of a scheme for generating an event tree based on a sequential search for cases in the case spaces of each class of states ( $S_k^i$  —  $k^{\text{th}}$  class of states of the  $i^{\text{th}}$  element,  $Tr_k^i$  — the vertex of the event tree corresponding to the  $k^{\text{th}}$  class of states of the  $i^{\text{th}}$  element)

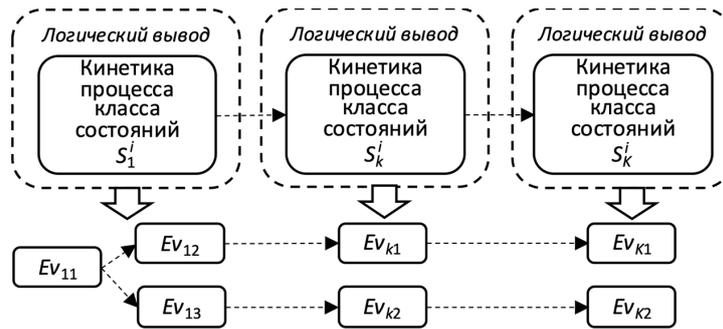


Рис. 5. Пример схемы генерирования дерева событий на основе последовательного логического вывода на продукциях, описывающих кинетику нежелательного процесса

Fig. 5. Example of a scheme for generating an event tree based on sequential logical inference on logical rules describing the kinetics of an undesirable process

личных комбинациях условий, в виде таблиц решений (decision tables) [13]:  $M_{DT} = (Conditions, Actions)$ , где  $M_{DT}$  — модель таблицы решений;  $Conditions, Actions$  — множества понятий предметной области, их свойств и ограничений на значения этих свойств:  $Conditions = \{c_{ic}\}$ ,  $Actions = \{c_{ia}\}$ ,  $c^{DT} = \{(name_i, type_i, valueConstraint_i)\}$ ,  $c_{ic}, c_{ia} \in C$ ,  $name_i \in Pr$ ,  $type_i, valueConstraint_i \in I$ ,  $Conditions \equiv Problem$ ,  $Actions \equiv Decision$ . Пример таблицы решений показан в табл. 2.

Перечисленные модели этого этапа являются результатом трансформации вычислительно независимой модели, в процессе преобразования которой осуществляется следующее: понятия онтологии объекта исследования, в частности технического объекта, преобразуются в шаблоны фактов и элементы представления знаний, а причинно-следственные связи — в логические правила, прецеденты или таблицы решений; RVMML (авторская нотация представления знаний) используется для визуализации и последующего уточнения элементов полученных моделей; понятия онтологий методологии и задач преобразуются в спецификацию для интерпретации/выполнения функциональности создаваемой программной системы.

*Этап 3. Построение платформи-зависимых моделей.* Количество моделей определяется числом платформ, для которых создается система, основанная на знаниях. В настоящее время спецификация ориентирована на платформы коллектива (PKBD, ADSkit) [6, 20]. Платформи-зависимые модели являются результатом автоматических преобразований платформи-независимой модели с помощью специальных инструментов с последующей модификацией конечным пользователем.

В рамках этого этапа конечный пользователь должен уточнить RVMML-модели правил и прецедентов с учетом особенностей определенного языка представления/программирования знаний (например, CLIPS), таких как: приоритеты правил, значения слотов “по умолчанию” и коэффициент уверенности (определенности).

Модели алгоритмов реализуются в рамках авторских платформ, поэтому трансформация моделей выполняется автоматически.

*Этап 4. Генерация программных кодов и спецификаций баз знаний и программной системы в целом.* В результате выполнения данного этапа пользователь получает:

- код базы знаний для определенного языка программирования (например, CLIPS);
- спецификации системы, основанной на знаниях, для интерпретатора платформы.

Для реализации предлагаемой технологии разработан комплекс взаимосвязанных средств автоматизации, интегрированных по данным через форматы файлов, содержащих коды и спецификации, в частности:

- CSV — текстовый формат представления табличных данных, используется для структурированного представления и хранения знаний в форме прецедентов и продукций;
- ЕКВ (external knowledge base) — XML-подобный формат, используется программным средством PKBD (personal knowledge base designer) для представления и хранения баз знаний;
- STD (state transition diagram) — XML-подобный формат программного средства KMS (knowledge modelling system) для представления и хранения информации о диаграммах (графах) переходов состояний;
- EETD (extended event tree diagram) — XML-подобный формат программного средства KMS для представления и хранения информации о деревьях событий.

Схема взаимодействия программных средств технологии с указанием их основных функций представлена на рис. 6.

В результате выполнения перечисленных этапов пользователь на основе онтологических моделей получает программную систему, основанную на знаниях, для решения задач исследования технического состояния и реализующую следующие функции:

- описание технического состояния объекта по имеющейся информации;
- выявление причинно-следственного комплекса факторов, обусловивших техническое состояние;

Т а б л и ц а 2. Фрагмент таблицы решений с описанием кинетики коррозионной усталости  
Table 2. A fragment of the decision table describing the “corrosion fatigue” kinetics

Свойство		Значение		
Идентификатор правила	R017	R018	R019	
Коэффициент уверенности	1	1	1	
Приоритет	0	0	0	
Условия				
Событие	Наименование	Слияние микротрещин	Развитие микротрещин, ориентированных в плоскостях, перпендикулярных максимальным растягивающим напряжениям	Образование микротрещин
	Кинетика	Коррозионная усталость	Коррозионная усталость	Коррозионная усталость
Действия (Решение)				
Событие	Наименование	Развитие микротрещин, ориентированных в плоскостях, перпендикулярных максимальным растягивающим напряжениям	Образование микротрещин	Зарождение и развитие внутреренных субмикротрещин вследствие растворения по плоскостям скольжения
	Кинетика	Коррозионная усталость	Коррозионная усталость	Коррозионная усталость
	Вероятность	1	1	1

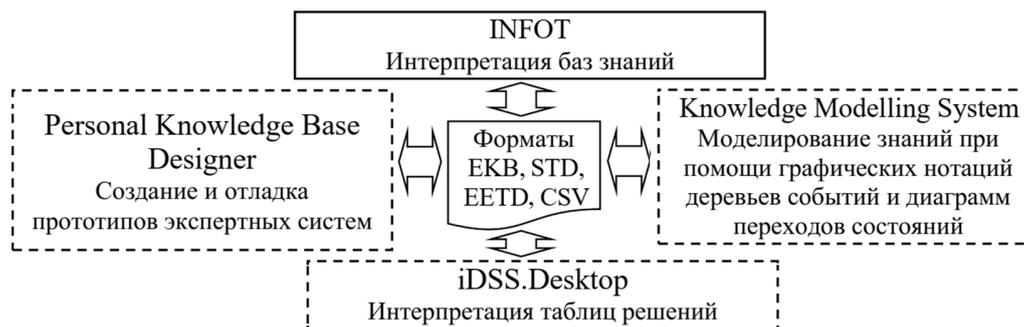


Рис. 6. Схема взаимодействия программных средств технологии  
 Fig. 6. The scheme of interaction of software technology

- диагностирование (идентификация) технического состояния объекта (с регистрацией результатов в базе данных);
- прогнозирование изменений технического состояния;
- генерирование специализированной, краткой, полной отчетной информации о техническом состоянии объекта.

Апробация подхода выполнена для решения задач экспертизы промышленной безопасности [21] и поддержки принятия решений при диагностировании систем воздушного судна [22].

## Заключение

Описана технология создания систем, основанных на знаниях, с использованием онтологических моделей и модельно-ориентированного подхода. Технология ориентирована на конечных пользователей — экспертов в некоторой предметной области. Именно поэтому особое внимание уделено описанию первого этапа технологии “Создание концептуальных (онтологических) моделей”. В частности, продемонстрированы последовательная трансформация теоретических моделей, описывающих динамику технического состояния, в онтологию, а затем в модели знаний различного вида, прецеденты, продукции и таблицы решений, а также формирование проблемно-ориентированных моделей — деревьев событий и диаграммы Исикавы.

Технология реализована в виде совокупности программных систем, интегрированных по данным. Апробация подхода выполнена для решения задач экспертизы промышленной безопасности и поддержки принятия решений при диагностировании систем воздушного судна.

**Благодарности.** Результаты получены в рамках госзадания Минобрнауки России по проекту “Методы и технологии облачной сервис-ориентированной цифровой платформы сбора, хранения и обработки больших объемов разноформатных междисциплинарных данных и знаний, основанные на применении искусственного интеллекта, модельно-управляемого подхода и машинного обучения” (№ госрегистрации: 121030500071-2).

## Список литературы

- [1] Махутов Н.А. Критериальная база прочности, ресурса, надежности, живучести и безопасности машин и человеко-машинных комплексов. Проблемы машиностроения и надежности машин. 2013; 42(5):25–36.

- [2] **Махутов Н.А., Берман А.Ф., Николайчук О.А.** Некоторые принципы самоорганизации для управления риском техногенных катастроф. Проблемы анализа риска. 2015; 12(4):34–45.
- [3] **Barricelli В.Р., Cassano F., Fogli D., Piccinno A.** End-user development, end-user programming and end-user software engineering: a systematic mapping study. Journal of Systems and Software. 2019; (149):101–137.
- [4] **Coronado E., Mastrogiovanni F., Indurkha B., Venture G.** Visual programming environments for end-user development of intelligent and social robots, a systematic review. Journal of Computer Languages. 2020; (58):100970.
- [5] **Da Silva A.R.** Model-driven engineering: a survey supported by the unified conceptual model computer languages. Systems & Structures. 2015; (43):139–155.
- [6] **Дородных Н.О., Юрин А.Ю.** Технология создания продукционных экспертных систем на основе модельных трансформаций. Новосибирск; 2019: 144.
- [7] **Берман А.Ф., Николайчук О.А.** Структуризация процесса исследования безопасности сложных технических систем. Проблемы безопасности и чрезвычайных ситуаций. 1999; (6):3–12.
- [8] **Берман А.Ф., Николайчук О.А.** Пространство технических состояний уникальных механических систем. Проблемы машиностроения и надежности машин. 2007; (1):14–22.
- [9] **Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И.** Инженерия знаний. Модели и методы. СПб.: Лань; 2016: 324.
- [10] **Berman A.F., Nikolaychuk O.A., Pavlov A.I.** The ontology model for automating the solution of multidisciplinary research tasks. Proceeding of the V International Workshop on Critical Infrastructures: Contingency Management, Intelligent, Agent-Based, Cloud Computing and Cyber Security (IWCI 2018). 2018; (158):1–6.
- [11] **Chandrasekaran B., Josephson J.R., Benjamins V.R.** Ontology of tasks and methods. IEEE Intelligent Systems. 1998; 14(1):20–26.
- [12] **Hitzler P., Gangemi A., Janowicz K., Krisnadhi A.A., Presutti V.** Ontology engineering with ontology design patterns: foundations and applications. Studies on the Semantic Web. Amsterdam: IOS Press/AKA; 2016: 388.
- [13] **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; 12(2):158–171.
- [14] **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):30–43.
- [15] **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):41–51.
- [16] **Aamodt A., Plaza E.** Case-based reasoning: foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications. 1994; 7(1):39–59.
- [17] **Bergmann R.** Experience management: foundations, development methodology, and internet-based applications. Berlin/Heidelberg: Springer; 2002: 398.
- [18] **De Mantaras L.R., McSherry D., Bridge D., Leake D., Smyth B., Craw S., Faltings B., Maher M.L., Cox M.T., Forbus K., Keane M., Aamodt A., Watson I.** Retrieval, reuse, revision and retention in case-based reasoning. Knowledge Engineering Review. 2005; 20(3):215–240.

- [19] **Nikolaychuk O.A., Yurin A.Yu.** Computer-aided identification of mechanical system's technical state with the aid of case-based reasoning. *Expert Systems with Applications*. 2008; (34):635–642.
- [20] **Pavlov A.I., Stolbov A.B.** Domain-oriented specialization tools for knowledge-based systems development platform. *CEUR Workshop Proceedings*. 2020; (2677):59–69.
- [21] **Берман А.Ф., Кузнецов К.А., Николайчук О.А., Павлов А.И., Юрин А.Ю.** Информационно-аналитическая поддержка экспертизы промышленной безопасности объектов химии, нефтехимии и нефтепереработки. *Химическое и нефтегазовое машиностроение*. 2018; (8):30–36.
- [22] **Dorodnykh N.O., Kotlov Y.V., Nikolaychuk O.A., Popov V.M., Yurin A.Yu.** Enduser development of knowledge bases for semi-automated formation of task cards. *CEUR Workshop Proceedings*. 2021; (2913):60–73.

---

Вычислительные технологии, 2023, том 28, № 4, с. 94–108. © ФИЦ ИВТ, 2023  
Computational Technologies, 2023, vol. 28, no. 4, pp. 94–108. © FRC ICT, 2023

ISSN 1560-7534  
eISSN 2313-691X

---

INFORMATION TECHNOLOGIES

---

DOI:10.25743/ICT.2023.28.4.008

**Technology for creating knowledge-based systems for studying the dynamics of the state of technical objects**

A. F. BERMAN, N. O. DORODNYKH, O. A. NIKOLAICHUK\*, A. YU. YURIN

Matrosov Institute of System Dynamics and Control Theory SB RAS, 664033, Irkutsk, Russia

\*Corresponding author: Olga A. Nikolaichuk, e-mail: nikoly@icc.ru

Received May 20, 2023, accepted June 19, 2023.

**Abstract**

The paper addresses the description of technology for the creation of knowledge-based systems. The systems are aimed at studying the dynamics of technical conditions with the help of a model-driven approach and ontological modelling. The technology is intended for end users who are experts in a particular subject domain. While studying the dynamics of the technical condition, the authors understand the problems of its genesis, identification, and forecasting. The technology is based on the principles of model-driven architecture (MDA), which defines the process of creating software and its components as a sequence of stages for building certain models and their transformation. The proposed technology implements the MDA approach for creating knowledge-based systems. The novelty of the technology is its problem-oriented specialization and ability to use various models for knowledge representation and methods of their processing, depending on the data and knowledge available to the researcher. The technology includes the following stages: the construction of problem-oriented domain models, in particular the ontology of the object under investigation, the task, and the methodology of its solution; the construction of platform-independent models — the description of rules, cases, decision tables and algorithms for solving the problem; the construction of platform-dependent models — within this stage, the end user must clarify the rule models, cases, and decision tables, taking into account the features of a certain knowledge representation/programming language. The technology is implemented in the form of a set of software tools integrated “by data”. The approbation of the approach was carried out in the cases of industrial safety inspection and decision support when diagnosing aircraft systems.

*Keywords:* dynamics of technical state, model-oriented approach, knowledge-based systems, ontology, case-based reasoning, rule-based system, decision tables.

*Citation:* Berman A.F., Dorodnykh N.O., Nikolaichuk O.A., Yurin A.Yu. Technology for creating knowledge-based systems for studying the dynamics of the state of technical objects. Computational Technologies. 2023; 28(4):94–108. DOI:10.25743/ICT.2023.28.4.008. (In Russ.)

**Acknowledgements.** The present study was supported by the Ministry of Education and Science of the Russian Federation (Project No. 121030500071-2 “Methods and technologies of a cloud-based service-oriented platform for collecting, storing and processing large volumes of multi-format interdisciplinary data and knowledge based upon the use of artificial intelligence, model-driven approach and machine learning”).

## References

1. **Makhutov N.A.** A criterion base for assessment of strength, lifetime, reliability, survivability, and security of machines and man-machine systems. *Journal of Machinery Manufacture and Reliability.* 2013; 42(5):25–36. (In Russ.)
2. **Makhutov N.A., Berman A.F., Nikolaichuk O.A.** Some principles of self-organization for managing the risk of man-made disasters. *Issues of Risk Analysis.* 2015; 12(4):34–45. (In Russ.)
3. **Barricelli B.R., Cassano F., Fogli D., Piccinno A.** End-user development, end-user programming and end-user software engineering: a systematic mapping study. *Journal of Systems and Software.* 2019; (149):101–137.
4. **Coronado E., Mastrogiovanni F., Indurkha B., Venture G.** Visual programming environments for end-user development of intelligent and social robots, a systematic review. *Journal of Computer Languages.* 2020; (58):100970.
5. **Da Silva A.R.** Model-driven engineering: a survey supported by the unified conceptual model computer languages. *Systems & Structures.* 2015; (43):139–155.
6. **Dorodnykh N.O., Yurin A.Yu.** Tekhnologiya sozdaniya produktsionnykh ekspertnykh sistem na osnove model'nykh transformatsiy [Technology for design of production expert systems based on model transformations]. Novosibirsk; 2019: 144. (In Russ.)
7. **Berman A.F., Nikolaichuk O.A.** Structuring the process of safety research of complex technical systems. *Security and Emergency Issues.* 1999; (6):3–12. (In Russ.)
8. **Berman A.F., Nikolaichuk O.A.** Technical state space of unique mechanical systems. *Journal of Machinery Manufacture and Reliability.* 2007; 36(1):10–16.
9. **Gavrilova T.A., Kudryavtsev D.V., Muromtsev D.I.** Inzheneriya znaniy. Modeli i metody [Engineering of knowledge. Models and methods]. St. Petersburg: Lan'; 2016: 324. (In Russ.)
10. **Berman A.F., Nikolaychuk O.A., Pavlov A.I.** The ontology model for automating the solution of multidisciplinary research tasks. *Proceeding of the V International Workshop on Critical Infrastructures: Contingency Management, Intelligent, Agent-Based, Cloud Computing and Cyber Security (IWCI 2018).* 2018; (158):1–6.
11. **Chandrasekaran B., Josephson J.R., Benjamins V.R.** Ontology of tasks and methods. *IEEE Intelligent Systems.* 1998; 14(1):20–26.
12. **Hitzler P., Gangemi A., Janowicz K., Krisnadhi A.A., Presutti V.** Ontology engineering with ontology design patterns: foundations and applications. *Studies on the Semantic Web.* Amsterdam: IOS Press/ AKA; 2016: 388.
13. **Dorodnykh N.O., Nikolaychuk O.A., Yurin A.Yu.** Using ontological content patterns engineering for formalize the process of maintenance and repair of critical objects. *Ontology of Designing.* 2022; 12(2):158–171. (In Russ.)
14. **Dorodnykh N.O., Nikolaychuk O.A., Yurin A.Yu.** Automated creation of rule-based knowledge bases on the basis of event trees. *Information and Mathematical Technologies in Science and Management.* 2017; 2(6):30–43. (In Russ.)
15. **Dorodnykh N.O., Nikolaychuk O.A., Yurin A.Yu.** Development of knowledge bases by means of transforming fishbone diagrams. *Vestnik Komp'uternykh i Infomatsionnykh Tekhnologii.* 2018; (4):41–51. (In Russ.)
16. **Aamodt A., Plaza E.** Case-based reasoning: foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications.* 1994; 7(1):39–59.
17. **Bergmann R.** Experience management: foundations, development methodology, and internet-based applications. Berlin/Heidelberg: Springer; 2002: 398.

18. **De Mantaras L.R., McSherry D., Bridge D., Leake D., Smyth B., Craw S., Faltings B., Maher M.L., Cox M.T., Forbus K., Keane M., Aamodt A., Watson I.** Retrieval, reuse, revision and retention in case-based reasoning. *Knowledge Engineering Review*. 2005; 20(3):215–240.
19. **Nikolaychuk O.A., Yurin A.Yu.** Computer-aided identification of mechanical system's technical state with the aid of case-based reasoning. *Expert Systems with Applications*. 2008; (34):635–642.
20. **Pavlov A.I., Stolbov A.B.** Domain-oriented specialization tools for knowledge-based systems development platform. *CEUR Workshop Proceedings*. 2020; (2677):59–69.
21. **Berman A.F., Nikolaychuk O.A., Pavlov A.I., Yurin A.Yu., Kuznetsov K.A.** Informational and analytical support of industrial safety expert review of chemical and petrochemical objects. *Khimicheskoe i Neftegazovoe Mashinostroenie*. 2018; (8):30–36. (In Russ.)
22. **Dorodnykh N.O., Kotlov Y.V., Nikolaychuk O.A., Popov V.M., Yurin A.Yu.** Enduser development of knowledge bases for semi-automated formation of task cards. *CEUR Workshop Proceedings*. 2021; (2913):60–73.