

МЕТОДОЛОГИЯ ПРОИЗВОДСТВА ЖИЗНЕСПОСОБНЫХ СИСТЕМ ДОВЕРИТЕЛЬНОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Е. А. ШАЛФЕЕВА

Аннотация. Представлены методы создания интеллектуальных систем, в которых для доверия пользователей обеспечиваются требуемый уровень знаний, объяснение решений и механизмы эволюционирования. На основе анализа исследования методов решения интеллектуальных задач предложен систематический подход к их автоматизации и установлен принципиально важный набор компонентов инфраструктуры для построения интеллектуальных систем. Установлены формальные характеристики задач интеллектуальной деятельности, используемые в процессе решения модели знаний, способы их обработки. Новая технология создания интеллектуальных систем акцентируется на декларативном представлении знаний, их прозрачном использовании и непрерывной адаптации к развитию предметной области. Для этого в работе поставлены и решены следующие задачи. 1. Классификация и спецификация задач интеллектуальной деятельности и анализ методов их решения. 2. Развитие онтологического подхода к формированию баз декларативных знаний для классов интеллектуальных задач для повышения повторной используемости готовых решений. 3. Разработка модели жизнеспособной системы с декларативными базами знаний и инструментами их эволюционирования. 4. Разработка методов оценивания и повышения качества баз знаний. 5. Разработка метода конструирования онтологических решателей с повторной используемостью готовых решений. 6. Разработка технологии производства, сопровождения и развития компонентов таких систем при поддержке облачного комплекса инструментальных средств. Поставленные задачи решались с использованием системного анализа, теории информации, теории графов и семантических сетей, онтологического инжиниринга, а также объектно-ориентированного программирования и облачных технологий.

Ключевые слова: задача интеллектуальной деятельности; декларативная база знаний; интеллектуальная система; объяснительный искусственный интеллект; жизнеспособная система; повторное использование.

ВВЕДЕНИЕ

На сегодняшний день актуальной для коллективов разработчиков программных систем является реализация всех текущих требований и механизмов, обеспечивающих неизбежное последующее сопровождение.

При автоматизации области со сложными и ответственными задачами принятия решений создаются системы искусственного интеллекта. При их длительной эксплуатации ожидаются не только изменения требований пользователей к функциональности и пользовательскому интерфейсу, но и изменения знаний предметной области. Это меняет характер сопровождения на эволюционирование.

Процесс сопровождения занимает в процессе жизненного цикла программной системы, по оценкам специалистов, не менее 50% трудозатрат, поэтому в программной инженерии для упрощения сопровождаемости ПС применяют архитектурные решения (разделение на слабо сцепленные компоненты с логически понятными функциями), использование типовых архитектур, декларативное представление компонентов ПС, средства автоматизации построения проектных моделей и автоматической генерации по ним фрагментов кода, модельно-управляемые архитектуры и подход к созданию программных систем, стандартизацию сервисов и интерфейсов, разделение компетенций между разными типами разработчиков.

Программная система, которая способна адаптироваться во времени под воздействием человека или инструментов (адаптируемость в ответ на новые требования) и при этом быть адаптивной к изменениям в среде, называется «жизнеспособной» [1, 2].

Поддержка эволюционирования системы искусственного интеллекта предполагает трудовые затраты не меньшие, чем для сопровождения, что накладывает необходимость использования дополнительных специализированных механизмов обеспечения их жизнеспособности.

СТЕПЕНЬ РАЗРАБОТАННОСТИ ТЕМЫ И ОБСУЖДЕНИЕ РЕШАЕМОЙ ЗАДАЧИ

К настоящему времени для особо значимых для человека областей создается множество интеллектуальных программных сервисов и систем разного масштаба. Некоторые используют явные знания, некоторые обучаются на массивах данных.

Для поддержки сложных и ответственных задач и решений от системы искусственного интеллекта требуются рекомендации с объяснениями. Такая поддержка возможна на основе формализуемых знаний, и такие системы имеют самостоятельный архитектурный компонент – базу знаний (БЗ). Предлагаемые российскими и зарубежными учеными методы и технологии позволяют создавать базы знаний и системы, основанные на знаниях, называемые интеллектуальными системами (ИС), тестировать их и вводить в эксплуатацию. Для областей, где число правил измеряется сотнями, долгое время был пригоден подход с универсальным представлением знаний в виде правил, обрабатываемых единой «машиной вывода». Расширение и развитие знаний в таком виде, особенно силами экспертов, становится практически невозможным для задач более сложных, чем задача идентификации или классификации сущностей. В публикациях не уделено достаточного внимания методам и инструментам развития ИС в процессе их эксплуатации, а также методам обеспечения доверия к выдаваемым советам или результатам.

Использование онтологий позволило абстрагировать некоторую «описательную» часть знаний (чаще как классы сущностей) и начать применять дедуктивный вывод, «используя в посылках и заключениях продукционных правил введенные в онтологиях понятия, т. е. рассуждать на основе онтологии («ontological reasoning»). Новые среды, сочетающие в себе онтологии и логическое программирование (например, «ontological logic programming»), более эффективны для реализации, но в отношении последующего сопровождения сделали лишь небольшой шаг – декларативное представление некоторой части знаний – той, где для сущностей можно определить свойств, ввести бинарные связи и выявить иерархии классов таких сущностей.

Многолетний опыт разработки ИС в различных ПрОбл и множественные публикации [3–6] показали, что объем серьёзных знаний не позволяет формализовать вручную базы знаний (БЗ) и проверить качество полностью за приемлемое время силами даже согласованного коллектива экспертов. Кроме того, знания в динамичных предметных областях (ПрОбл) расширяются и корректируются.

Для обеспечения возможности внесения знаний, заслуживающих доверия специалистов, проверки их применения и перманентного усовершенствования БЗ вышеперечисленным разработкам не хватает технологически обеспеченного отделения декларативных знаний от процедурных. Такое отделение лежит в основе независимости процессов разработки баз знаний и решателей, механизмов создания и модифицирования понятных экспертам БЗ, разработки процедур оценивания БЗ в течение всего процесса использования ИС. Ориентированность механизмов на экспертов означает возможность развития ими систем с базами знаний (СБЗ) без участия программистов и инженеров по знаниям. Технологически обеспеченная независимость процессов разработки баз знаний и решателей означает возможность автоматизированного согласования допустимых действий экспертов и программистов.

Целью работы является разработка методологии создания интеллектуальных систем на основе онтологий с декларативным представлением проверяемых баз знаний, объяснением генерируемых решений и механизмами эволюционирования.

Для достижения этой цели важно: а) учесть хорошо проявившие себя методы и алгоритмы решения задач интеллектуальной деятельности; б) предложить систематический подход к решению задач и непрерывной адаптации к развитию ПрОбл; в) установить принципиально важный набор компонентов инфраструктуры для построения СБЗ.

АНАЛИЗ СОСТОЯНИЯ ПРОБЛЕМЫ

Проведен анализ литературных источников по предметным областям (ПрОбл) со сложными и ответственными задачами принятия разных типов решений (области с интеллектуальной деятельностью), по вопросам автоматизации выработки рекомендаций на основе баз знаний и построения объяснений таких рекомендаций.

Множество публикаций [7–9] посвящены обсуждению того, для решения каких задач следует разрабатывать так называемые экспертные системы. Были предложены некоторые классификации задач интеллектуальной деятельности, постановки и описания отдельных задач. Несмотря на отсутствие единой классификации задач были предложены универсальные методы решения и технологии, и начали создаваться многочисленные *системы с базами знаний* (экспертные системы) для поддержки решения таких задач.

В этих системах задачи решаются «с применением эвристик, включая эмпирическую индукцию, аналогию и дедукцию» [10], а в гибридных интеллектуальных – с использованием нескольких интеллектуальных методов. Современные СБЗ могут получать данные от аппаратных средств измерения и часто интегрируются со статистическими и другими программными компонентами. Огромная трудоемкость их создания и ответственность за возлагаемую на них поддержку деятельности специалистов выдвигает непереносимое требование к возможности их длительной эксплуатации (с перманентным усовершенствованием знаний).

Однако в реальной сфере практической интеллектуальной деятельности очень мало случаев повседневной и продолжительной эксплуатации таких систем для поддержки принятия ответственных решений, поскольку технологии, предлагаемые российскими и зарубежными коллективами, сосредоточены преимущественно на методах эффективной разработки.

Используемые формализованные базы знаний часто содержат упрощенные модели процессов и объектов. Реальные задачи неизмеримо сложнее, а объем информации, привлекаемый для их решения, не позволяет формализовать все накопленные вербально представляемые знания раз и навсегда за приемлемое время. Поэтому во многих важных областях приходится регулярно расширять или уточнять формализуемые базы знаний. Разработаны различные подходы к пополнению знаний и дообучению систем [6, 11, 12]. Но получаемые новые версии баз знаний либо не удобны для оценивания экспертами, либо требуют перепрограммирования компонентов системы.

Инструменты и технологии создания и усовершенствования ИС в процессе всего жизненного цикла (ЖЦ) [4, 5] должны обеспечивать:

- 1) понятность представления знаний, особенно таких как причинно-следственные, временные, пространственные связи, для экспертов ПрОбл;
- 2) возможность формирования и сопровождения знаний, не влияющих на работоспособность решателей;
- 3) механизмы оценки баз знаний, сформированных вручную, их обучения и дообучения;
- 4) поддержку коллективной разработки и сопровождения;
- 5) разделение компетенций разработчиков с предоставлением каждому типу разработчиков инструментов, ориентированных на их специализацию;
- 6) интегрированность со сторонними ПС.

Ни одни из известных универсальных технологий и инструментов для создания ИС не поддерживают всех указанных требований одновременно. Для обеспечения возможности согласованно и постоянно усовершенствовать БЗ этим технологиям не хватает:

- либо отделения знаний от данных и декларативных знаний от процедурных знаний;
- либо декларативного представления причинно-следственных связей;

- либо доступности инструментов коллективной разработки;
- либо интегрированности со средствами обучения.

Другие технологии, которые минимизируют эти недостатки, предлагают узкоспециализированные решения: проблемно-ориентированные средства автоматизации или проблемно-предметно-ориентированные средства автоматизации (например, только диагностика и только в медицине).

По аналогии с тем, как жизнеспособность программных систем обеспечивается к изменениям в среде и в ответ на новые требования, жизнеспособность для системы, основанной на знаниях, проявляется, прежде всего, в условиях изменчивости знаний предметной области (и реже – среды функционирования или средств пользовательского интерфейса). Поэтому для большинства предметных областей, связанных с решением интеллектуальных задач, подразумевается эволюционируемость знаний. А в предметных областях, где важны влияние факторов и событий на состояние системы (объекта, ситуации), их изменение во времени, влияние индивидуальных характеристик объекта, системы и одних процессов на другие, эволюционируемость баз знаний – главный «вызов» современных «условий» по отношению к СБЗ.

Таким образом, для создания жизнеспособных и заслуживающих доверия программных систем поддержки специалистов, решающих важные для науки и практики интеллектуальные задачи, необходима современная методология. Технология создания эволюционирующих интеллектуальных программных систем с базами знаний должна реализовать вышеперечисленные требования.

КЛАССИФИКАЦИЯ И СПЕЦИФИКАЦИЯ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ И АНАЛИЗ МЕТОДОВ ИХ РЕШЕНИЯ

Хотя и предложены некоторые классификации задач интеллектуальной деятельности, отсутствует единство в их количестве и названиях. Поэтому требуется их систематизация. В зависимости от того, требуется ли сформировать БЗ или она задана, можно различать задачи *поиска гипотезы о решении, критики гипотезы* (как вариантов «дедукции») и задачи *индуктивного обобщения данных* для построения БЗ. А в зависимости от того, заданы ли в качестве входных данных результаты наблюдения ситуации или условия на решение задачи, можно различать *задачи анализа результатов наблюдений* и *задачи учета условий и ограничений на решение*.

Применимыми на практике являются следующие пять групп задач, получаемые путем комбинации вышеописанных абстрактных категорий.

Задача поиска гипотез, объясняющих результаты наблюдений, рассматривается как комбинация задачи поиска гипотез и задачи анализа результатов наблюдений, в которой заданы БЗ и результаты наблюдений ситуации, требуется найти все гипотезы, соответствующие результатам наблюдений и формализованным знаниям.

Задача проектирования рассматривается как комбинация задачи поиска гипотез и задачи анализа ограничений на решение, в которой заданы БЗ и непустое множество ограничений на результат решения задачи: требуется найти все проекты, соответствующие БЗ и ограничениям на решение задачи.

Задача критики объяснения результатов наблюдений рассматривается как комбинация задачи критики гипотезы и задачи анализа результатов наблюдений, в которой заданы БЗ и результаты наблюдения ситуации и их объяснение и требуется установить, соответствует ли объяснение результатов наблюдения базе знаний.

Задача критики проекта рассматривается как комбинация задачи критики гипотезы и задачи анализа ограничений на решение, в которой требуется проверить соответствие проекта базе знаний и ограничениям, которым он должен удовлетворять.

Задача поиска БЗ по обучающей выборке ситуаций, представленных результатами наблюдений, рассматривается как комбинация задачи индукции и задачи анализа результатов

наблюдений, в которой каждый элемент обучающей выборки представляет собой результаты наблюдения некоторой ситуации, и по такой обучающей выборке требуется сформировать БЗ.

Во многих ПрОбл существуют классы ситуаций и решаются задачи, учитывающие эти классификации: поиск гипотез о классах, критики гипотез, формирование знаний о классах по обучающей выборке ситуаций, представленных результатами наблюдений. Они рассматриваются как уточнение вышеназванных абстрактных задач, когда заданы не только результаты наблюдений ситуаций, но и класс, к которому относится каждая ситуация.

Во многих ПрОбл анализируются или проектируются сложные объекты или системы, состоящие из компонентов, и решаются задачи, учитывающие эти компоненты и отношения между ними (в том числе пространственные). Некоторые задачи связаны с моделями знаний, обязательно включающими термины, денотаты которых являются функциональными отношениями, зависящими от времени.

Задачи анализа [13,14], критики гипотез, проектирования [15, 16], критики проекта могут ставиться для динамических или статических систем; в задачах проектирования динамической системы присутствуют отношения или формулы, учитывающие *временные* аспекты развивающейся или функционирующей системы.

Для динамических систем ставятся задачи прогноза, мониторинга – уточнение задачи поиска гипотез, объясняющих результаты наблюдений. В задаче мониторинга требуется определить те наблюдения динамической системы, по результатам которых можно отслеживать критические состояния системы.

В ряде ПрОбл решаются задачи относительно систем, в которых во времени протекают процессы, связанные между собой посредством причинно-следственных отношений. Процессы, протекающие в такой системе, условно можно разделить на внешние (наблюдаемые) и внутренние. Наблюдаемые процессы обычно называют признаками, они имеют значения, которые изменяются с течением времени. Наблюдаемыми могут быть происходящие в различные моменты времени события, внешние по отношению к системе, но воздействующие на протекающие в ней. Например, в *задаче диагностики* требуется по результатам наблюдения признаков, характеристик системы и произошедших событий определить возможные модели изменений в системе. В *задаче прогноза результата воздействий*, зная характеристики системы, диагноз и планируемые события, определяют значения признаков в некоторые моменты времени в таких причинно-следственных моделях, которые соответствуют БЗ.

Во многих ПрОбл решаются задачи, рассматривающие действия (упорядоченные хотя бы частично), которые должны привести к некоторой заданной цели.

В задаче *планирования* (или *линейного планирования*), зная текущее состояние системы и характеристики целевого состояния системы, множество *действий*, которые можно выполнять, *предусловие* и *постусловие* для каждого из них, надо определить такую комбинацию частично упорядоченных действий, которые изменят текущее состояние системы на целевое (выбор последовательности действий по достижению поставленной цели, упорядочение возможных действий для достижения заданного суммарного эффекта) [17, 18].

В задаче *управления* [18, 19] требуется, зная характеристики и признаки системы или процесса в текущем состоянии (а также часто вид отклонения от нормы или диагноз) и целевые значения признаков, определить и применить совокупность воздействий в соответствующие моменты времени, чтобы признаки стали удовлетворять этим условиям.

В задаче *планирования управления* (планирования воздействий на систему (объект, ситуацию), то есть *управления* в смысле *Clancey* [9], зная характеристики системы, вид ее отклонения от нормы или диагноз и целевые значения признаков, определить такую совокупность воздействий и соответствующие им моменты времени, при которых признаки (как функции времени в причинно-следственных моделях системы) будут удовлетворять этим условиям.

В соответствии с такой классификацией и иерархией задач их постановки, выраженные в терминах единых математических абстракций, тоже образуют иерархию: от более абстрактных к уточняющим [20].

Пример постановки задачи для диагностики таков.

Дано: R (результаты наблюдения признаков, характеристик системы и произошедших событий); K_n (база знаний, связывающая диагноз Δ с результатами наблюдения R).

Найти: все возможные причинно-следственные модели ($AS(R)$) системы, согласованные с результатами наблюдений (R), относительно которых все предложения из БЗ (K_n) истинны.

База знаний (K_n) для абстрактной области состоит из предложений об отклонениях $\{f_i^n\}$, каждое из которых связывает отклонение с его внешними проявлениями в виде признаков со значениями $\{signName_{ik}, signValuesRange_{ijk}\}$.

Для области, где проявления изменчивы, $f_{in} = \{(period_{ij} + \{signName_{jk}, signValuesRange_{ijk}\})\}^{n1}$.

Для области, где набор проявлений зависит от условий, $f_{in} = [\{factor_m | event_m\} + \{signName_{jk}, signValuesRange_{ijk}\}]^{n1}$.

Такой иерархической классификацией охвачен практически весь спектр задач, относящихся к области разработки систем, основанных на знаниях.

В искусственном интеллекте поддержка решения таких задач связана с направлением разработки систем, основанных на явных знаниях. Предполагается, что существует такая «правильная» база знаний $k^* \in K_n(X, Y)$, что для любой входной ситуации найдется решение на основе этой базы знаний:

$$\forall x \in X \exists y \in Y P(x, k^*, y),$$

где предикат $P(x, k, y)$ есть спецификация задачи на основе базы знаний.

Если справедливо утверждение о существовании «правильной» базы знаний, то можно разработать и реализовать алгоритм решения задачи

$$(A: \langle X, K_n(X, Y) \rangle \rightarrow Y).$$

Для выявления общих, повторяемых методов, алгоритмов и процедур решения задач из представленной классификации проведен их анализ с применением обобщенных онтологий соответствующих знаний. В областях и задачах анализа результатов наблюдений часто рассматривают развивающийся во времени процесс, который может отклоняться от нормального случая, а управляемая система – от нормального функционирования.

Все виды связи таких отклонений с причинами, факторами, внешними проявлениями должны поддерживаться онтологией. В обобщенной онтологии причинно-следственных связей определены основные типы и подтипы предложений для представления связей отклонений в функционировании системы (объекта) с причинами, факторами, проявлениями. При решении задач, связанных с управлением ситуациями, важны объект воздействия, его свойства, воздействующий объект и/или средства воздействия, характеристики воздействия, действия и состояния.

Обобщающая модель знаний *о процессах* (или динамических объектах, функционирующих системах) для решения задач, связанных с анализом динамических ситуаций, может быть выражена (языком «content description notation») и сгруппирована так:

$$\langle f_{dyn}, [\{factor_i\} +] \{ [\{factor_k\} +] \{(period_i + interval_i) + \{sign_{jk}, range_i \text{ of } sign_{jk} \text{ in } period_i} + \{event_u + (new-range_{iu} \text{ of } sign \text{ values} | sign \text{ values } trend_{iu}) + delay_{kju}\}^{n2}\}] \} \}^{n1} \rangle.$$

Структурированная обобщающая модель знаний *об аномальных процессах* может быть выражена так:

$$\langle f_{in}, [\{factor_m | event_m\} +] [CriticRate +] \{ [\{factor_{ij}\} +] \{(period_{ij} + [StageCriticality +] interval_i) + \{signName_{jk}, signValuesRange_{ijk}\} + \{event_u + new-range_i \text{ of } signValues_{jkju} + delay_{kju}\}^{n2}\} + signMinQaunt_{ji}\} \}^{n1} \rangle.$$

Модель знаний для решения задач, *связанных с управлением ситуациями*, выглядит так:

$$\langle \text{ObjClass}, \{\text{ObjProp}_i\}^{i=1, \dots, I}, \{\text{Actor}_j, \{\text{ActProp}_{i3}\}^{i3=1, \dots, I}, \text{impact}_{i2}, \{\text{ImpChrct}_{i4}\}^{i4=1, \dots, I}\}^{j=1, \dots, J} \rangle$$

$$\langle \text{Actor}_j, \text{ObjClass}, \text{St}_{ij}, \text{Impact}_j, \text{St}_{i(j+1)} \rangle.$$

Были выделены *вычислительные операции* для реализации методов решения задач разных типов. По каждой обобщенной онтологии знаний, содержащей несколько типов причинно-следственных связей, специфицированы необходимые *операции запросов* к базе знаний. Выявлено общее множество вычислительных операций, требуемых для реализации решателей задач разных типов (распознавания, диагностики, запроса дополнительной информации; планирование воздействий и др.):

- проверить выполнение условий рассмотрения гипотезы-диагноза;
- проверить выполнение условий признакового комплекса;
- найти факты и наблюдения для признакового комплекса;
- проверить соответствие результатов наблюдения вариантам динамики признака и т. д.

Общие вычислительные операции претендуют на повторную используемость в алгоритмах специализированных программных решателей таких задач.

МОДЕЛЬ СИСТЕМЫ ПОДДЕРЖКИ РЕШЕНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ ЗАДАЧИ

Модель СБЗ (KBS-Arch), решающей одну задачу на основе баз знаний, можно представить так:

$$\text{KBS-Arch} = \langle \bigcup_{i=1, N1} (\text{KB}_i), \text{Solv}_1, \bigcup_{m=0, M} (\text{U}_{i_m}), \bigcup_{p=0, P} (\text{DB}_p) \rangle.$$

Здесь совокупность баз знаний обозначена как $\bigcup_{i=1, N1} (\text{KB}_i)$, программный решатель – как Solv_1 ; часто необходимы компоненты, обеспечивающие пользовательский интерфейс (U_{i_m}), и хранилища оперативной или справочной информации (DB_p). Решатель Solv_j – программно реализованное «рассуждение», алгоритм, проводящий обход каждой Базы знаний KB_{i_u} (на основе знания ее структуры) для сопоставления ее сведений входным данным (фактам или условиям).

Программные внутренние интерфейсы ($\Delta i \epsilon$), обеспечивающие интеграцию этих компонентов, особенно между Solv_1 , между Solv_1 и DB_p , могут быть явно указаны в этой модели путем добавления ($\bigcup_{p=1, P} (\Delta i \epsilon (\text{Solv}_1, \text{DB}_p))$, $\bigcup_{m=1, N1} (\Delta i \epsilon (\text{Solv}_1, \text{KB}_i))$, ...

Каждый решатель Solv_i для выдвижения и объяснения гипотез обрабатывает входную информацию и строит выходную, фиксируя в ней шаги принятия или отклонения возможных логических заключений.

На практике в информационной поддержке нуждаются как задачи одного из классов, так и композиции нескольких таких задач [20]. И *системный анализ* ПрОбл, предшествующий началу автоматизации интеллектуальной деятельности, выявляет все задачи, нуждающиеся в информационной поддержке на основе знаний. А *системное проектирование* распределяет интеллектуальные задачи и их подзадачи и сопутствующие функциональные требования по компонентам системы, в частности, решателям известных типов.

В результате СБЗ, предназначенная для поддержки решения нескольких подзадач разного типа, имеет в своей архитектуре несколько программных решателей (Solv_j), каждый из которых может обращаться к разным или общим БЗ. Дополнительные нужные подзадачи, вычислительные и другие, решаемые для тех же данных (документов), могут создаваться, подключаться как компоненты или микросервисы (Soft_k).

Архитектурную модель *системы, основанной на знаниях*, поддерживающей решение нескольких задач и интегрированной с другими программными компонентами (Soft_k), можно представить так:

$$\text{KBS-Arch} = \langle \bigcup_{i=1, N1} (\text{KB}_i), \bigcup_{j=1, N2} (\text{Solv}_j), \bigcup_{m=1, N3} (\text{U}_{i_m}),$$

$$\bigcup_{p=1, N4} (\text{DB}_p), \bigcup_{k=1, N5} (\text{Soft}_k) \rangle.$$

При разработке систем, основанных на знаниях, специальное требование предъявляется к представлению информации – все должно быть в форме, понятной пользователям без посредников: данные, знания, результат (гипотезы решения задач и объяснение гипотез).

Для формирования интерпретируемых баз сложноустроенных знаний (для диагностики, прогнозирования или др.) единственно пригоден онтологический подход, который обеспечивает смысл представляемых сведений и понимаемость информации сообществом специалистов.

Формализуемая *онтология ПрОбл* охватывает все виды связей и зависимостей между понятиями.

Онтология ПрОбл (Onto) для решения интеллектуальной задачи включает онтологию знаний (KnOnt) о связях терминов, существенных для этой задачи, *онтологию* входных данных (фактов) и ожидаемых результатов (SitOnt); ограничения интерпретации для них и *онтологические соглашения* (Agreem) о правилах сопоставления фактов знаниям. Номенклатура названий для именования понятий, разрешенных или рекомендуемых в сообществе специалистов, часто формируется в самостоятельный терминологический словарь (или справочник (Dict)) и может считаться частью онтологии ПрОбл, поскольку является результатом соглашения экспертов.

$$\text{Onto} = \text{KnOnt} \cup \text{SitOnt} \cup \text{Agreem} \cup \text{Dict}.$$

Ключевая роль отводится онтологии знаний (KnOnt), устанавливающей достаточную систему связи профессиональных понятий. Их формализованное представление, отделенное от собственно профессиональных знаний, позволяет самостоятельно разрабатывать каждый онтологический компонент, рассчитывая на их интегрируемость.

Онтология знаний фиксирует термины, связи, утверждения, достаточные для рассуждений и обсуждения таких ситуаций и решений.

Знания и другая информация должны формироваться в соответствии со структурой, ограничениями и соглашениями, определяемыми этой онтологией ПрОбл, а также и с помощью единых понятий из словаря предметной области.

Онтологический решатель Solv_j производит обход каждой Базы знаний KB_{ii} на основе знания ее структуры и онтологии в целом – KnOnt_{ii}). Решатель, способный выдвигать гипотезы на основе онтологических соглашений, структуры предложений и ограничений на интерпретацию, сохраняет аргументы в пользу гипотез о решении (объясняет их).

Каждый программный решатель создается по онтологии знаний, чтобы работать с любой БЗ, сделанной в терминах этой онтологии.

Включение в архитектуру декларативных БЗ (одновременно понятных человеку и программному обработчику) – ключевой принцип проектирования жизнеспособных СБЗ. С учетом этого принципа модель жизнеспособной СБЗ выглядит как

$$\text{KBS-Arch} = \langle \cup_{i=1, N1} (\text{KB}_i(\text{KnOnt}, \text{Dict})), \cup_{j=1, N2} (\text{Solv}_j(\text{KnOnt}, \text{SitOnt}, \text{Agreem})), \\ \cup_{m=1, N3} (\text{Ui}_m(\text{SitOnt})), \cup_{p=1, N4} (\text{DB}_p(\text{SitOnt}, \text{Dict})), \cup_{k=1, N5} (\text{Soft}_k) \rangle.$$

Конструирование СБЗ производится с использованием онтологического подхода, инструментарий для формирования всех информационных компонентов поддерживает соответствие онтологии предметной области.

При промышленной разработке СБЗ окончательная версия *онтологии ПрОбл* должна существовать к началу работ по созданию компонентов системы, поскольку все перечисленные компоненты (реализующие интеллектуальность в системе) разрабатываются на ее основе.

Для применения интеллектуальных сервисов в практике ответственных специалистов требуется обеспечение доверия к советам или результатам, которые будут вырабатываться.

Если нет возможности продемонстрировать 100%-ю точность на любых подмножествах прецедентов, то сочетание 100%-й правильности (на тестовых подмножествах) со способностью вести диалог (направленный на повышение точности результата) и со способностью объяснять гипотезы создает *доверие*.

Правильность (для задач анализа) означает наличие правильного (будущего) решения среди неотвергнутых гипотез. Диалог – путь к исключению ненужных гипотез, то есть к уточнению.

Правильность (для задач синтеза или учета условий и ограничений на решения) обычно означает наличие гипотезы-решения, в некоторой мере обобщенного, но содержащего внутри себя как частный случай будущее решение (обобщенное не будет отвергнуто при поступлении дополнительной информации). Диалог – путь к уточнению гипотезы.

Рассмотрение аргументации в пользу оставленных гипотез (и против откинутых) – есть «разговор» в профессиональных терминах.

Для проверки наличия в СППР способности выдавать только правильные гипотезы, направленного диалога и точной аргументации требуется особое двухуровневое тестирование (сначала каждый компонент, а потом целиком система).

МЕТОДЫ ОБЕСПЕЧЕНИЯ КАЧЕСТВА ФОРМИРУЕМЫХ ДЕКЛАРАТИВНЫХ ИНФОРМАЦИОННЫХ КОМПОНЕНТОВ

Конструирование СБЗ производится с использованием онтологического подхода, инструментарий для формирования всех информационных компонентов поддерживает их соответствие онтологии предметной области. Поэтому формальная правильность и пригодность для обработки гарантирована. Однако для правильных решений требуется правильный смысл, полнота и другие свойства похожести на «копилку знаний» в области и обиходе ее специалистов.

Актуальность этой «заботы» о качестве знаний (баз, коллекций и банков знаний) становится все очевиднее, как и актуальность интеллектуальной поддержки принятия решений в особо значимых для человека областях.

Интеллектуальный программный помощник полезен для некоторого специалиста, если мера качества БЗ «лучше» знаний этого специалиста. Тогда в некоторых ситуациях сервис поможет ему: с помощью БЗ он правильно сможет решить большее множество задач, чем полагаясь только на собственные знания. Причина в том, что множество задач с известным решением (базы эталонных прецедентов) составлено из ситуаций, для которых сообщество специалистов уже знают правильное решение (и оно задокументировано).

Создаваемые интеллектуальные программные сервисы должны оставаться полезными в течение всего времени использования. Для этого нужна такая система управления базами знаний, которая будет обеспечивать их пригодность (готовность, качество) и их усовершенствование в процессе эксплуатации.

Уникальной мерой качества БЗ становится такая, которая соотносит ее с задокументированным «опытом» решения задач. *Мера качества знаний* – то множество задач, которые именно эти знания позволяют решить правильно (когда БЗ – компонент интеллектуального сервиса). Эта мера может рассматриваться и количественно – как доля правильно решаемых задач по отношению к эталонному множеству задач (задач с известным решением). Такая мера является условно-объективной по отношению к текущему моменту познания и накопления опыта в предметной области [21].

Применяемое для оценки БЗ или СППР (СБЗ) эталонное множество может обнаружить, что в БЗ отсутствует категория (вариант исхода, вид риска, диагноз, класс), к которой относится прецедент (эталонный). Такого рода случаи (после утверждения ответственными экспертами) будут использованы для расширения знаний (добавления сведений и законов именно по этой категории). Или может обнаружить, что категория в БЗ есть, но она отвергается для прецедента, то есть ассоциирована с другими наблюдениями. Такого рода случаи приведут

к расширению знаний по этой категории, чтобы охватывался и найденный случай проявления (после утверждения экспертами). Или может обнаружить, что в БЗ категория не подтверждается полностью (то есть частично другая). Это означает либо устаревание знаний, которые излишне категоричны в отличие от действительности, допускающей «стертые картины», ранее неизученные влияющие факторы и т. д., либо нетипичный случай (кандидат на попадание в Базу прецедентов). Устаревание же (подтвержденное экспертами) корректируется в БЗ, например, механизмами нечеткости проявлений.

Во многих областях, где повседневные задачи «уникальны», с точки зрения количества и содержания исходных данных или редко повторяются, эталонное множество обязано увеличиваться. И измерение качества БЗ в этом смысле становится частью непрерывного мониторинга компонентов СБЗ.

В медицине *прецедентами для оценивания* предлагается считать множество завершенных историй болезни, когда их правильные диагнозы и методы лечения становятся известны. Если при оценивании правильности и точности баз знаний выявлены прецеденты, где по БЗ не генерируется нужное решение, прецедент попадает в обучающую выборку (для до-обучения и усовершенствования). В рамках модели монотонного улучшения качества формализованных баз знаний предложен алгоритм улучшения темпоральных БЗ.

Хотя СППР перед выпуском проверяется на эталонах, тестах и выборках данных, но в реальных условиях (то есть в условиях эксплуатации сторонними пользователями с новыми входными данными) интеллектуальные программные сервисы практически могут оказаться не «владеющими» всеми знаниями для выдачи адекватного (потребителю) совета, результата, решения. Поэтому в сложных и ответственных областях (в медицине – особенно) есть все основания применять, объединять и внедрять несколько способов поддержки решений с налаженными процедурами их последующего оценивания (внешней валидации) и сравнения гипотез и решений.

Способ поддержки решений по базам знаний не только обеспечивается налаженными процедурами последующего оценивания, но и предусматривает сравнение «уровня» (полноты, современности и др.) разных (по содержанию) версий БЗ и версий БЗ, создаваемых разными методами.

Базы, содержащие предметные и проблемные знания, формируются (а) путем распознавания текстовых знаний, например учебников, (б) «индуктивно» (автоматически) по обучающим выборкам из датасетов, архивов и баз данных, (в) силами экспертов в интерактивных редакторах с поддержкой понятийной структуры и использованием профессиональных словарей (рисунок 1).

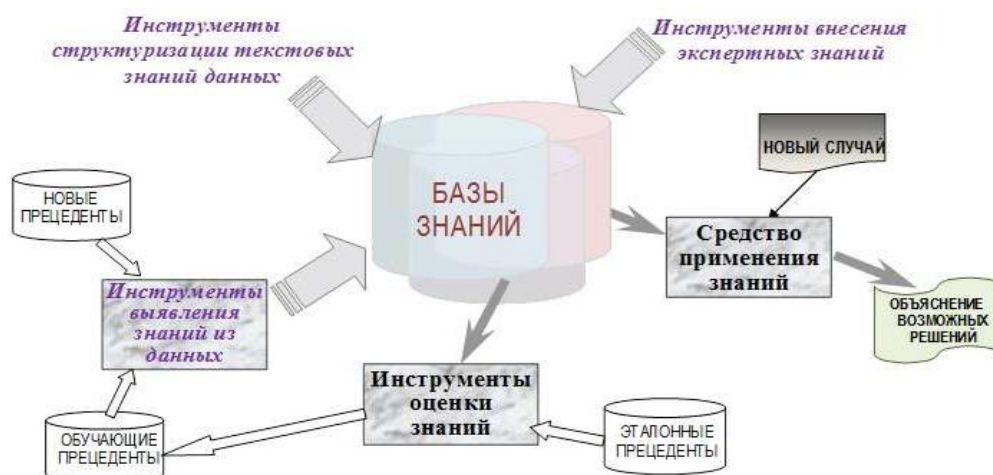


Рис. 1 Средства, влияющие на качество БЗ.

Для объяснительного ИИ применяемые знания, независимо от их источника, должны быть представлены декларативно. При этом в сложных и ответственных областях есть все основания использовать разные или даже все способы их накопления и формирования.

Готовность к внесению изменений плюс метод перманентного монотонного усовершенствования баз знаний с оригинальной мерой качества БЗ (а также метод оценивания корректности, наличия дефектов и несогласованностей в БЗ) суть механизмов эволюционирования. По отношению к БЗ применяется регулярная проверка после каждого изменения и расширения (с целью ее улучшения), включающая проверку правильности решения с ее помощью эталонных задач.

МОДЕЛЬ ЖИЗНЕСПОСОБНОЙ СИСТЕМЫ ПОДДЕРЖКИ ИНТЕЛЛЕКТУАЛЬНЫХ РЕШЕНИЙ

Обеспечение доверия (через правильность, способность вести направленный диалог и способность объяснять гипотезы) дает условия для принятия специалистом решения в сложной ситуации.

Также важно, чтобы ИС (с БЗ) «не отставало» от развития знаний области. (На первый взгляд, *grt*-интеллект делает это. Но ответственный специалист обратится к нему только из любопытства. Ведь обучение таких систем не прозрачно и не управляемо в том смысле, чтобы сохранять в их «копилке знаний» только одобренные экспертами знания. И вряд ли доступны отчеты по тестированию правильности, направленного диалога и точной аргументации.) В то же время объяснимый искусственный интеллект (ХАИ) с БЗ делает принятие решений в системах ИИ прозрачным и доступным.

Благодаря декларативным (понятным человеку и ЭВМ) базам знаний подбор и перебор утверждений с искомым смыслом проводится и для выдачи ответов на запросы, и для объяснения ответов. А при проверке полноты декларативных баз знаний (и качества решений по ним) становятся очевидные ее неполные или устаревшие «места».

Инструментарий, позволяющий создавать и улучшать компоненты систем с базами знаний, должен, как минимум, предоставлять:

- средства редактирования БЗ;
- средства ввода данных и/или редактирования баз данных (БД) и просмотра данных и результатов;
- средства формирования структуры результатов решения задачи и их объяснения;
- библиотеку программных решателей, соответствующих классам решаемых задач;
- библиотеку компонентов решателей – программных единиц, соответствующих типам обрабатываемых отношений между понятиями;
- средства поиска и выбора повторно используемых компонентов;
- средства интеграции БЗ и решателей;
- средства конструирования решателей из программных единиц,
- средства проверки БЗ и результатов работы СБЗ на эталонных наборах решенных задач.

Инструментарий для развития баз знаний (в онтологической среде) делает главный вклад в жизнеспособность СБЗ.

Для построения СБЗ, способных выдавать аргументированный совет или решение, принципиально важны адекватные современному уровню базы знаний (содержащие предметные знания). Формирование БЗ происходит или вручную (с помощью инструмента редактирования под управлением онтологии *Ed(KnOnt)*), или индуктивно (с помощью инструмента обучения – извлечения знаний из базы прецедентов под управлением онтологии *TrnSft(KnOnt, PrecB)*), или программными онтологическими интерпретаторами текстов, описывающих предметные знания.

Такой онтолого-ориентированный подход к формированию БЗ (*KB(KnOnt)*) обеспечивает ее декларативное представление.

Развиваемость баз знаний позволяет надеяться на получение «эталонной» базы знаний $k^* \in K_n(X, Y)$, от актуальности и качества которой зависит успех применения СБЗ (для получения объяснения результата $y \in Y$ при любых входных данных $x \in X$).

Ввиду важности этого аспекта следует всерьёз рассматривать только такие СБЗ, которые интегрированы с системой управления ее БЗ (СУБЗ).

Актуальность знаний (базы знаний) достигается адаптируемостью (интерактивным изменением базы знаний) и адаптивностью (возможностью применять средства индуктивного формирования знаний по обновляемому набору прецедентов).

Определены функции системы управления БЗ (накапливать прецеденты из реальной практики и классифицировать их для выбора тех, с помощью которых производить монотонное улучшение «меры правильности» БЗ). Для этого принципиально важны программные инструменты оценивания их качества (AssSft). *Среда развития СБЗ* должна предоставлять средства оценивания БЗ по эталонным архивам решенных задач (онтологии которых SitOnt зафиксированы) и средства индуктивного формирования баз знаний (или фрагментов БЗ) по выборкам решенных задач, сформированным по той же онтологии. Это дает возможность иметь всегда актуальную версию БЗ:

$$KB_{ver} = KB_{ver-1}(KnOnt) * (TrnSft(KnOnt, PrecB(SitOnt)) | (KBEd(KnOnt)) * AssSft(KnOnt, SamplB(SitOnt))).$$

(Здесь символом «*» указаны применяемые в процессе развития СБЗ инструменты для усовершенствования БЗ.)

Актуальность версии БЗ означает ее близость к совокупному знанию специалистов и экспертов ПрОбл. От СБЗ ожидают, что они будут оказывать поддержку на таком уровне.

Средства оценивания баз знаний по эталонным архивам дают принципиально другой уровень качества по сравнению с традиционными средствами оценивания баз знаний – средствами контроля формальных свойств построенной БЗ и привлечением экспертов для оценки решений, предлагаемых системой.

Модель жизнеспособной СБЗ есть представление всех компонентов, участвующих в процессе ее функционирования и в процессе ее адаптации, и связей этих компонентов друг с другом.

$$KBS-Arch = \langle \bigcup_{i=1, N1} (KB_i(Kn_iOnt, Dict)), \bigcup_{j=1, N2} (Solv_j(KnOnt, SitOnt, Agree)), \bigcup_{m=1, N3} (U_{i_m}(SitOnt)), \bigcup_{p=1, N4} (DB_p(SitOnt, Dict)), \bigcup_{k=1, N5} (Soft_k) \bigcup_{i=1, N6} (TrnSft_i(Kn_iOnt)) \bigcup_{i=1, N7} ((KB_iEd(Kn_iOnt)) \bigcup_{i=1, N1} (AssSft(Kn_iOnt))). N6 \leq N1, N7 \leq N1.$$

Таким образом, к ключевым принципам методологии создания жизнеспособных систем (проектирование декларативных БЗ по онтологии знаний и проектирование архитектуры решателей задач в соответствии с онтологией знаний и данных) добавляется создание (под)системы управления качеством всех БЗ с инструментами для оценки БЗ, для накопления прецедентов и формирования базы эталонов, обучения базы знаний с помощью прецедентов.

Компонентами конфигурации СБЗ являются все ее программные и информационные компоненты, документы, предопределяющие ее конструирование, СУБЗ, наборы тестовых данных и отчетов.

Методы обеспечения трех принципиальных способностей ИИ (способности объяснять гипотезы, способности вести диалог, направленный на отсеечение неподходящих гипотез, и способности развиваться, расширяя и уточняя знания) могут быть разными, но есть один ключ к решению всех – формирование интерпретируемых (декларативных) баз знаний, понятных человеку и вычислительной машине.

Проектирование архитектуры СБЗ, интегрированной с подсистемами для управления качеством БЗ – ключевой принцип жизнеспособности.

РАЗРАБОТКА РЕШАТЕЛЕЙ СБЗ С ПОВТОРНЫМ ИСПОЛЬЗОВАНИЕМ

Онтолого-ориентированный (или онтологический) решатель Solv (KnOnt^U SitOnt^U Agreeem) в составе СБЗ должен быть способен выдвигать гипотезы по результату сопоставления входной информации (об объекте) предложениям из БЗ KB(KnOnt, Dict) на основе онтологических соглашений Agreeem. Онтологические соглашения для решения задач содержат стратегии принятия решения, правила выстраивания зависимости понятий-причин с понятиями-следствиями в процессе рассуждений и правила соотнесения данных и знаний (с учетом ограничений на интерпретацию смысла терминов).

Задача решателя – предложить путем «рассуждения» на основе соглашений одно или несколько обоснованных решений-гипотез, согласованных с Базой знаний KB_{iu} (KnOnt_{iu}) и входной информацией.

Его логический вывод, в отличие от логических выводов, реализованных как исчисления, создается как алгоритм последовательного подтверждения или опровержения подцелей, соответствующих элементам формализованных знаний (для задач диагностики, прогнозирования, распознавания класса).

Логический вывод (решателя) для задач планирования, проектирования, управления – последовательный переход от целей к подцелям, а от них – к обеспечивающим их элементам или действиям, которым можно поставить в соответствие исходные условия задачи. Полная совокупность подцелей или соответствующих им элементов становится гипотезой.

Так как же в программной инженерии проектировщик-архитектор решателя, создаваемого по онтологии, проектирует его из программных единиц (ПрЕд). Естественно, выделяются единицы разных типов, например, для задач анализа: ПрЕд для вывода или промежуточного заключения и ПрЕд для поиска фактов и ПрЕд для объяснения ответа. ПрЕд для вывода или промежуточного заключения работает с теми видами связей (между элементами БЗ), которые есть в онтологической модели. Она делает промежуточные заключения процесса логического вывода. ПрЕд для поиска фактов получает из входного документа значения указанных наблюдений и дает ответ: {данные соответствуют; данные противоречат; данные отсутствуют}. ПрЕд для объяснения ответа записывает найденные, противоречащие или отсутствующие факты «рядом» с преследуемыми подцелями, а подтвержденные подцели – с заключениями. В зависимости от того, «спланировано» ли объяснение как отчет с уникальной структурой или как более универсальный отчет об анализе всех связей понятий, относящихся к решаемой задаче или к анализируемому типу знаний, реализация компонентов решателя, формирующих объяснение, специализирована (под пользователя) или универсальна (по связям из онтологии знаний).

Решатель задачи некоторого типа становится ядром оболочки для построения множества СБЗ или ИС, компонуемых с разными БЗ или их версиями. И это самая очевидная возможность повторного использования обсуждаемого подхода.

Другие «границы» и возможности повторного использования открывает анализ общих свойств классов задач. В построенной многоуровневой классификации задач задачи общего класса – выше в иерархии, а ниже – более частные. Два вида повторного использования применимы при построении решателей задач, расположенных «внизу» иерархии на «фоне» автоматизации задач «выше» по иерархии. Задачи специализируются с учетом свойств предметных областей. Это связано с моделью знаний, в терминах которой ставится (формулируется) задача. Подразумеваемая «наверху» модель обобщает самые общие понятия и связи. Чем ниже по иерархии задача, тем ее модель (и постановки задач) ближе к реальности. Приближенные к реальности задачи анализа связаны с областями, где процессы связаны между собой посредством причинно-следственных отношений, это более частные постановки. Обобщенные модели знаний предметных областей важны для постановок связанных задач и определения методов решения задач.

Обобщающая модель знаний о процессах (см. выше) объединяет разные виды связи понятий, обеспечивая потенциал повторного использования их обработки в разных задачах. Системный анализ интеллектуальной деятельности и анализ свойств предметных областей и задач, связанных с анализом динамических ситуаций, показывают, что знания обычно содержат:

- связи вариантов признаков для класса ситуаций (процессов);
- свойства классов процессов;
- варианты изменения свойств под влиянием известных факторов;
- варианты последовательностей периодов развития процесса;
- варианты изменения диапазона значений признаков в каждом периоде развития;
- варианты реакции признаков на события;
- законы изменения свойств с течением времени и т. п.

Но обобщающая модель знаний об аномальных процессах [22] содержит уже и причины аномалий (обычно это события), атрибут уровня опасности, элементы для нечеткости или ранжирования:

$$f_{in} = [\{ \text{factor}_m \mid \text{event}_m \} +] [\text{CriticRate} +] [\{ \{ \text{factor}_{nj} \} + \} \{ (\text{period}_{ij} + [\text{StageCriticality} +] \text{interval}_i) + \{ \text{signName}_{jk}, \text{signValuesRange}_{ijk} [+ \{ \text{event}_u + \text{new-range}_i \text{ of } \text{signValues}_{jkju} + \text{delay}_{kju} \}^{n2}] + \text{signMinQaunt}_{ji} \} \}]^{n1} .$$

Это дает основу для точных постановок большего числа более частных задач. Решатели их соответственно будут сложнее, так как понятий и отношений больше.

Но разработчики СБЗ могут применить более простой способ, применив более общие «решения». Применить абстрактное решение для более частных задач означает сознательно огрубить знания (и процесс).

Популярный пример огрубления: модель знаний в симптом-чекерах. Модель медицинской диагностики сложна, но она является частным случаем модели распознавания нарушений по набору признаков. По общей модели связей понятия «нарушение» с понятием «признак» строятся симптом-чекеры – системы, претендующие на роль помощника врача в дифференциальной диагностике. (Иногда явные БЗ строятся усилиями экспертов.)

Такого рода сведение более частной модели к более общей не соответствует медицинской литературе, так как приближенная к реальности модель диагностики учитывает и описывает развивающееся заболевание [23].

Для целей такого повторного использования готового решения происходит трансформационное усечение модели. Например, в БЗ диагностики

$$K_n = \{ D_{sk} = \{ \text{signName}_{jk}, \text{signValuesRange}_{ijk} \}^{n1} \}$$

вместо полной модели

$$K_n = \{ f_{in} = [\{ \text{factor}_m \mid \text{event}_m \} +] [\text{CriticRate} +] [\{ \{ \text{factor}_{nj} \} + \} \{ (\text{period}_{ij} + [\text{StageCriticality} +] \text{interval}_i) + \{ \text{signName}_{jk}, \text{signValuesRange}_{ijk} [+ \{ \text{event}_u + \text{new-range}_i \text{ of } \text{signValues}_{jkju} + \text{delay}_{kju} \}^{n2}] + \text{signMinQaunt}_{ji} \} \}]^{n1} .$$

При «усечении» модели важно, по возможности, не потерять контент. Например, в медицинской БЗ по усеченной модели можно описывать как отдельные классы D_{sk} не только разные болезни в самой яркой стадии проявления, но и состояния (совокупности достигаемых изменений) на разных этапах болезни.

Можно «отсекаемые» понятия «сводить» к оставшимся, а именно: события и прочие факторы представить как признаки (с бинарными значениями или значением-датой), например, event_m – к $\langle \text{signName}_m, \text{«да»} \rangle$, динамические признаки – свести к спектру признаков, именованных по времени проявления, например, $(\text{period}_{ij}, \text{signName}_{jk}, \text{signValuesRange}_{ijk})$ – к $\langle \text{signName}_{jk}^{\text{period}_{ij}}, \text{signValuesRange}_{ijk} \rangle$.

Степень повторного использования общего решателя зависит от его внутреннего архитектурного решения. Можно применить архитектуру и часть компонентов абстрактного решения

(перебор гипотез, стратегию), заменив компоненты проверки значения компонентом проверки динамики значений на указанном отрезке развития.

Следующий вид повторного использования применим при решении задач одного класса одна «проблема», любое «место» в иерархической модели в разных конкретных предметных областях. Суть: проблемно-ориентированный решатель становится оболочкой для построения множества СБЗ для частных случаев с настройкой (адаптацией) на конкретную предметную область.

В каждой конкретной предметной области есть свой терминологический словарь понятий и множества названий для них. Предлагаются термины общего решения соотнести с предметными, но подменять их только для GUI-компонентов как замену отображаемому слову.

В архитектуре оболочки появляется дополнительный компонент – Таблица соответствия общих понятий с предметными (рисунок 2).

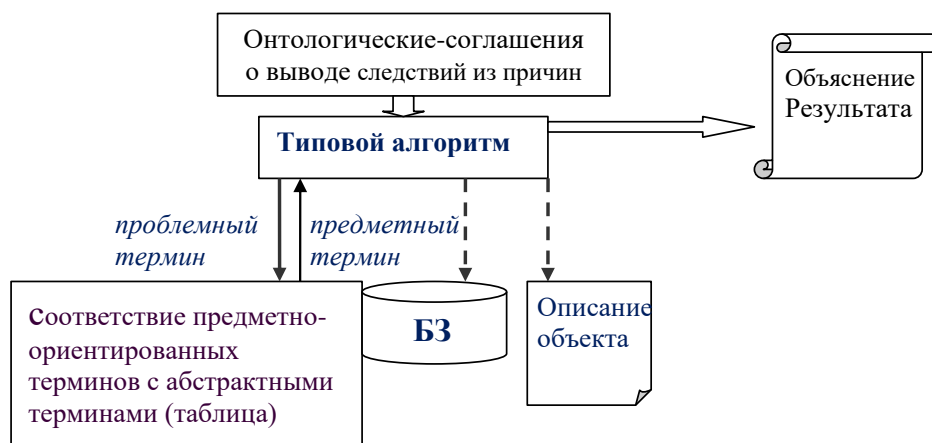


Рис. 2 Концепция системы адаптации проблемно-ориентированной оболочки к ПрОбл.

Чтобы специалисты конкретной предметной области могли понимать (или формировать) базы знаний и данных в понятных им терминах («разметке», семантическом шаблоне), их средства редактирования и визуализации должны быть специализированы. Поэтому применяется метод адаптации универсальных онтологий к предметно-ориентированным, специализированным.

В процессе работы (в проблемных терминах) базовым (типовым) алгоритмом в момент передачи запросов на поиск и на запись проблемный термин заменяется предметно-ориентированным для работы со специализированными материалами. Такое повторное использование обеспечивает снижение трудозатрат для СППР с моделями знаний, приближенными к реальности.

Решатель проектируется из программных единиц (ПрЕд, $Unit_m$) разных типов.

ПрЕд для поиска фактов получает значения наблюдений, список условий (события и прочие факторы), ищет во входном описании (документе) факты с такими названиями, сравнивает и дает ответ: {данные соответствуют; противоречат/ не соответствуют; отсутствуют}.

ПрЕд для вывода или промежуточного заключения запрашивает и получает информацию о подтверждении или отрицании элементов отношений или их цепочек, прописанных в Кп (в частности, ответ о соответствии условию или не соответствии). Программная единица, обрабатывая такие виды связей между элементами информации, делает промежуточные заключения процесса логического вывода.

ПрЕд, которые делают некоторое заключение о соответствии подмножества фактов – знаниям некоторого типа (причинно-следственных или других структурных связей между понятиями), реализуют те или иные соглашения из Agreeм.

Решатель, создаваемый из ПрЕд, программируемых по такому принципу, – онтологический. Особенности, принципиально отличающие его от других программ-решателей, таковы:

- он является алгоритмом, а не набором продукций (или исчислением предикатов);
- алгоритм формируется по онтологии и не зависит от самих БЗ, что соответствует современному подходу к разработке интеллектуальных систем;
- в нем «естественно» реализуется подробное и понятное специалисту объяснение предлагаемого решения.

КОНЦЕПЦИЯ ТЕХНОЛОГИЧЕСКОЙ СРЕДЫ ДЛЯ СОЗДАНИЯ И РАЗВИТИЯ СБЗ

Проектирование архитектуры решателей задач в соответствии с типом решаемой задачи и набором соответствующих связей (структурных и причинно-следственных) между сущностями и сборка решателя из *повторно используемых* программных единиц совокупно обеспечивают прозрачность решателей задач и их сопровождаемость.

Повышению прозрачности способствует декларирование решателя (как программного компонента СБЗ): явное специфицирование его внешних свойств и потенциально заменяемых собственных компонентов). Это особенно важно в условиях коллективной разработки СБЗ и их развития.

Следовательно, важна инструментальная поддержка декларирования программных компонентов, нужны средства создания заготовок исходных кодов по декларациям, средства кодирования новых ПрЕд, средства каталогизации ПрЕд как потенциально *повторно используемых компонентов* и средства интеграции готовых и новых ПрЕд в новые решатели или их новые версии.

Минимально необходимая среда разработки СБЗ представляет собой:

- онтологию ПрОбл (включая базу терминов);
- сгенерированные редакторы Информации (БЗ и баз данных);
- инструменты декларирования решателя и ПрЕд;
- среда разработки программ (IDE) для кодирования ПрЕд.

Более развитая среда разработки дополняет *минимальную* множеством подготовленных и проверенных Информационных Компонентов (БЗ, архивов и баз данных); библиотеками протестированных ПрЕд, их обрабатывающих; множеством готовых решателей задач, созданных из подготовленных ПрЕд.

В состав *полной среды разработки и развития СБЗ* включаются средства оценивания БЗ и средства индуктивного формирования БЗ и другие инструменты усовершенствования баз знаний. Методология проведения тестирования решателя включает проверки правильности выдачи гипотез, направленного диалога и точной аргументации. На этапе аттестационного тестирования решателя задачи прогнозирования оценка правильности гипотез проверяет выдачу каждого возможного состояния из тестовой модели знаний, которое может (согласно этой тестовой модели знаний о процессе) наступить после состояния, указанного во входных данных через указанное время с учетом событий и факторов. На этапе аттестационного тестирования решателя задачи диагностики оценка правильности гипотез проверяет выдачу каждого диагноза из тестовой модели, который (согласно тестовой модели знаний о процессе) имеет проявления, указанные во входных данных за указанный отрезок времени. При этом в рамках единой среды специализированные инструменты обеспечения качества используют общее подмножество онтологических ПрЕд, что и решатель (так как ориентированы на те же виды связей), реализуя принцип повторного использования.

Среда разработки и развития обычно обеспечивает:

- поддержку обновления знаний;
- поддержка изменения компоновки онтолого-ориентированного решателя;
- поддержка усовершенствования интерфейса пользователя в связи с изменением функций;
- поддержка усовершенствования интерфейса эксперта (при расширении онтологии);

- поддержка кодирования новых версий программных единиц онтолого-ориентированного решателя или поддержка изменения кода ПрЕд такого Решателя.

Работа с БЗ по их улучшению или расширению и замена компонентов решателя представляет собой изменения в декларативных компонентах СБЗ. Построение СБЗ с помощью инструментальных онтологических сред развития принципиально увеличивает их жизнеспособность за счет увеличения роли и доли средств управления (внесение изменений в декларативные компоненты) по отношению к средствам сопровождения (внесение изменений в исходный код).

КОНЦЕПЦИЯ ОБЛАЧНОЙ ТЕХНОЛОГИИ РАЗРАБОТКИ СБЗ

Предложена технология разработки современных СБЗ, нацеленная на многообразие повторного использования готовых решений [24]. В отличие от ранних технологий создания экспертных систем, где предусмотрен многократный возврат к начальным этапам для добавления новой порции знаний-правил к проверенному на тестах прототипу, она обеспечивает планируемый линейный путь построения (за счет использования готовых решений) и коллективно-параллельную работу разных участников, снижает технические риски (риск «не построить нужную систему»).

Системный анализ позволяет декомпозировать проблему на совокупность интеллектуальных и обычных подзадач. Автоматизация каждой из них осуществляется с учетом последующей интеграции в единый комплекс или систему. Определение типа задачи открывает возможность выбора универсальной онтологии знаний (типы понятий и отношений для решения такой задачи) или построения онтологии, уточненной с учетом свойств ПрОбл. Возможность определить виды экспертных задач в составе поставленной проблемы и использовать готовые для них решатели или их компоненты и другие элементы конфигурации открывает перспективу использования готовых решений.

Предлагаемая технология предусматривает такую последовательность работ: 1) онтология (знаний и данных); 2а) и 2б) параллельная разработка нового решателя по онтологии и БЗ. Работа 2а – это коллективный процесс: каждый участник формирует свой фрагмент или одни формируют, другие уточняют, добавляют, проверяют. Работа 2б – типичный процесс модульной разработки ПС. Каждый модуль (среди которых есть онтолого-ориентированные) декларирован. Управляющий граф (как декларативный компонент) обеспечивает прозрачность вновь создаваемого Решателя, а декларирование и каталогизация созданных программных компонентов – удобство их использования в последующих разработках в этой ПрОбл.

Технология обеспечения качества БЗ включает процессы проверки их соответствия эталонным решениям и непрерывного улучшения баз знаний по новым фактам, поступающим из реальной практики.

Целям поддержки всех видов работ представленной технологии (с доступностью, прозрачностью и повторной используемостью его результатов) и коллективности процесса «ответают» облачная платформа и среда IASPaas (<http://iaspaas.dvo.ru>) [25].

Инструментарий среды построения и развития СБЗ – это редактор онтологии, генератор редакторов баз знаний и данных, редактор деклараций ПрЕд с генерацией заготовки ПрЕд и загрузкой байт-кода, редактор декларативного решателя, средства кодирования новых ПрЕд (на языке Java) с использованием библиотеки операций доступа и обработки Информационных Ресурсов IASPaas (ИнфРес). Инструментарий позволяет управлять размещением готовых ПрЕд в папках хранилища для обеспечения их поиска и повторного использования в процессе создания программных решателей или их новых версий.

Стратегия накопления и хранения позволяет формировать необходимую и достаточную среду разработки и развития СБЗ – минимальную (как на рисунке 3), типичную или полную. Минимально важны онтология ПрОбл, база терминов, сгенерированные IASPaas редакторы ИнфРес, штатные редакторы деклараций Решателя и ПрЕд, а также JavaIDE.

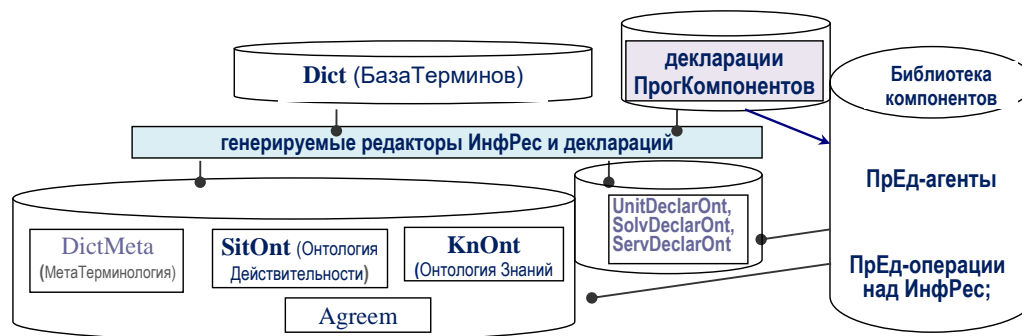


Рис. 3 Модель специализированной технологической среды.

Такая IASaaS-среда разработки и развития *СБЗ* формируется из накопленных в облачном хранилище единиц разных типов. В зависимости от роли *СБЗ* для пользователей (прототип, программа для одного профессионального коллектива или для применения в профессиональной среде) будет «собрана» *среда разработки и развития СБЗ* – минимальная или полная.

Технология разработки и развития *СБЗ* реализуется с помощью инструментального комплекса – облачной инструментальной среды развития на IASaaS [26].

В гибридных ИС для подключения внешнего микросервиса, решающего дополнительную задачу для тех же входных данных, используется модуль-коннектор. У него стандартные задачи: прочитать в декларации (спецификации) этого внешнего сервиса список имен требуемых данных, найти значения этих данных во входном документе, составить *put*-запрос с указанным URL, послать его. Если микросервис не интерактивный, то: дождаться ответа, выбрать (указанные в декларации) фрагменты для объяснения, добавить их в предусмотренную подструктуру итогового объяснения. Объяснения гипотез от прочих AI-модулей могут содержать характеристики применяемой модели, характеристики обучающего материала (если известны) и перечисления взятых на вход данных.

ОЦЕНКА ПРАКТИЧЕСКОГО ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИИ РАЗРАБОТКИ СБЗ

Технология IASaaS была многократно апробирована для разработки и развития *СБЗ* и порталов знаний.

Процесс формирования инфраструктуры медицинского портала включал разработку словаря терминов (около 15 000); адаптацию онтологии диагностики к онтологии медицинской диагностики, развитие ее до модели течения заболеваний по периодам динамики, разработку семантической структуры («онтологии») современной истории болезни и медицинской карты (около 200 понятий), онтологии фармакологического справочника (около 30 понятий), онтологии схем лечения заболеваний (около 80 понятий).

Процесс разработки управляемых интеллектуальных сервисов в предметной области «медицина» включал разработку форматов сложно структурированных отчетов о проведенном анализе для объяснения гипотез (для задач диагностики, лечения, прогнозирования) и алгоритмов их формирования на основе интерпретируемых знаний для решаемых задач.

При эксплуатации медицинского портала специалисты по эпидемиологии сформировали знания о диагностике разных форм геморрагической лихорадки на разных этапах, апробировали возможности программной поддержки ранней диагностики с этими БЗ на реальных примерах пациентов из Приморского края. Специалисты по неврологии сформировали терминологическую базу симптомов для оцифровки имеющихся архивов и формирования знаний о диагностике, лечении и прогнозе выздоровления. Специалисты по мукополисахаридам сформировали знания, апробировали их на реальных примерах пациентов из разных стран и приступили к уточнению знаний. Созданы терминологическая база симптомов кардиологических патологий, база знаний о влиянии патологических состояний и прочих факторов

на возникновение и развитие кардиологических заболеваний и их осложнений и многие другие базы. Специалисты формируют клиническую картину заболеваний с десятками симптомов, как правило, динамических. Базы знаний о схемах лечения были сформированы для нескольких кардиологических заболеваний и ЖКТ [27].

Для компонуемых СБЗ проводилось тестирование в соответствии с методологией. На «уровне компонентов» испытывались решатели диагностики, лечения, оценки рисков (правильность выдачи гипотез, направленный диалог и аргументация в соответствии со знаниями). Отдельно оценивались БЗ. Каждая СППВР проверялась «на уровне» целой сборки. В единую систему управления качеством интегрированы средства проверки знаний для медицины, средства пополнения знаний из доверенных источников и средства извлечения знаний на основе документов из практики работы врачей.

Проведен сравнительный анализ двух подходов на примере поддержки практической интеллектуальной деятельности в медицине. Оценивались возможности обеспечения точности и правильности решений и возможности генерации советов. Сравнивались трудозатраты на автоматизацию рабочего места и поддержание актуальности СБЗ.

Первый анализ сопоставлял IASaaS-интегрированных ХАИ и других доступных сервисов: пять наиболее свежих медицинских диагностических IASaaS-сборок и 5 медицинских интернет-симптом-чекеров.

На десяти случайно выбранных из архива истории болезни (ИБ) пятью чекерами верный диагноз выдавался в топ-3 минимум 2, максимум 4 раза, а в топ-10 от 8 до 9 раз. Для тех же ИБ IASaaS-сервисами верный диагноз отправили в топ-3 минимум 7 раз, а в топ-10 – во всех случаях.

Второй анализ сопоставлял облачную реализацию диагностических СБЗ с разделением компетенций со случаем «традиционной» разработки. До создания облачной платформы и развития онтологического подход затраты выглядели так.

На разработку первой версии сервиса диагностики (с БЗ на правилах) для одной группы заболеваний (терапевта, офтальмолога и гастроэнтеролога) требовалось в среднем 20 чел-мес (в том числе выбор единой терминологии для знаний и данных), тестирование с отладкой – 8. На расширение еще одним заболеванием тратился 1 чел-мес, а новое тестирование с отладкой – 2 чел-мес.

В ситуации облачной реализации СБЗ с отделенной декларативной БЗ и с разделением компетенций для одной группы заболеваний (терапевта, гастроэнтеролога, кардиолога) требовалось в среднем 10 чел-мес + 2 чел-мес на единый терминологический словарь, диагностический движок – 8 чел-мес, включая его тестирование, тестирование БЗ с отладкой – 4 чел-мес. На расширение БЗ еще одним заболеванием – 1 чел-мес, и новое тестирование с отладкой – 1 чел-мес.

Тогда, если рассмотреть 5 условных медицинских коллективов, работающих по разным медпрофилям, пять команд разработчиков систем диагностики потребуют при традиционной разработке $(20 + 8) \times 5 = 140$ чел/мес. При автоматизации деятельности через «облачные» системы с декларативными БЗ и универсальными решателями: $10 \times 5 + (2 + 1 \times 4) + 8 + 4 \times 5 = 84$ чел-мес. Экономия за счет того, что не 5 решателей и терминологических словарей, а единый решатель и наращиваемый (1×4) терминологический словарь.

Оценочное число специалистов, участвующих ежегодно в развитии (поддержании актуальности и усовершенствовании) СБЗ, при традиционном подходе – 5 чел. на усовершенствование баз знаний диагностики и 5 чел. на обработку. В новой технологии с декларативной БЗ $(5 + 1)$ чел (экономия за счет единого движка).

В облачной технологии СБЗ с отделенной декларативной БЗ для многопользовательских СППР ожидается экономия в разы за счет единого центра обновления знаний и проверки на накапливаемой БазеЭталонов. Общие базы знаний по всем классам задач и профилям (все ее модули размещаются на защищенных центральных серверах отрасли), там же устанавливаются универсальные решатели и система управления БЗ и накапливается единый архив правильно решенных задач

по каждому профилю. Сборка сервисов (СБЗ) из базы знаний и соответствующих решателей осуществляется на рабочих местах в зависимости от потребностей специалистов. Имея доступ в Интернет, коллектив экспертов каждого профиля (в идеале – специально выделенные высококвалифицированные специалисты) управляет качеством баз знаний посредством «облачно» доступных инструментов.

На сегодняшний день медицинские специалисты различных направлений используют средства медпортала для развития формализованных знаний и их использования.

ЗАКЛЮЧЕНИЕ

В проведенных исследовании и разработке методологии создания интеллектуальных систем на основе онтологий получены следующие результаты.

1. Модель жизнеспособной системы поддержки решения задач интеллектуальной деятельности в рамках их постановок.

2. Метод непрерывного улучшения баз знаний.

3. Принципы формирования инструментальной среды развития СБЗ и технология коллективного создания программных систем поддержки деятельности специалистов на основе формализованных баз знаний и их развития, преимущественно за счет развития БЗ и обеспечения их качества.

Совокупные ключевые принципы методологии создания жизнеспособных систем – проектирование декларативных БЗ по онтологии знаний, проектирование архитектуры решателей задач из онтологических ПрЕд, создание подсистемы управления качеством всех БЗ. Эти принципы лежат в основе логики и механизмов объяснения результатов, возможности специалистам просматривать наполнение баз знаний для оценивания полноты, возможности работать с реальными данными из ПрОбл; механизмов перманентного усовершенствования баз знаний и регулярного их оценивания, – наиболее ценных с точки зрения обеспечения доверия специалистов к системам искусственного интеллекта.

СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. Breivold H. P., Crnkovic I., Eriksson P. J. Analyzing software evolvability // Computer Software and Applications. 2008. COMPSAC'08. 32nd Annual IEEE International. IEEE. 2008. Pp. 327–330.
2. Izurieta C., Bieman J. M. A multiple case study of design pattern decay, grime, and rot in evolving software systems // Software Quality Journal. 2013. Vol. 21. № 2. Pp. 289–323.
3. Горшков С. В. Онтологическое моделирование предприятий: методы и технологии. Екатеринбург: Изд-во Урал ун-та, 2019. [[Gorshkov S.V. Ontological Modeling of Enterprises: Methods and Technologies. Ekaterinburg: Ural University Publishing House, 2019. (In Russian).]]
4. Islam M., Katiyar V. Development of a software maintenance cost estimation model: 4th GL perspective // International Journal of Technical Research and Applications. 2014. Vol. 2. Issue 6. Pp. 65–68.
5. Musen M. The Protégé Project: a look back and a look forward // AI Matters. 2015. Jun; No. 1 (4). Pp. 4–12.
6. Müller L., Gangadharaiah R. et. al. An open access medical knowledge base for community driven diagnostic decision support system development // BMC Medical Informatics and Decision Making. 2019. Vol. 19. Pp. 1–7.
7. Уотермен Д. Руководство по экспертным системам/Пер. с англ. М.: Мир, 1989. [[Waterman D. Guide to Expert Systems: Trans. from English Mjscow: Mir, 1989. (In Russian).]]
8. Хейес-Рот Ф., Уотермен Д., Ленат Д. Построение экспертных систем. М.: Мир, 1987. [[Hayes-Roth F., Waterman D., Lenat D. Construction of Expert Systems. Moscow: Mir, 1987. (In Russian).]]
9. Clancey W. J. Heuristic classification // Artificial Intelligence. 1985. No. 27. Pp. 289–350.
10. Финн В. К. Об интеллектуальном анализе данных // Новости искусственного интеллекта. 2004. № 3. С. 3–18. [[Finn V. K. About intelligent data analysis // News of Artificial Intelligence, No. 3, 2004, pp. 3–18. (In Russian).]]
11. Воронцов К. В. Математические методы обучения по прецедентам. 2012. URL: [http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_\(курс_лекций,_К.В.Воронцов\)](http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций,_К.В.Воронцов)). [[Vorontsov K.V. Mathematical Teaching Methods Based on Precedents. 2012. (In Russian).]]
12. Финн В. К. Индуктивные методы Д. С. Милля в системах искусственного интеллекта // Искусственный интеллект и принятие решений. 2010. № 3. С. 3–21. [[Finn V. K. Inductive methods of D.S. Mill in artificial intelligence systems // Artificial Intelligence and Decision Making. 2010. No. 3, pp. 3–21. (In Russian).]]
13. Шахмаметова Г. Р., Христовуло А. Д., Береговая С. П. Анализ эндокринологических данных на основе моделей классификации // Системная инженерия и информационные технологии. 2022. Т. 4. № 2 (9). С. 30–36. [[Shakhmametova G. R.,

Christodulo A. D., Beregovaya S. P. Analysis of endocrinological data based on classification models // System Engineering and Information Technologies. 2022. Vol. 4, No. 2 (9), pp. 30-36. (In Russian).]]

14. Юсупова Н. И., Нургаянова О. С., Зулкарнеев Р. Х. Формализация этапов риск-анализа в СППР с учетом оценок клинических рисков при бронхолегочных заболеваниях // Системная инженерия и информационные технологии. 2023. Т. 5. № 1 (10). С. 11–24. [[Yusupova N. I., Nurgayanova O. S., Zulkarneev R. Kh. Formalization of the stages of risk analysis in DSS taking into account assessments of clinical risks for bronchopulmonary diseases // System Engineering and Information Technologies. 2023. Vol. 5, No. 1 (10), pp. 11-24. (In Russian).]]

15. Брекоткин В. Е. и др. Информационная технология подбора компонентов сложных технических систем на основе интеллектуального анализа их желаемых и фактических характеристик из распределенных баз данных // Системная инженерия и информационные технологии. 2022. Т. 4. № 2 (9). С. 13–23. [[Brekotkin V. E. et al. Information technology for selecting components of complex technical systems based on intelligent analysis of their desired and actual characteristics from distributed databases // System Engineering and Information Technologies. 2022. Vol. 4, No. 2 (9), pp. 13-23. (In Russian).]]

16. Усманова А. Р., Валиахметова Ю. И. Особенности метода поиска с запретами для задачи упаковки // Системная инженерия и информационные технологии. 2022. Т. 4. № 2 (9). С. 37–42. [[Usmanova A. R., Valiakhmetova Yu. I. Features of the tabu search method for the packaging problem // System Engineering and Information Technologies. 2022. Vol. 4, No. 2 (9), pp. 37-42. (In Russian).]]

17. Тельнов Ю. Ф. Интеллектуальные информационные системы. М.: МЭСИ, 2004. [[Telnov Yu. F. Intelligent Information Systems. Moscow: MESI, 2004. (In Russian).]]

18. Джексон П. Введение в экспертные системы. М.: Вильямс, 2001. [[Jackson P. Introduction to Expert Systems. Moscow: Williams, 2001. (In Russian).]]

19. Павлов С. Н. Системы искусственного интеллекта: уч. пос. В 2-х частях. Ч. 1. Томск: Эль Контент, 2011. [[Pavlov S. N. Artificial Intelligence Systems: textbook. In 2 parts. Part 1. Tomsk: El Content, 2011. (In Russian).]]

20. Клещев А. С., Шалфеева Е. А. Постановки практически полезных задач интеллектуальной деятельности // Дальневост. матем. журн. 2016. № 16(1). С. 44–61. [[Kleshchev A. S., Shalfeeva E. A. Statements of practically useful problems of intellectual activity // Far East. Math. Magazine. 2016. No. 16 (1), pp. 44-61. (In Russian).]]

21. Грибова В. В., Шалфеева Е. А. Модель управления качеством баз знаний с оценением // Знания – Онтологии – Теории (ЗОНТ-2019): Мат-лы VII Международной конференции. 2019. С. 138–147. [[Gribova V. V., Shalfeeva E. A. Model of quality management of knowledge bases with assessment // Knowledge–Ontologies–Theories (ZONT-2019). Proceedings of the VII International Conference. 2019, pp. 138-147. (In Russian).]]

22. Грибова В. В., Шалфеева Е. А. Онтология диагностики процессов // Онтология проектирования. 2019. Т. 9. № 4(34). С. 449–461. [[Gribova V. V., Shalfeeva E. A. Ontology of process diagnostics // Design Ontology. 2019. Vol. 9, No. 4(34), pp. 449-461. (In Russian).]]

23. Грибова В. В., Петряева М. В., Окунь Д. Б., Шалфеева Е. А. Онтология медицинской диагностики для интеллектуальных систем поддержки принятия решений // Онтология проектирования. 2018. Т. 8. № 1 (27). С. 58–73. [[Gribova V. V., Petryaeva M. V., Okun D. B., Shalfeeva E. A. Ontology of medical diagnostics for intelligent decision support systems // Design Ontology. 2018. Vol. 8, No. 1 (27), pp. 58-73. (In Russian).]]

24. Грибова В. В., Клещев А. С., Крылов Д. А., Москаленко Ф. М., Тимченко В. А., Шалфеева Е. А. Базовая технология разработки интеллектуальных сервисов на облачной платформе IACPaaS. Ч. 1. Разработка базы знаний и решателя задач // Программная инженерия. 2015. № 12. С. 3–11. [[Gribova V. V., Kleshchev A. S., Krylov D. A., Moskalenko F. M., Timchenko V. A., Shalfeeva E. A. Basic technology for developing intelligent services on the IACPaaS cloud platform. Part 1. Development of a knowledge base and problem solver // Software Engineering. 2015. No. 12, pp. 3-11. (In Russian).]]

25. Грибова В. В., Клещев А. С., Москаленко Ф. М., Тимченко В. А., Федорищев Л. А., Шалфеева Е. А. Облачная платформа IACPaaS для разработки оболочек интеллектуальных сервисов: состояние и перспективы развития // Программные продукты и системы. 2018. Т. 31. № 3. С. 521–526.12. [[Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A., Fedorishchev L. A., Shalfeeva E. A. Cloud platform IACPaaS for the development of shells for intelligent services: status and development prospects // Software Products and Systems. 2018. Vol. 31, No. 3, pp. 521-526. (In Russian).]]

26. Грибова В. В., Шалфеева Е. А. Комплекс средств поддержки процессов разработки и сопровождения решателей для систем с онтологическими базами знаний // Информационные и математические технологии в науке и управлении. 2020. № 4(20). С. 34–43. [[Gribova V. V., Shalfeeva E. A. A set of tools to support the development and maintenance of solvers for systems with ontological knowledge bases // Information and Mathematical Technologies in Science and Management, 2020. No. 4(20), pp. 34-43. (In Russian).]]

27. Грибова В. В., Петряева М. В., Шалфеева Е. А. Облачный сервис поддержки принятия диагностических решений в гастроэнтерологии // Врач и информационные технологии. 2019. № 3. С. 65–71. [[Gribova V. V., Petryaeva M. V., Shalfeeva E. A. Cloud service for supporting diagnostic decision-making in gastroenterology // Doctor and Information Technologies. 2019. No. 3, pp. 65-71. (In Russian).]]

Поступила в редакцию 17 августа 2023 г.

МЕТАДААННЫЕ / METADATA

Title: Methodology for producing viable trustworthy artificial intelligence systems.

Abstract: A review of the results of the study of the viability of intelligent systems with declarative knowledge bases is presented. The aim of the study is to develop models, methods, and technologies for creating intelligent systems based on ontologies with a declarative representation of knowledge bases and mechanisms of evolution. To achieve this goal, it is important to establish

the range of tasks of intellectual activity solved based on formalized knowledge, formal characteristics (of these tasks), the knowledge models used, methods of their processing; propose a systematic approach to solving problems, establish a fundamentally important set of infrastructure components for building a SBZ and continuous adaptation to the development of the subject area. Therefore, the following tasks are set in this work. 1. Classification and specification of tasks of intellectual activity and analysis of methods for their solution. 2. Development of a viable system model based on declarative knowledge. 3. Development of an ontological approach to the formation of declarative knowledge bases for classes of intellectual tasks to increase the reusability of ready-made solutions. 4. Development of methods for evaluating and improving the quality of knowledge bases, semantic descriptions of input and output data and their ontologies. 5. Development of a method for declaring and constructing ontological solvers with reusability of ready-made solutions. 6. Development of cloud technology, infrastructure (in the form of tools) to support the development, maintenance, and development of SBZ components. The object of research is the process of developing systems to support solutions based on knowledge bases of tasks in responsible economic sectors and ensure their viability in the context of continuously evolving knowledge. The subject of research is the methods, technology and infrastructure for building and improving systems with knowledge bases. Research methods. The tasks are solved using system analysis, information theory, graph theory and semantic networks, set theory, object-oriented programming technology, cloud technologies, ontological engineering, and other methods of artificial intelligence.

Key words: problem of intellectual activity; declarative knowledge base; intelligent system, explanatory artificial intelligence, viable system, reuse.

Язык статьи / Language: русский / Russian.

Об авторе / About the author:

ШАЛФЕЕВА Елена Арефьевна

ФГБУН Институт автоматизации и процессов управления Дальневосточного отделения РАН, Россия.

Вед. науч. сотр. лаборатории интеллектуальных систем им. А. С. Клещева. Дипл. спец. по прикл. математике (Дальневост. гос. ун-т, 1989). Д-р техн. наук по мат. и прогр. обеспечению выч. систем, комплексов и комп. сетей (Ин-т автоматизации и процессов управления ДВО РАН, 2021). Иссл. в обл. обеспечения жизнеспособности интеллектуальных систем с декларативными базами знаний.

E-mail: shalf@dvo.ru

ORCID: <https://orcid.org/0000-0001-5536-2875>

URL: https://elibrary.ru/author_profile.asp?authorid=11682

SHALFEEVA Elena Arefyevna

Institute of Automation and Control Processes Far Eastern Branch of the Russian Academy of Sciences, Russia.

Leading researcher at the Laboratory of Intelligent Systems named after. A. S. Kleshev. Dipl. specialist in applied math (Far Eastern State University, 1989). Dr. Tech. Sciences in mathematical and software support of computer systems, complexes and computer networks (Institute of Automation and Control Processes, 2021). Research in the field of ensuring the viability of intelligent systems with declarative knowledge bases.

E-mail: shalf@dvo.ru

ORCID: <https://orcid.org/0000-0001-5536-2875>

URL: https://elibrary.ru/author_profile.asp?authorid=11682