

На правах рукописи



ШАЛФЕЕВА ЕЛЕНА АРЕФЬЕВНА

**МЕТОДЫ, МОДЕЛИ И ТЕХНОЛОГИЯ ОБЕСПЕЧЕНИЯ
ЖИЗНЕСПОСОБНОСТИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ
С ДЕКЛАРАТИВНЫМИ БАЗАМИ ЗНАНИЙ**

Специальность: 2.3.5 – Математическое и программное
обеспечение вычислительных систем, комплексов и
компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

доктора технических наук

Владивосток – 2021

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институте автоматике и процессов управления Дальневосточного отделения Российской академии наук

- Научный консультант: **Грибова Валерия Викторовна**,
доктор технических наук, с.н.с.
- Официальные оппоненты: **Фридман Александр Яковлевич**,
доктор технических наук, профессор,
ФГБУН Институт информатики и
математического моделирования
технологических процессов, Кольского
научного центра РАН, ведущий научный
сотрудник лаборатории информационных
технологий управления промышленно-
природными системами, г. Апатиты
- Пальчунов Дмитрий Евгеньевич**,
доктор физ.-мат. наук, доцент,
ФГБУН Институт математики
им. С. Л. Соболева СО РАН,
ведущий научный сотрудник лаборатории
теории вычислимости и прикладной логики,
г. Новосибирск
- Ярушкина Надежда Глебовна**,
доктор технических наук, профессор
ФГБОУ ВО «Ульяновский государственный
технический университет», ректор,
г. Ульяновск.
- Ведущая организация: ФГБОУ ВО «Уфимский государственный
авиационный технический университет», г. Уфа.

Защита состоится «20» декабря 2021г. в 10 часов 00 минут на заседании диссертационного совета Д 24.1.027.01 на базе Института автоматике и процессов управления ДВО РАН по адресу: 690041, г. Владивосток, ул. Радио, 5.

С диссертацией можно ознакомиться в библиотеке ИАПУ ДВО РАН и на сайте <http://www.iacr.dvo.ru/diss/dis4>

Автореферат разослан «__» _____ 2021 г.

Ученый секретарь
диссертационного совета Д24.1.027.01



Петрунко Н.Н.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы диссертации и степень разработанности темы.

На сегодняшний день актуальным для коллективов разработчиков программных систем является не только создание системы, отвечающей всем текущим требованиям к ней, но и реализация механизмов, обеспечивающих ее неизбежное последующее сопровождение, вызываемое изменениями знаний предметной области, условий эксплуатации или требований пользователей к функциональности и пользовательскому интерфейсу. Процесс сопровождения занимает значительную долю трудозатрат в процессе жизненного цикла программной системы: по оценкам специалистов – от 50 до 90% (Dehaghani S. M.H. & Najrahimi N. в 2013 г). Поэтому необходимо проектировать программные системы (ПС), закладывая механизмы для упрощения этого процесса.

Известные механизмы, используемые в программной инженерии для упрощения сопровождаемости ПС, таковы: архитектурные решения (разделение на слабо сцепленные компоненты с логически понятными функциями), декларативное представление компонентов ПС, средства автоматизации построения проектных моделей и автоматической генерации по ним фрагментов кода, средства связывания компонентов друг с другом, использование типовых архитектурных решений, стандартизация сервисов и интерфейсов (model-driven архитектуры и модельно-управляемый подход к созданию программных систем), 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» (Izurieta C. & Bieman J.M. в 2013.; Breivold H.P. et al в 2008; Herring C. & Kaplan, S. в 2000).

При автоматизации области со сложными и ответственными задачами принятия разных типов решений, где выработка рекомендаций возможна на основе формализуемых знаний, создают программные системы с дополнительным архитектурным компонентом – базой знаний (БЗ). Команда разработчиков при этом пополняется экспертами предметной области (ПрОбл) и когнитологами. Указанная специфика накладывает необходимость использования дополнительных специализированных механизмов обеспечения их жизнеспособности.

Существенный вклад в методы, технологии, инфраструктуру построения систем с базами знаний (СБЗ) внесли российские и зарубежные ученые и основанные ими школы: Артемьева И.Л.; Гаврилова Т.А., Голенков В.В., Грибова В.В., Еремеев А.П., Загорюлько Ю.А., Клещев А.С., Кузнецов О.П., Осипов Г.С., Рыбина Г.В., Соснин П.И., Стефанюк В.Л., Финн В.К., Ярушкина Н.Г., Дж. Джарратано, Г. Райли; П. Джексон; Лавилла С.; Musen M., Mortensen J., Noy N.; Hayes-Roth F.; Moore B.; Calero C.; Taboada M.; Emmanuel C.

Ogu, Adekunle, Y.; R. M.Sonar, R. Fikes and D. McGuinness; Müller L., Perry J. и многие другие.

Предлагаемые ими методы и технологии позволяют создавать базы знаний и системы, основанные на знаниях, называемые интеллектуальными системами (ИС), тестировать их и вводить в эксплуатацию.

Однако не уделено достаточного внимания методам и инструментам развития компонентов ИС в процессе их создания и особенно эксплуатации – как автоматически, так и каждым типом разработчика ИС (экспертом ПрОбл, когнитологом, программистом) независимо, но с обязательным автоматическим или автоматизированным согласованием допустимых действий.

Подход с универсальным представлением знаний в виде правил, обрабатываемых единой «машиной вывода», пригоден для областей, где число правил измеряется сотнями. Далее их создание и развитие особенно силами экспертов становится практически невозможным, что подтверждают многочисленные публикации. 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 и, 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. Другой опыт в отношении создания основы для сопровождения СБЗ – через библиотеки накапливаемых элементов решенных задач для повторного их использования – не стал эффективным, поскольку размещаемые туда семантические компоненты привязаны к частным задачам и не помогают при автоматизации новых задач и деятельности.

Собственный многолетний опыт разработки ИС в различных ПрОбл, множественные публикации (Müller L. & Perry J. в 2019; Islam M. & Katiyar V. в 2014; Dehaghani S.M.H. & Najrahimi N. в 2013; Musen M. в 2015; Selby R.W. в 2007; Горшков С.В. в 2019) показали, что объем знаний не позволяет формализовать базы знаний (БЗ) полностью за приемлемое время силами даже согласованного коллектива экспертов. Кроме того, знания динамичны: они расширяются и корректируются.

Для обеспечения возможности перманентного усовершенствования БЗ вышеперечисленным разработкам не хватает технологически обеспеченного отделения декларативных знаний от процедурных, независимости процессов разработки баз знаний и решателей, механизмов создания, модифицирования и оценивания БЗ, ориентированных на экспертов в течение всего процесса использования ИС без участия программистов и инженеров по знаниям.

Таким образом, несмотря на современные достижения в программной инженерии и инженерии знаний имеется необходимость разработки методов, моделей и технологии для обеспечения жизнеспособности интеллектуальных систем, которые могут продолжительное время быть полезными специалистам, решающим сложные и ответственные задачи.

Целью диссертационной работы является разработка моделей, методов и технологии создания интеллектуальных систем на основе онтологий с декларативным представлением баз знаний и механизмами эволюционирования.

Для достижения этой цели важно:

- установить спектр задач интеллектуальной деятельности, решаемых на основе формализуемых знаний, формальные характеристики (этих задач), используемые модели знаний, способы их обработки;
- предложить систематический подход к решению задач,
- установить принципиально важный набор компонентов инфраструктуры для построения СБЗ и непрерывной адаптации к развитию ПрОбл.

Поэтому в работе были поставлены следующие **задачи**.

1. Классификация и спецификация задач интеллектуальной деятельности и анализ методов их решения.
2. Разработка модели жизнеспособной системы, основанной на декларативных знаниях.
3. Развитие онтологического подхода к формированию баз декларативных знаний для классов интеллектуальных задач для повышения повторной используемости готовых решений.
4. Разработка методов оценивания и повышения качества баз знаний, семантических описаний входных и выходных данных и их онтологий.
5. Разработка метода декларирования и конструирования онтологических решателей с повторной используемостью готовых решений.
6. Разработка облачной технологии, инфраструктуры (в виде инструментальных средств) поддержки разработки, сопровождения и развития компонентов СБЗ.

Объект исследования – процесс разработки систем для поддержки решения на основе баз знаний задач в ответственных народнохозяйственных отраслях и обеспечения их жизнеспособности в условиях непрерывно развивающихся знаний.

Предмет исследования – методы, технология и инфраструктура построения и усовершенствования систем с базами знаний.

Методы исследования. Поставленные задачи решаются с использованием системного анализа, теории информации, теории графов и семантических сетей, теории множеств, технологии объектно-ориентированного программирования, облачных технологий, онтологического инжиниринга и других методов искусственного интеллекта (ИИ).

Научная новизна

В диссертационной работе получены следующие научные результаты: впервые разработаны *методы, модели и технология* обеспечения жизнеспособности *систем с базами знаний*, включающие:

1. новую иерархию постановок задач интеллектуальной деятельности в терминах единых математических абстракций для всех содержательных понятий, открывающую повторное использование готовых решений;
2. модель жизнеспособной системы для поддержки решения задач интеллектуальной деятельности в рамках разработанных постановок;
3. метод непрерывного развития баз знаний экспертами предметной области (без участия инженеров и программистов) на основе потока прецедентов с оцениванием корректности внесенных изменений;
4. новый метод комплексного оценивания корректности, наличия дефектов и несогласованностей в онтологических информационных компонентах на основе широкого спектра графовых моделей онтологий;
5. метод конструирования решателей задач интеллектуальной деятельности с повторным использованием онтолого-ориентированных программных единиц и операций на основе иерархии постановок, предлагающей готовые решения;
6. методология конструирования жизнеспособных систем для поддержки решения задач интеллектуальной деятельности (с учетом их места в иерархии постановок).

Теоретическая значимость работы заключается в развитии методов обеспечения важных свойств систем, основанных на знаниях – развиваемость баз знаний по мере развития ПрОбл и повторная используемость онтологических компонентов. Результаты диссертационного исследования использованы при выполнении государственных заданий (№ 0262-2014-00025 и 0262-2019-0004) по теме «Интеллектуальные системы обработки данных, знаний и принятия решений» и следующих проектов: РФФИ: № 19-29-01077, 18-07-01079, 17-07-00299, 16-07-00340, 15-07-03193, 14-07-00270, 13-07-00024, 12-07-00179; ДВО РАН: 12-И-П15-03, 12-III-A-01И-019; 12-II-УО-01И-001, 12-III-A-01И-016, проект 12-I-ОНИТ-04, 15-I-4-029.

Практическую значимость диссертационного исследования имеют следующие результаты:

1. готовая к применению технология конструирования СБЗ на основе онтологий, которая обеспечивает снижение трудозатрат на производство программных систем для интеллектуальной информационной поддержки при принятии решений, развиваемых по мере развития ПрОбл;
2. специализированная инфраструктура для выполнения всех этапов конструирования СБЗ отдельно взятой предметной области, включающая:
 - совокупность онтологий для формирования портала знаний и сервисов для отдельно взятой предметной области,

- информационные и программные повторно-используемые компоненты сервисов для поддержки решения задач,
- комплекс практически-полезных сервисов для этой ПрОбл,
- инструментальные подсистемы для развития баз знаний и других информационных компонентов в этой ПрОбл и оценивания их качества.

Область применения полученных результатов: разработка практически-полезных развиваемых интеллектуальных сервисов в предметных областях, где знания специалистов формализуемы.

Результаты исследований были апробированы при создании сервисов поддержки решений и комплексов сервисов для специалистов из медицины и вирусологии; проводится перманентное развитие комплекса практически-полезных сервисов этой ПрОбл. Полученные в работе результаты могут быть использованы при построении интегрированных информационных систем в других сферах деятельности.

Достоверность научных и практических результатов подтверждается созданием инструментальной среды развития - комплекса инструментальных средств для разработки и сопровождения всех компонентов программных систем с базами знаний. Построенные онтолого-ориентированные компоненты программной системы эволюционируют по мере изменения фактологического материала, расширения знаний и выявления на практике новых случаев, требующих корректировки знаний человеком-экспертом или автоматически.

Основные положения, выносимые на защиту:

1. Многоуровневая классификация интеллектуальных задач, отражающая важные свойства предметных областей.
2. Спецификация в единой формальной системе понятий классов задач, решаемых на основе формализованных знаний.
3. Алгоритмы решения практически полезных интеллектуальных задач: запроса дополнительной информации, диагностики развивающегося процесса, планирования воздействий на систему или объект и др.
4. Модель жизнеспособной системы для поддержки решения задач интеллектуальной деятельности в рамках их постановок.
5. Структурный подход к оцениванию информационных компонентов систем с базами знаний.
6. Методология обеспечения качества баз знаний интеллектуальных систем через монотонное повышение «оценки» их правильности.
7. Принципы архитектурного проектирования онтологического решателя (через композицию из повторно-используемых модулей, обрабатывающих конкретные типы отношений между понятиями).
8. Методология разработки жизнеспособных систем с базой знаний.

Апробация работы

Основные результаты диссертационной работы докладывались на следующих научных конференциях: конференция по искусственному интеллекту

(КИИ) (г. Москва, 2020, г. Ульяновск, 2019, г. Москва, 2018, г. Смоленск, 2016 и др.), "Знания - Онтологии – Теории» (ЗОНТ) (г. Новосибирск, 2019, г. Новосибирск, 2017 и др.), "Открытые семантические технологии проектирования интеллектуальных систем" (OSTIS) (г. Минск, 2019, г. Минск, 2015 и др.), «Russian-Pacific Conference on Computer Technology and Applications» RPC (г. Владивосток, 2010, г. Владивосток, 2017), Крымская конференция «СВЧ-техника и телекоммуникационные технологии» (КрыМиКо) (г. Севастополь, 2012 и др.), международная мультikonференция по инженерным, компьютерным и информационным наукам SIBIRCON (г. Новосибирск, 2019), «Системный анализ в медицине» (г. Благовещенск, 2018), на научных семинарах ИАПУ ДВО РАН и др. Кроме того, цикл работ «Облачные эволюционирующие инфраструктуры для диагностики процессов, управляемые онтологическими базами» (авторы Грибова В.В., Шалфеева Е.А.) получил I место в конкурсе ДВО РАН на премию имени академика А.А.Воронова (постановление № 13 Президиума ДВО РАН от 20 мая 2021 года).

Содержание диссертации соответствует специальности 2.3.5 «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей» (05.13.11). Задача 1 связана с методами и алгоритмами проектирования программных систем (п. 1 паспорта специальности); Задача 2 – с обеспечением качества, стандартизации и сопровождения программных систем (п. 10); Задача 3 – с управлением базами знаний (п. 4); Задача 4 – с методами верификации и тестирования, управлением базами знаний и оценкой качества и стандартизацией программных систем (п. 1, п. 4 и п. 10); Задача 5 – с моделями и алгоритмами проектирования программных систем и с сопровождением программных систем (п. 1 и п. 10). Задача 6 – с моделями, методами и программной инфраструктурой для организации территориально распределенной обработки данных и с сопровождением программных систем (п. 9 и п. 10).

Публикации

По теме диссертационной работы опубликованы 52 научные работы, из них: 15 статей в журналах из списка ВАК; 2 переводные версии и 14 публикаций, индексируемых в базах SCOPUS, WoS и MathSciNet; 21 статья в научно-технических журналах и сборниках; 2 свидетельства о регистрации программы.

Структура и объем работы. Диссертация состоит из введения, 6 глав, заключения, списка литературы. Общий объем диссертации составляет 217 страниц, в том числе список литературы из 170 наименований, 24 рисунка, 1 таблица. Диссертацию дополняет 1 приложение.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность диссертационной работы, сформулированы цель и задачи исследования, определены научная новизна и практическая значимость работы.

В первой главе 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. Проведен обзор методов и технологий такой автоматизации и методов обеспечения жизнеспособности для систем, основанной на знаниях. Обоснована цель диссертационной работы.

Проблематика систем, основанных на знаниях, стала актуальной в 1970-80-е годы. Множество публикаций (Уотермана, Лената, Хейеса-Рота, Кленси и др.) 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, и предполагалось, что полный обзор таких задач может позволить разработать методы их решения и перейти в этой области от искусства разработки систем, основанных на знаниях, называемых интеллектуальными системами (ИС), к технологии. В то же время представляемые классификации интеллектуальных задач не были основаны на единых принципах, каждый автор описывал некоторый набор задач в «своих» терминах, чаще всего в виде текста, без формальных постановок. Никто не определил формально в единой терминологии постановки всех известных экспертных задач.

Несмотря на отсутствие единой классификации задач интеллектуальной деятельности, в сфере искусственного интеллекта предложены универсальные 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» (Финн В.К. в 2004), а в гибридных интеллектуальных – с использованием нескольких интеллектуальных методов. Современные *системы с базами знаний* 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 методы и механизмы пополнения знаний и дообучения систем, известные из публикаций Финна В.К. в 2010 г; Братко И. в 2001; Воронцова К. в 2012; Fikes R. & McGuinness D. в 2000; Müller L. & Perry J. в 2019; Еремеева А.П. в 2008. Но получаемые новые версии баз знаний либо не удобны для оценивания экспертами, либо требуют перепрограммирования компонентов системы.

Инструменты и технологии обеспечения возможности усовершенствования ИС в процессе всего жизненного цикла (ЖЦ) согласно Boehm В., Dehaghani S.M.H. & Najrahimi N., Islam M. & Katiyar V., Musen M. должны обладать следующими основными свойствами:

- 1) доступность инструментов разработки широкому кругу потенциальных разработчиков,
- 2) поддержка коллективной разработки и сопровождения,
- 3) разделение компетенций разработчиков с предоставлением каждому типу разработчиков инструментов, ориентированных на их специализацию (БЗ должны разрабатывать эксперты ПрОбл, пользовательский интерфейс – дизайнеры интерфейса, решатели – программисты),
- 4) понятность представления знаний, особенно таких, как причинно-следственные, временные, пространственные связи, для экспертов ПрОбл,
- 5) возможность формирования и сопровождения знаний, не влияющих на работоспособность решателей,
- 6) реализация механизмов оценки баз знаний, сформированных вручную, их обучения и до-обучения,
- 7) интегрированность со сторонними ПС.

Реализация требований 1, 2 осуществляется через облачность или Интернет-доступность инструментов, реализация требования 2 – через типовую верхнеуровневую архитектуру системы с унифицированными внутренними интерфейсами, инструментальную поддержку компоновки подсистем для разных участников и хранилища повторно-используемых компонентов.

Требования 3, 4, 5 реализуются через представления знаний в привычных терминах и отделение декларативных знаний от процедурных, использование онтологий для создания БЗ и их сопровождения, семантическое представление знаний.

Требование 6 – через единую терминологическую основу для представления знаний и случаев (ситуаций из действительности), через различные механизмы обучения и пополнения знаний, в том числе индуктивные методы, методы и средства оценивания. Требование 7 - через разработку экспортруемых форматов обрабатываемой информации (и выбор распространенного языка кодирования программных компонентов).

Каждое из известных универсальных технологий и инструментов для создания ИС (их обзор дан в диссертации), (а также *Ontology-Driven Software*

development, модельно-управляемый подход, Collaborative Device Modeling Environment (DARPA)) отвечает нескольким из этих требований, но ни одно не поддерживает всех указанных требований одновременно. Для обеспечения возможности согласованному коллективу экспертов постоянно усовершенствовать БЗ этим технологиям не хватает:

- либо отделения знаний от данных и декларативных знаний от процедурных знаний;

- либо декларативного представления причинно-следственных,

- либо доступности инструментов коллективной разработки,

- либо интегрированности со средствами обучения.

Другие технологии, которые минимизируют эти недостатки, предлагают узко специализированные решения: проблемно-ориентированные средства автоматизации, описанные Грищенко М.А. и Дородных Н.О., Жилевым А.А. и др. или проблемно-предметно-ориентированные средства автоматизации, например, только диагностика и только в медицине (как Doknosis.org).

По аналогии с тем, как жизнеспособность программных систем обеспечивается (через адаптивность) к изменениям в среде и (через адаптируемость) в ответ на новые требования, жизнеспособность для системы, основанной на знаниях, проявляется, прежде всего, в условиях изменчивости знаний предметной области (реже – онтологии, среды функционирования и средств пользовательского интерфейса). Поэтому для большинства предметных областей, связанных с решением интеллектуальных задач, подразумевается эволюционируемость знаний. Более того, 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 программных систем поддержки специалистов, решающих важные для науки и практики интеллектуальные задачи, необходима современная методология обеспечения их жизнеспособности. Поэтому актуальной является разработка моделей, методов и технологии создания эволюционирующих интеллектуальных программных систем с базами знаний, реализующих требования к СБЗ, описанные выше.

Во второй главе предложен математический аппарат для обозначения содержательных понятий, необходимых для формулировки интеллектуальных задач. Средствами этого аппарата даны постановки задач из классификации интеллектуальных (экспертных) задач. В этих терминах введены и формально представлены различные общие и частные свойства ПрОбл и предложены постановки задач на более низких уровнях абстракции или новые характерные задачи, опирающиеся на эти свойства. Выделены вычислительные операции для реализации методов решения задач разных типов.

В зависимости от того, требуется ли сформировать БЗ или она задана, можно различать задачи индуктивного обобщения данных для построения БЗ и поиска гипотезы о решении или критики гипотезы (как вариантов дедукции), 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 анализа условий (ограничений) на решения. Эти категории «возглавляют» иерархию экспертных задач, специализируемых далее для охвата многообразия их формулировок в разных ПрОбл. 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 соответствиями и отношениями, зависящими от времени. Задачи анализа, критики гипотез, проектирования, критики проекта могут ставиться для динамических или статических систем; в задачах проектирования динамической системы присутствуют формулы, учитывающие временные аспекты развивающейся или функционирующей системы. Кроме уточненных задач, для динамических систем возникают и новые задачи (рис. 1).

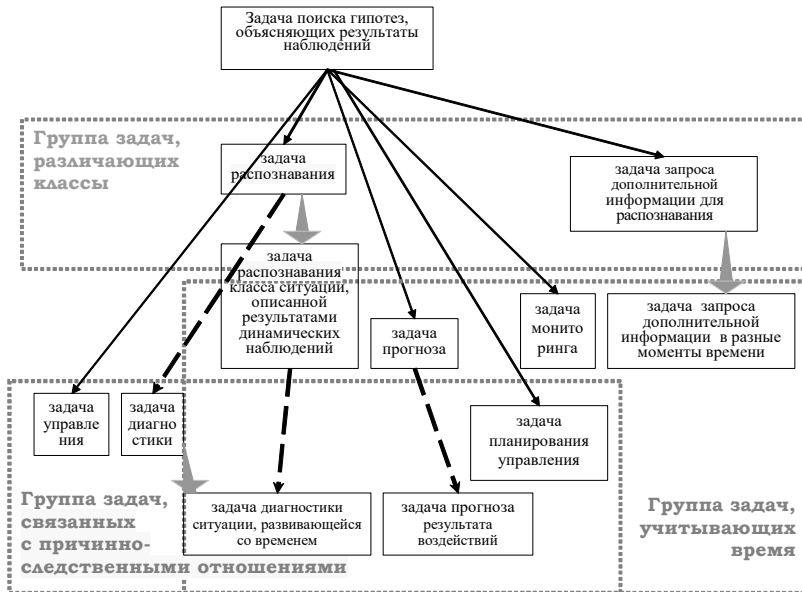


Рисунок 1 – Фрагмент многоуровневой классификации задач с учетом свойств предметных областей

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

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 раз-

личные моменты времени события, внешние по отношению к системе, но воздействующие на протекающие в ней. Нижеперечисленные примеры задач учитывают процессы, связанные между собой посредством причинно-следственных отношений.

В задаче *диагностики* требуется по результатам наблюдения признаков, характеристик системы и произошедших событий определить возможные причинно-следственные 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.

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), как у Братко И.; Джексона П.; Тельнова Ю.Ф.

В задаче *управления* (в описании Павлова С. Н. или Тельнова Ю.Ф.) 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), т.е. *управления* в смысле Clancey (1985), 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 задач, сформирована иерархия их постановок, выраженных в терминах единых математических абстракций для всех содержательных понятий.

Пример постановки задачи ниже дан для диагностики.

Дано: R (результаты наблюдения признаков, характеристик системы и произошедших событий); Kп (база знаний).

Найти: все возможные причинно-следственные модели (AS(R)) системы, согласованные с результатами наблюдений (R), относительно которых все предложения из Kп истинны. При этом диагноз (обозначаемый Δ) в каждой

такой модели – это внутренний процесс, не присущий системе (т.е. отклонение от нормы, часто – критическое).

Предложенная многоуровневая классификация задач позволяет в процессе системного анализа переходить к математическим постановкам различных задач, конкретизируя те понятия и их абстракции, которые содержатся в этой классификации, и увидеть задачи, более общие в своих постановках и более частные, характерные для ПрОбл с определёнными свойствами. Этой классификацией охвачен практически весь спектр задач, относящихся к области разработки систем, основанных на знаниях. Вместе с тем классификация является открытой, допускающей введение в неё абстракций новых понятий и новых задач. На практике в информационной поддержке нуждаются как задачи одного из классов, так и композиции нескольких таких задач.

В искусственном интеллекте поддержка решения таких задач связана с направлением разработки систем, основанных на знаниях, в предположении, что существует «правильная» база знаний $k^* \in K_n(X, Y)$ такая, 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 разрабатывать алгоритм решения задачи ($A: \langle X, K_n(X, Y) \rangle \rightarrow Y$) и реализовать его на алгоритмическом языке.

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* развивающийся во времени процесс 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.

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

Представлен структурированный анализ методов и алгоритмов решения задач распознавания, диагностики, запроса дополнительной информации; планирование воздействий и др. Алгоритмы программных компонентов (спе-

циализированных) решателей для таких задач предлагается создавать с использованием специфицированных операций.

В третьей главе рассмотрен подход к жизнеспособности и разработана модель жизнеспособной системы поддержки решения задач на основе баз знаний, установлены некоторые ключевые принципы проектирования архитектуры жизнеспособной СБЗ.

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

Ядро системы, решающей на основе баз знаний одну задачу, можно представить так.

$$\text{KBS-Arch} = \langle \cup^{i=1, N1} (\text{KB}_i), \text{Solv}_1, \cup^{m=0, M} (\text{U}_{i_m}), \cup^{p=0, N2} (\text{DB}_p) \rangle.$$

Здесь совокупность баз знаний обозначена как $\cup^{i=1, N1} (\text{KB}_i)$, программный решатель – как Solv_1 ; часто необходимы компоненты, обеспечивающие пользовательский интерфейс (U_{i_m}), и хранилища оперативной или справочной информации (DB_p).

Для интеграции компонентов явно специфицируются единые или специфичные правила связывания, т.е. внутренние интерфейсы ($\Delta\epsilon$): между KB_i и Solv_1 , между Solv_1 и DB_p), между Solv_1 и U_{i_m} :

$$\text{KBS-Arch} = \langle \cup^{i=1, N1} (\text{KB}_i), \text{Solv}_1, \cup^{m=0, M} (\text{U}_{i_m}), \cup^{p=0, N2} (\text{DB}_p) \rangle \cup \langle \cup^{p=1, N2} (\Delta\epsilon (\text{Solv}_1, \text{DB}_p)), \cup^{m=1, M} (\Delta\epsilon (\text{Solv}_1, \text{U}_{i_m})) \rangle.$$

Системный анализ ПрОбл, предшествующий началу автоматизации интеллектуальной деятельности, выявляет все задачи, нуждающиеся в информационной поддержке на основе знаний. *Системное проектирование* распределяет интеллектуальные задачи и их подзадачи и сопутствующие функциональные требования по компонентам (системы) 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.

Архитектурная модель *системы, основанной на знаниях*, интегрированной с другими программными компонентами (Soft_k), реализующими вычислительные и другие функции, можно представить так.

$$\text{KBS-Arch} = \langle \cup^{i=1, N1} (\text{KB}_i), \cup^{j=1, N2} (\text{Solv}_j), \cup^{m=1, N3} (\text{U}_{i_m}), \cup^{p=1, N4} (\text{DB}_p), \cup^{k=1, N5} (\text{Soft}_k) \rangle \cup \langle \cup^{1, N6} (\Delta\epsilon (\text{KB}_i, \text{Solv}_j)), \cup^{p=1, N2} (\Delta\epsilon (\text{Solv}_j, \text{DB}_p)), \cup^{m=1, M} (\Delta\epsilon (\text{Solv}_1, \text{U}_{i_m})), \cup^{1, N8} (\Delta\epsilon (\text{Solv}_j, \text{Solv}_r)), \cup^{1, N9} (\Delta\epsilon (\text{Solv}_j, \text{Soft}_k | \text{Soft}_k, \text{U}_{i_m} | \text{Soft}_k, \text{Ext}_{i_n} | \text{Soft}_k, \text{DB}_p | \text{Soft}_k, \text{Soft}_s)) \rangle.$$

Системный анализ и системное проектирование – ключевые этапы технологии создания жизнеспособных СБЗ, задающие их архитектуру как совокупность решателей подзадач.

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 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.

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). Выявление и формализация онтологии ПрОбл (понятие введено и уточнено Gruber T.R., Guarino N., Chandrasekaran V., Клещевым А.С. и Артемьевой И.Л., Боргестом Н.М.), охватывающей все виды связей и зависимостей между понятиями, обеспечивает понимаемость информации сообществом специалистов. Знания и другая информация должны формироваться в единых понятиях предметной области в соответствии со структурой, ограничениями и соглашениями, определяемыми онтологией ПрОбл.

Принципиально важно, что онтология ПрОбл создается как самостоятельный (отдельно хранимый и адресуемый) компонент. При промышленной разработке СБЗ ее окончательная версия должна существовать к началу работ по созданию компонентов системы (поскольку все компоненты, реализующие интеллектуальность в системе, разрабатываются на ее основе).

Онтология (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. $Onto = KnOnt \cup SitOnt \cup Agreem \cup Dict$.

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:

$$\langle \bigcup_{i=1, N1} (KB_i(KnOnt, Dict)), \bigcup_{j=1, N2} (Solv_j(KnOnt, SitOnt, Agree_m)), \bigcup_{m=1, N3} (U_{i_m}(SitOnt)), \bigcup_{p=1, N4} (DB_p(SitOnt, Dict)), \bigcup_{k=1, N5} (Soft_k) \rangle \cup \langle \bigcup_{i=1, N^*} (\Delta \in (KB_i - Solv_j \mid \mid Solv_j - DB_p \mid Solv_1 - U_{i_m} \mid Solv_j - Solv_r \mid Solv_j - Soft_k \mid Soft_k - U_{i_m} \mid Soft_k - ExtI_n \mid Soft_k - DB_p \mid Soft_k - Soft_s)) \rangle.$$

Это значительно повышает сопровождаемость интеллектуальных программных сервисов (в отличие от распространенных подходов к построению экспертных систем, когда решатель работает с представлением знаний, которое эксперт самостоятельно не сможет формализовать для сложной области (например, такой как медицина).

Конструирование СБЗ производится с использованием онтологического подхода, инструментарий для формирования всех информационных компонентов поддерживает соответствие онтологии предметной области. Каждый программный решатель создается по онтологии знаний, чтобы работать с любой БЗ, сделанной в терминах этой онтологии.

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

В четвертой главе представлены методы обеспечения качества формируемых декларативных информационных компонентов (ИнфКомп) для СБЗ: метод перманентного усовершенствования баз знаний, метод поиска дефектов и несогласованностей в разных ИнфКомп.

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, PrссВ)), 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 (соответствия современному представлению) 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 всерьез 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: $KB_{ver} = KB_{ver-1} (KnOnt) * (TrnSft (KnOnt, PrecB(SitOnt)) | (KBEd (KnOnt)) * AssSft (KnOnt, SamplB(SitOnt)))$. Примечание. 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.

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.

Помимо типичных методов программной инженерии, применяемых к программным решателям, другим программным единицам и пользовательскому интерфейсу, для информационных компонентов СБЗ применимы методы оценивания структурных свойств, полноты, влияния на процесс и результат решения задач, соответствия эталонам, и т.п. В зависимости от типа информационного компонента (KB, DB, KnOnt, SitOnt) оцениваются завершенность, морфологические характеристики, либо полнота и правильность).

Предложенный подход к мониторингу ИнфКомп (регулярной их проверке перед компоновкой версии СБЗ) позволяет получать объективные значения структурных свойств, выявлять дефекты и другие недостатки, избыточность (использованность элементов онтологий в БД и БЗ) и задавать субъективные показатели качества информации.

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. По отношению ко всем хранимым ИнфКомп применяется ранний контроль свойств их качества. Средства, реализующие методы мониторинга и контроля их качества – элементы модели конфигурации жизнеспособной СБЗ.

Таким образом, определен метод перманентного монотонного усовершенствования баз знаний, а также метод оценивания корректности, наличия дефектов и несогласованностей в БЗ и других ИнфКомп. Проектирование архитектуры СБЗ, интегрированной с подсистемами для управления качеством БЗ – ключевой принцип жизнеспособности.

В пятой главе рассмотрен подход к разработке решателей СБЗ и разработан метод декларирования и конструирования таких решателей с повторно используемыми вычислительными операциями.

Определены требования к решателю в СБЗ. Онтолого-ориентированный (или онтологический) решатель $Solv$ ($KnOnt \cup SitOnt \cup Agreeem$) в составе СБЗ должен быть способен выдвигать гипотезы по результату сопоставления входной информации (об объекте) – предложениям из БЗ КВ ($KnOnt, Dict$) на основе онтологических соглашений $Agreeem$ о связи данных и знаний и с учетом знания структуры предложений (связей терминов) $KnOnt$ и ограничений на интерпретацию смысла терминов. Задача решателя $Solv_j$ ($KnOnt_{j1}, \dots, KnOnt_{jw}, SitOnt_{jy}$) – предложить путем «рассуждения» одно или несколько обоснованных решений-гипотез, использующих входную информацию и согласованных с Базой знаний KB_{iu} ($KnOnt_{iu}$)

Онтологический решатель $Solv_j$ - программно реализованное «рассуждение», алгоритм, проводящий обход каждой Базы знаний KB_{iu} (на основе знания ее структуры и онтологии в целом - $KnOnt_{iu}$) для сопоставления входным данным (фактам или условиям). Решатель для выдвижения и объяснения гипотез обрабатывает входную информацию и строит выходную, фиксируя в ней шаги принятия или отклонения возможных логических заключений.

Решатель, способный выдвигать гипотезы на основе онтологических соглашений, структуры предложений и ограничений на интерпретацию, сохраняет аргументы в пользу гипотез о решении (объясняет их). 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, прогноза, распознавания класса). Это делается через: сопоставление фактов элементам знаний на основе онтологических соглашений и перехода к их анализу. Логический вывод решателя для задач планирования, проектирования, управления – последовательная проверка подцелей, соответствующих элементам знаний, и поиск соответствующих им элементов или действий, полная совокупность которых, согласованная с онтологией, становится гипотезой.

Решатель (умеющий выдвигать гипотезы и объяснять их) проектируется из программных единиц (ПрЕд, Unit_m) разных типов, с доступом и без доступа к ИнфКомп, для вычислений или связи с внешним окружением.

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 (Unit_g), 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 Agreeem.

В совокупности такие ПрЕд, как правило, «покрывают» все утверждения из $\text{KnOnt} \cup \text{Agreeem}$. Решатель, создаваемый из ПрЕд, программируемых по такому принципу, – онтологический.

В алгоритме решателя происходит поочередный вызов ПрЕд – процедур вывода следствий из обработанных посылок, которые записывают результат проверки посылок в объяснение. Онтологическими являются *ПрЕд* двух типов: *операции* SemOp_m, реализующие обработку одного онтологического ИнфКомп в соответствии с указанными в его онтологии структурными связями, и *ПрЕд* SemUnit_m – обработку множества онтологических ИнфКомп в соответствии с конкретными типами причинно-следственных и других связей в онтологии решаемой задачи. Для проведения тестирования ПрЕд решателя разработан инструмент мониторинга состояния программных агентов.

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). Т.е. степень повторного использования кода будет разной.

Операции доступа к содержимому онтологической информации (такие как: *прочитать значения наблюдений, найти метрики объекта, найти временной ряд указанного наблюдения*) наиболее связаны со структурными и причинно-следственными связями используемых понятий (определяемыми KnOnt и SitOnt). И с конкретной онтологией (KnOnt_m, или SitOnt_n) связывается множество таких Op_{ms}, Op_{nz}, чтобы применяться затем к любой информации (знаниям, данным), управляемой соответствующей онтологией.

Так же, как в программной инженерии, проектировщик-архитектор решателя, создаваемого по онтологии, планирует на роль онтологических операций прежде всего те, которые повторяются при реализации полного рассуждения.

Формирование и использование операций доступа Op_{ij} для управляемого программного доступа к целевым базам знаний и др. хранимой информации – один из принципов реализации решателя.

В общем случае решатель задач создается из (процедурных) ПрЕд (модулей) разных типов:

$$\text{Solv}_j(\text{KnOnt}_{jy_1}, \dots, \text{KnOnt}_{jy_w}, \text{SitOnt}_{yx}) = \{ \cup^{m=1, M} (\text{SemUnit}_m(\text{KnOnt}_{m_1}, \dots, \text{SitOnt}_{m_q})) \cup \{ \cup^{m=1, M} (\text{SemOp}_m(\text{KnOnt}_{m_1}), \dots, \dots, \text{SemOp}_k(\text{SitOnt}_{m_q})) \} \cup \{ \cup (\text{Prс}_x; \text{Prd}_y, \text{HrdOp}_u, \text{UiOp}_v) \}.$$

Предложенная архитектурная классификация соответствует потребностям создания сложных эволюционирующих систем, основанных на знаниях.

В результате применения этого подхода онтологический решатель вместе с пользовательским интерфейсом (ПИФ) многократно используется для множества систем (сервисов). Он не зависит от наполнения баз знаний (использует ее как параметр), поэтому их полнота и качество могут улучшаться бесконечно, интегрируясь с решателем в новые и разные версии полезных сервисов. Тем самым обеспечивается **возможность усовершенствования сервисов без участия программистов**, т.е. в условиях уже существующих программных компонентов.

Чтобы решатель Solv_j мог быть использован как часть множества разных сервисов, должен быть специфицирующий Solv_j документ. В нем важно точное указание всех $\text{KnOnt}_{jy_1}, \dots, \text{KnOnt}_{jy_w}, \text{SitOnt}_{jy_1}, \dots, \text{SitOnt}_{jy_x}$ и ExplOnt . Специфицирование онтологии обрабатываемых Решателем знаний и данных является основой интегрируемости решателя задач с другими компонентами сервисов. Этот документ может быть декларативным (интерпретируемым человеком и машиной). Кроме спецификации форматов знаний и данных, могут быть определены все Unit_m и даже порядок их выполнения и связей друг с другом. Такая декларация Решателя может использоваться для а) автоматического контроля сборки, динамической (run time) загрузки частей по мере потребности в них, б) ручной сборки СБЗ из Решателя и всех обрабатываемых ИнфКомп и сборки Решателя из ПрЕд (специфицирующие документы SemUnit и SemOp явно указывают онтологию обрабатываемой информации). Проектирование архитектуры решателей задач из декларируемых программных компонентов разных типов – шаг в направлении создания жизнеспособных систем (и еще один ключевой принцип проектирования).

Декларирование решателей задач, проектирование архитектуры решателей задач в соответствии с типом решаемой задачи и набором рассматриваемых структурных и причинно-следственных связей между сущностями, сборка решателя из заменяемых программных компонентов, – обеспечивают про-

зрачность его архитектуры и сопровождаемость. Это особенно важно в условиях коллективной разработки СБЗ и их развития.

Следовательно, важна инструментальная поддержка декларирования решателя (произвольной задачи) и декларирования его компонентов. Кроме средств (инструментальных сервисов) декларирования нужны средства создания заготовок исходных кодов по декларациям, средства кодирования новых ПрЕд, средства каталогизации ПрЕд как потенциально *повторно используемых компонентов* и средства интеграции готовых компонентов и новых ПрЕд в новые решатели или их новые версии.

Комплекс таких инструментальных сервисов может быть создан на основе онтологического подхода. Тогда технология разработки рассматривается как ПрОбл, а онтология процесса разработки станет основой онтологических *программных* средств разработки (инструментальных сервисов). В частности, инструментальные сервисы декларирования решателя и ПрЕд – это Редакторы (соответствующей информации), генерируемые по определенной части *онтологии технологии разработки СБЗ*, описывающей содержание (и формат) декларации.

На основе этого онтологического подхода в зависимости от потребностей разработчиков можно по-разному организовать «среду разработки и развития» для СБЗ.

Минимальная среда разработки и развития СБЗ представляет собой:

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

Типичная среда разработки и развития дополняет *минимальную* множеством подготовленных и проверенных ИнфКомп (БЗ, архивов и баз данных – DB, KB_{iw}); библиотеками протестированных ПрЕд (KnOnt_{m1}, SemOp_m, Prc_x, Prd_y, HrdOp_u, UiOp_v), обрабатывающих ИнфКомп; множеством готовых решателей задач Solv_j, 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.

Инструментарием разработчиков программных частей обеспечиваются свойства СБЗ, относящиеся к жизнеспособности:

- допустимость усовершенствования метода принятия решения;
- допустимость изменения или добавления функций (например, формирования дополнительных результатов);
- адаптивность ПИФ пользователя в связи с изменением входных данных и формируемых результатов функций.

Для предложенного (в гл. 4) управления качеством баз знаний на основе прецедентов разработан *метод онтологической реализации*. Он состоит в

использовании онтологии входных и выходных данных соответствующей задачи при построении компонентов проверки баз знаний (с отчетом о несоответствиях и их объяснении) и поиска обучающих прецедентов.

Программное средство проверки правильности БЗ работает с ExpOnt, в котором собрано и объяснено несоответствие утверждений БЗ описанию ситуации в эталонных примерах.

Специализированные инструменты обеспечения качества используют общее подмножество онтологических ПрЕд, что и решатель (т.к. ориентированы на те же виды связей), реализуя принцип повторного использования.

Особенности, принципиально отличающие онтологический решатель от других 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 (в отличие от сервисов, реализуемых через механизм, например, нейронных сетей).

Построение СБЗ с помощью инструментальных онтологических средств 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).

Совокупные ключевые принципы методологии создания жизнеспособных систем – проектирование декларативных БЗ по онтологии знаний, проектирование архитектуры решателей задач из онтологических ПрЕд в соответствии с онтологией ПрОбл, создание подсистемы управления качеством всех БЗ с проектированием ее компонентов.

В шестой главе представлена концепция облачной технологии разработки СБЗ, методика онтолого-ориентированного представления и обработки информации для поддержки решения экспертных задач, требования к комплексу инструментальных средств разработки и сопровождения всех компонентов конфигурации жизнеспособной СБЗ, метод реализации программного комплекса инструментальных средств, описана инструментальная среда развития.

Предложенная технология разработки современных СБЗ нацелена на многообразное повторное использования готовых решений. 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 путь построения (за счет использования готовых решений) и коллективно-

параллельную работу разных участников, снижает технические риски («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.

Онтология (знаний и данных) связана с решателем, который может быть непосредственно использован как подсистема создаваемой СБЗ. Если онтология уточнена, то решатель потребует адаптации, а база знаний создается экспертами с нуля. Если онтология не меняется и если она применялась для создания БЗ в той же ПрОбл, что и рассматриваемая, есть шанс повторно использовать готовые версии БЗ (пример – фармакологический справочник). Возможна ситуация, когда библиотека не содержит готовых решений для выбранной задачи в ПрОбл. 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.

Разработана технология обеспечения качества БЗ, реализующая 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 «отвечает» облачная среда IACPaaS (<http://iacpaas.dvo.ru>). 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 IACPaaS (ИнфРес). Инструментарий позволяет управлять размещением готовых ПрЕд в папках хранилища для обеспечения их поиска и по-

вторного использования в процессе создания программных решателей или их новых версий.

Последнее распределение функциональности при планировании архитектуры поддерживается многоуровневой концепцией хранения всех видов компонентов (ресурсов) и конструирования:

- уровень метаязыка;
- уровень формализованных онтологий (или метаинформации IACPaaS) и API (application programming interface) доступа к элементам ИнфРес (двух-уровневых иерархических семантических сетевых моделей);
- уровень онтологических баз знаний, данных (ИнфРес, построенных по метаинформации) и классов онтологических операций (реализованных через API) над любыми ИнфРес, созданными по этой онтологии;
- уровень ПрЕд (агентов), работающих с информацией через операции;
- уровень решателей задач, построенных через агентов, ПрЕд обработки информации,
- уровень сервисов, конструируемых из баз знаний, данных, решателей.

Эта стратегия накопления и хранения позволяет формировать необходимую и достаточную *среду разработки и развития СБЗ* – минимальную (как на рис. 2 с апробированной онтологией ПрОбл, базой терминов, сгенерированными IACPaaS редакторами ИнфРес, штатными редакторами деклараций ПрЕд, Java IDE), *типичную* или *полную*.

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

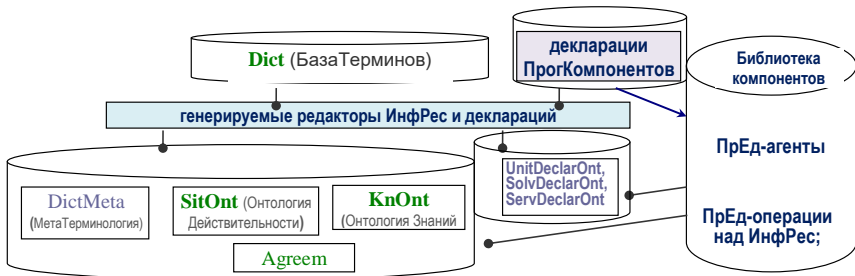


Рисунок 2.– Модель специализированной технологической среды

Жизнеспособность СБЗ обеспечивается свойствами специализированной среды ее развития:

- поддержка обновления знаний;

- поддержка изменения компоновки онтолого-ориентированного Решателя (в связи с заменой компонента, реализующего другую стратегию принятия решения или метода получения результата или в связи с добавлением компонента, реализующего дополнительную функцию (формирования дополнительного результата));

- поддержка усовершенствования интерфейса пользователя в связи с изменением функций;

- поддержка изменения в онтологии;

- поддержка усовершенствования интерфейса эксперта (и пользователя) в связи с обновлением онтологии;

- поддержка кодирования новых версий программных единиц онтолого-ориентированного решателя или поддержка изменения кода ПрЕд такого Решателя (т.е. адаптации обхода декларативной Базы знаний в связи с обновлением онтологии знаний).

Технология разработки и развития СБЗ реализуется с помощью инструментального комплекса - облачной инструментальной среды развития на IACPaaS.

Представлено практическое использование инструментального комплекса для коллективной разработки и обеспечения жизнеспособности СБЗ медицинского облачного Портала знаний. Дана сравнительная оценка эффективности внедрения таких облачных СБЗ в отрасль, сотрудники которой занимаются интеллектуальной деятельностью с повышенной ответственностью за правильность решений.

Процесс формирования портала и разработки управляемых интеллектуальных сервисов в предметной области «медицина» включал разработку: словаря терминов (около 15000) для представления объектов исследования и рассуждений об их состоянии; онтологии истории болезни и медицинской карты (около 200 понятий – узлов семантической сети) как онтологии входных данных и результатов; адаптацию онтологии диагностики к онтологии медицинской диагностики (добавлено около 30 понятий); онтологии фармакологического справочника (около 50 понятий), онтологии схем лечения заболеваний (около 80 понятий); форматов сложно структурированных отчетов о проведенном анализе для объяснения гипотез (для решаемых задач), получаемых на основе знаний.

На примере поддержки практической интеллектуальной деятельности в медицине проведен сравнительный анализ трудозатрат и эффекта от автоматизации медицины как отраслевой деятельности в двух случаях: облачной реализации СБЗ с разделением компетенций и традиционной разработки СБЗ (для каждого заказчика-учреждения).

Рассмотрим N условных медучреждений, работающих по P медпрофилям. Пусть $N = 10$ учреждений, $P = 5$ профилей (и K команд разработчиков,

вовлеченных в автоматизацию, предположительно работающих по разным технологиям).

На рис.3 график сверху – оценка затрат при традиционной разработке, снизу – при автоматизации отраслевой деятельности через «облачные» системы с декларативными БЗ и универсальными решателями.

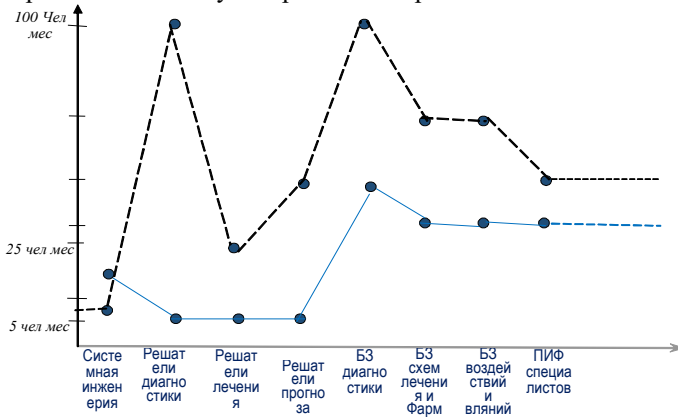


Рисунок 3 – Графическое представление сравнения оценочных трудозатрат для разных подходов к автоматизации интеллектуальной медицинской деятельности

Графики на рис.3 получены путем измерения трудозатрат при выполнении работ по построению компонентов интеллектуальных систем для нескольких медицинских коллективов и сообществ. По горизонтали представлены некоторые принципиальные этапы работ, по вертикали – оценки на их выполнение в человеко-месяцах, например:

один решатель диагностики потребовал около 10 чел/мес vs затрат на множество таких решателей (для $K \cdot P = 10/5 \cdot 5 = 10$) = $10 \cdot 10 = 100$ чел/мес.

Аналогично:

1 Решатель лечения = 5 чел/мес vs M решателей лечения (для $K \cdot P = 10/5 \cdot 5/2 = 5$) $5 \cdot 5 = 25$ чел/мес.

1 Решатель прогноза = 10 чел/мес vs M решателей ($K \cdot P = 10/5 \cdot 5/2 = 5$) $5 \cdot 10 = 50$ чел/мес.

База знаний диагностики по 1 профилю = 10 чел-мес, P баз = 50 чел/мес, но множество дублирующих друг друга ($N \cdot P$) баз требует $10/5 \cdot 5 = 100$ чел/мес.

На диаграмме ниже (Рис.4) штриховкой показано оценочное число специалистов (по вертикали), участвующих в развитии (поддержании актуальности и усовершенствовании) СБЗ при традиционном подходе, нижний уровень показывает ситуацию при облачной реализации СБЗ с разделением компетен-

ций. По горизонтали представлены некоторые принципиальные этапы и результаты работ, выполняемые разными категориями участников.

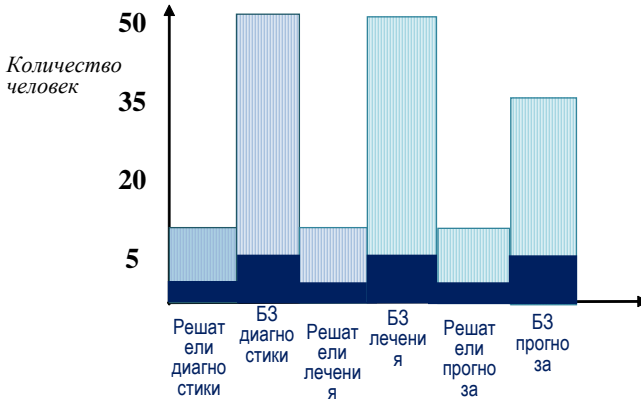


Рисунок 4 – Графическое представление сравнения усилий для разных подходов к автоматизации деятельности

Усовершенствование баз знаний диагностики = P чел = 5 чел

$N * P$ специалистов = $10 * 5 = 50$ чел

Усовершенствование баз знаний лечения = 5 чел-мес

$N * P$ специалистов = $5 * 10 = 50$ чел

Усовершенствование Решателя диагностики = 2 чел

$K * P = 10/5 * 5 = 10$ чел.

Более подробные расчеты типичных трудозатрат (на системную инженерию, программное обеспечение, базы знаний, включая управление ими, вспомогательные и организационные процессы), показан в виде таблицы.

Таблица 1. Сравнительная оценка усилий

| Виды работ | традиционная разработка для N учреждений (работающих по P профилям) систем с БЗ | автоматизация отраслевой деятельности через «облачные» системы с декларативными БЗ |
|--------------------------------|---|--|
| <i>на software:</i> | | |
| Создание Решателей диагностики | M штук решателей, $1 < M < N$ или даже $1 < M < N * P$. Чаще M ближе к N или $N * P$. В идеале $M = K$ (числу | 1 |

| | | |
|--|--|---|
| | компаний-разработчиков ПС и МИС), но скорее $M = K * P$ (если решатель не универсален). Поскольку не стандартизована терминология специалистов, мало шансов на $M = K$. | |
| Создание Решателей планирования лечения | M штук, $1 < M < K * P$ (можно ожидать стандартизованности терминологии специалистов разных учреждений в области лечения). | 1 |
| Создание Редактора знаний диагностики | M штук, $P < M < N * P$. Возможно, 1, если универсальный, тогда необходимы $N * P$ инженеров по знаниям,. | 1 (или 0, если генерируемый по онтологии) |
| <i>на базы знаний:</i> | | |
| Создание Базы знаний диагностики | L штук баз, $1 < L < N * P$. $L = S$ (числу компетентных экспертов знаний). | P |
| Создание Базы знаний планирования лечения | P (баз схем лечения от МинЗдрава) +1 (ФармСправ) | P (баз схем лечения от МинЗдрава) +1(ФармСправ) |
| Усовершенствование БЗ диагностики | $N * P$ специалистов | P специалистов |
| Усовершенствование баз знаний планирования лечения | $N * P$ специалистов – каждый вносит изменение в свою БЗ или проверяет автоматически сгенерированные БЗ | P специалистов |
| Усовершенствование Решателя диагностики | $M = K * P$ | 1 |

Трудозатраты на системную инженерию (обычно выполняемую организацией-разработчиком) складываются из:

- проведения системного анализа профессиональной деятельности и формирования онтологии предметной области;

- концептуального проектирования системы автоматизации, документирования требований на разработку системы;
- интеграции всех подсистем и БЗ в единую систему;
- процессов комплексирования, верификации, технического обслуживания и других, традиционно относящихся к системной инженерии.

Трудозатраты на программное обеспечение состоят в разработке, а затем сопровождении следующих видов программных средств: решатель, редактор знаний, подсистема документирования, подсистема формирования обучающей выборки, подсистема модификации БЗ, подсистема оценивания БЗ.

Трудозатраты на формирование базы знаний складываются из затрат:

- на разработку первого варианта каждой базы знаний (с помощью редакторов знаний и подсистем оценивания);
- на сопровождение и управление базами знаний (с помощью редакторов знаний, АРМов, подсистемы формирования вариантов модификации БЗ и подсистемы оценивания БЗ).

Автоматизация одного учреждения состоит в разработке СБЗ для всех классов решаемых там задач интеллектуальной деятельности (например, в больнице, как основном звене медицинской отрасли) и обеспечивает эффект в получении преимуществ в повседневной деятельности и в совершенствовании знаний только специалистами одного учреждения (т.е. в зависимости от сложности задач, которые этим специалистам приходится решать).

Трудозатраты (для m профилей деятельности):

- на базы знаний и системы управления ими – их $n * m$ штук,
- на интеграцию n штук решателей с подсистемой документооборота и подсистемами управления качеством $n * m$ баз знаний.

«Облачная» реализация системы означает, что для всей отрасли предлагается использование общей 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, специалисты из основных звеньев отрасли используют «облачные» СБЗ, а коллектив экспертов каждого профиля (в идеале - специально выделенные высококвалифицированные специалисты) управляет качеством баз знаний посредством «облачно» доступных инструментов. Эффект от этой парадигмы состоит в получении всех преимуществ в повседневной деятельности и в достижении наивысшего качества БЗ, т.к. оно определяется полным спектром решаемых по всей отрасли задач – от типичных до самых сложных.

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

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 мукополисахаридам сформировали знания, апробировали их на реальных примерах пациентов из разных стран и приступили к уточнению знаний. Специалисты по неврологии сформировали терминологическую базу симптомов для оцифровки имеющихся архивов и формирования знаний о диагностике, лечении и прогнозе выздоровления. Создана терминологическая база симптомов кардиологических патологий. Формируется база знаний о влиянии патологических состояний и прочих факторов на возникновение и развитие кардиологических заболеваний и их осложнений. Специалисты формируют клиническую картину заболеваний с десятками симптомов, как правило, динамических (рис. 5).

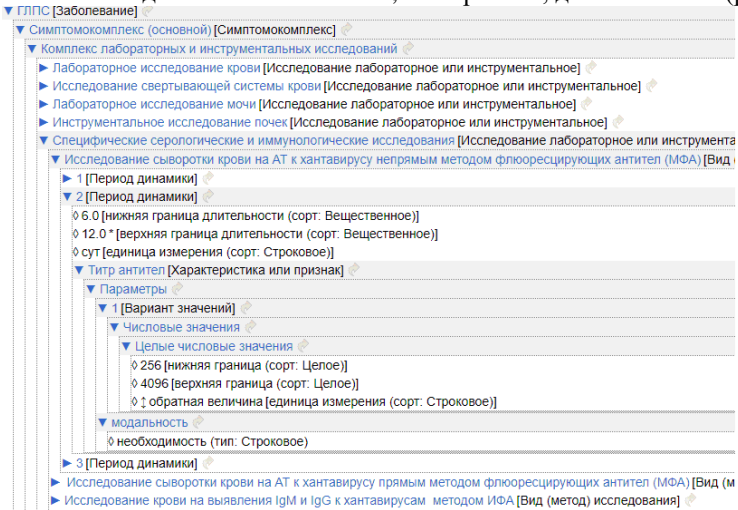


Рисунок 5 – Фрагмент формализации знаний инфекционных заболеваний

В заключении приведены основные результаты, полученные в диссертационной работе.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Новая многоуровневая классификация задач интеллектуальной деятельности, основанная на принципе усложнения свойств ПрОбл и позволяющая определить место решаемой задачи ПрОбл среди множества известных задач и готовых решений.

2. Формальные постановки задач интеллектуальной деятельности, обеспечивающие основу для поиска готовых решателей и конструирования повторно используемых компонентов систем.

3. Алгоритмы решения практически важных задач интеллектуальной деятельности (запроса дополнительной информации, диагностики развивающегося процесса, выработки вариантов решений по воздействию на систему или объект и др.) и метод архитектурного проектирования онтологических решателей задач из повторно-используемых программных единиц.

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

5. Метод непрерывного улучшения баз знаний и структурный подход к оцениванию всех информационных компонентов для их проверки на ранних этапах разработки интеллектуальных программных систем.

6. Технология формирования 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.

Перспективами дальнейшей разработки темы являются создание методов интеграции компонентов СБЗ с подсистемами, реализующими иные методы искусственного интеллекта, и подготовка инфраструктуры для декларативного формирования таких гибридных систем.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в журналах, рекомендованных ВАК России:

1. Шалфеева, Е.А. Преобразование моделей анализа требований в проектные представления интеллектуального решателя // Информатика и системы управления. 2013. №4(38). С. 100-110.
2. Шалфеева, Е.А. Мониторинг информационных ресурсов жизнеспособной интеллектуальной программной системы // Программная инженерия, 2012. №1, С. 10-15.
3. Шалфеева, Е.А. Управляемая графовыми грамматиками разработка оболочек интеллектуальных сервисов на облачной платформе IASPAAS / Грибова В.В., Клещев А.С., Москаленко Ф.М., Тимченко В.А., Федорищев Л.А., Шалфеева Е.А. // Программная инженерия. 2017. Т. 8. № 10. С. 435-447.
4. Шалфеева, Е.А. Базовая технология разработки интеллектуальных сервисов на облачной платформе IASPaas. Часть 1. Разработка базы знаний и решателя задач / В.В. Грибова, А.С. Клещев, Д.А. Крылов, Ф.М. Москаленко, В.А. Тимченко, Е.А. Шалфеева // Программная инженерия. 2015. № 12. С. 3-11.
5. Шалфеева, Е.А. Базовая технология разработки интеллектуальных сервисов на облачной платформе IASPaas. Часть 2. Разработка агентов и шаблонов сообщений / Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А. // Программная инженерия. 2016. №1. С. 14-20.
6. Шалфеева, Е.А. Базовая технология разработки интеллектуальных сервисов на облачной платформе IASPaas. Часть 3. Разработка интерфейса и пример создания прикладных сервисов / В.В. Грибова, А.С. Клещев, Ф.М.

Москаленко, В.А. Тимченко, Е.А. Шалфеева, Л.А. Федорищев // Программная инженерия. 2016. Т. 7. № 3. С. 99-107.

7. Шалфеева, Е.А. Облачная платформа IASaaS для разработки оболочек интеллектуальных сервисов: состояние и перспективы развития / Грибова В.В., Клещев А.С., Москаленко Ф.М., Тимченко В.А., Федорищев Л.А., Шалфеева Е.А. // Программные продукты и системы. 2018. Т. 31. № 3. С.521-526.

8. Шалфеева, Е.А. Особенности автоматизации интеллектуальной деятельности / А.С. Клещев, М.Ю. Черняховская, Е.А. Шалфеева // Научно-техническая информация. Сер.2, 2015, № 1, с. 10–20.

9. Шалфеева, Е.А. Управление интеллектуальными системами / Грибова В.В., Клещев А.С., Шалфеева Е. А. // Известия РАН. Теория и системы управления. 2010. № 6. С. 122-137. (Переводная версия: Gribova V.V., Kleshchev A.S., Shalfeeva E.A. Control of intelligent systems // Journal of Computer and Systems Sciences International, 2010. № 49(6), p. 952-966.)

10. Шалфеева, Е.А. Метод оценивания структуры онтологий для реализации редакторов знаний и данных // Вестник компьютерных и информационных технологий, Москва, "Издательство Машиностроение", 2010, № 2 (68). С. 23-32.

11. Шалфеева, Е.А. Медицинские ресурсы IASaaS для дифференциальной диагностики заболеваний желчного пузыря / Петряева М.В., Лифшиц А.Я., Шалфеева Е.А. // Информатика и системы управления. 2018. № 3(57). С. 81-92.

12. Шалфеева, Е.А. Онтология медицинской диагностики для интеллектуальных систем поддержки принятия решений / Грибова В.В., Петряева М.В., Окунь Д.Б., Шалфеева Е.А. // Онтология проектирования. 2018. Т. 8. - №1(27). С. 58-73.

13. Шалфеева, Е.А. Метод решения задачи запроса дополнительной информации / Грибова В.В., Клещев А.С., Шалфеева Е.А. // Онтология проектирования. 2017. Т. 7. N3(25). С. 310-322.

14. Шалфеева, Е.А. Определение структурных свойств онтологий / Клещев А.С., Шалфеева Е.А. // Известия РАН. Теория и системы управления, 2008. №2. С. 69–78. (Переводная версия: Kleshchev A.S. Shalfeeva E.A. Defining structural properties of ontologies // Journal of Computer and Systems Sciences International, 2008. № 47(2), p. 226-234.

15. Шалфеева, Е.А. Измерение структурных свойств для оценивания, сравнения и выбора онтологий // Информатика и системы управления, 2008. - №4 (18), С. 106-115.

Публикации, индексируемые международными базами

16. Shalfeeva, E. A Cloud Computing Platform for Lifecycle Support of Intelligent Multi-agent Internet-services / Gribova V, Kleshchev A, Krylov D, Moskalenko P, Timchenko V., Shalfeeva, E. // International Conference on Power Electronics and Energy Engineering (PEEE 2015) Hong Kong, People's Republic of China, 2015. P.: 231-235.

17. Shalfeeva, E.A. Monitoring of conceptual informational resources for intelligent software systems // RPC 2010 - 1st Russia and Pacific Conference on Computer Technology and Applications, p. 140-144.
18. Shalfeyeva, Ye.A. Features of the system analysis at automation of intellectual professional activity / Kleshchev A.S., Shalfeyeva Ye.A. // 22nd International Crimean conference microwave and telecommunication technology (CRIMICO 2012), Conference proceedings. 2012. P. 419-420.
19. Shalfeeva, E. Ontology of anomalous processes diagnosis / Gribova, V., Shalfeeva, E. // International Journal of Intelligent Systems. 2021. Volume 36, Issue 1. p. 291-312.
20. Шалфеева, Е.А. Постановки практически полезных задач интеллектуальной деятельности / А.С. Клещев, Е.А. Шалфеева // Дальневост. матем. журн., 2016. № 16(1). С. 44–61.
21. Shalfeeva, E. Ontology for differential diagnosis of acute and chronic diseases / Gribova, V., Okun, D., Petryaeva, M., Shalfeeva, E., Tarasov, A. // Communications in Computer and Information Science, 2018. V. 934, p. 152-163.
22. Shalfeeva, E. Features of the automation of intellectual activities /A.S. Kleshchev, M.Yu. Chernyakhovskaya, E.A. Shalfeeva // Automatic Documentation and Mathematical Linguistics, 2015, V. 49, Issue 1, p. 10-20.
23. Shalfeeva, E. Knowledge bases' control for intelligent professional activity automatization / Kleshev, A., Shalfeeva, E. // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015. V. 9227, p. 519-525.
24. Shalfeeva, E. Technology for the development of intelligent service shells based on extended generative graph grammars / Gribova, V., Kleshev, A., Moskalenko, Ph., Timchenko, V., Fedorischev, L., Shalfeeva, E., Zamburg, E. // Proceedings of the 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC). 2018. 8482129.
25. Shalfeeva, E. The IACPaaS cloud platform: Features and perspectives / Gribova V., Kleshev, A., Moskalenko P., Timchenko V., Fedorischev L., Shalfeeva E. // Second Russia and Pacific Conference on Computer Technology and Applications (RPC). IEEE, p. 80-84 (2017).
26. Shalfeeva, E. A generalised structural model of structured programs for software metrics definition / Cogan B., Shalfeeva E.A. // Software Quality J., 2002. № 10. P. 147–165.
27. Shalfeeva, E. The Model of Knowledge Base Quality Control with Assessment / Gribova V.V., Shalfeeva, E.A. // International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia, 2019, IEEE Xplore. P. 0872-0877.
28. Shalfeeva, E. Cloud service for the differential clinical diagnostics of acute respiratory viral diseases (including those associated with highly contagious coronaviruses) with an application of methods of artificial intelligence / Gribova V.V., Okun D.B., Shcheglov B.O., Shchelkanov M.Yu., Shalfeeva E.A. // Yakut Medical Journal, 2020. N 2. P. 44–47.

29. Shalfeeva, E. Formalization of medical records using an ontology: patient complaints / Klyshinsky E., Gribova V.V., Shakhgeldyan C.I., Shalfeeva E.A., Okun D.B., Geltser B.I., Gorbach T.A., Karpik O.D. // Communications in Computer and Information Science. 2020. Vol. 1086. P. 143-153.

Публикации в других изданиях

30. Шалфеева, Е.А. Облачный сервис поддержки принятия диагностических решений в гастроэнтерологии / Грибова В.В., Петряева М.В., Шалфеева Е.А. // Врач и информационные технологии, Москва, ООО Издательский дом «Менеджер здравоохранения», 2019. № 3. С. 65–71.

31. Shalfeeva, E. Ontological Approach to Viable Decision Support Services Development / Gribova V., Shalfeeva E. // Advances in social science, education and humanities research, 2020. Том 483. p. 274-277.

32. Шалфеева, Е.А. Онтология задач интеллектуальной деятельности / А. С. Клещев, Е. А. Шалфеева // Онтология проектирования. Самара: “Новая техника”, 2015. № 2 (16), с. 179-205.

33. Шалфеева, Е.А. Принципы организации каталога свойств онтологий / Клещев А.С., Шалфеева Е.А. // Научно-техническая информация, серия 2, 2007, № 6. С. 7-15.

34. Шалфеева, Е.А. Методы и средства разработки жизнеспособных интеллектуальных сервисов / В.В. Грибова, А. С. Клещев, Ф. М. Москаленко, В.А. Тимченко, Е. А. Шалфеева, Л.А. Федорищев // Вестник ДВО РАН Владивосток: Изд. "Дальнаука", 2016. №4 с.133-141.

35. Шалфеева, Е.А. Комплекс средств поддержки процессов разработки и сопровождения решателей для систем с онтологическими базами знаний / Грибова В.В., Шалфеева Е.А. // Информационные и математические технологии в науке и управлении, 2020. № 4(20). С. 34-43.

36. Шалфеева, Е.А. Парадигма автоматизации интеллектуальной профессиональной деятельности. Часть 1. Особенности интеллектуальной профессиональной деятельности / А.С. Клещев, М.Ю. Черняховская, Е.А. Шалфеева // Онтология проектирования. Самара: “Новая техника”, 2013. № 3(9). 53-69.

37. Шалфеева, Е.А. Парадигма автоматизации интеллектуальной профессиональной деятельности. Часть 2. Парадигма автоматизации отрасли/ А.С. Клещев, М.Ю. Черняховская, Е.А. Шалфеева // Онтология проектирования. Самара: “Новая техника”, 2013. №4(10). С. 28-40.

38. Москаленко Ф.М., Тимченко В.А., Шалфеева, Е.А. «Административная система (версия 1.0) интернет-комплекса IАСРaaS» Свидетельство о регистрации программы для ЭВМ №2012618861. (Зарегистрировано в Реестре программ для ЭВМ 28 сентября 2012 г.).

39. Москаленко Ф.М., Тимченко В.А., Шалфеева, Е.А. "Инструментальное средство мониторинга состояния программных агентов интеллектуальных многоагентных сервисов". Свидетельство Федеральной службы по интеллектуальной собственности о государственной регистрации программы для ЭВМ № 2013619809. (Зарегистрировано в Реестре программ для ЭВМ 16 октября 2013 г.).

40. Shalfeeva, E. The methods and the IACPAAS platform tools for semantic representation of knowledge and development of declarative components for intelligent systems / Gribova V., Kleschev A., Moskalenko Ph., Timchenko V., Fedorischev L., Shalfeeva E. // *Open Semantic Technologies for Intelligent Systems (Research Papers Collection)*. 2019. Issue 3. P. 21-24. (OSTIS-2019).
41. Шалфеева, Е.А. Классификация для задач, выявляемых при системном анализе интеллектуальной деятельности // *Материалы IV Междунар. научн.-техн. конф. "Открытые семантические технологии проектирования интеллектуальных систем" (OSTIS-2015) – Минск: БГУИР, 2015. С. 187 – 192.*
42. Шалфеева, Е.А. Классификация структурных свойств онтологий // *Искусственный интеллект, 2005, т.3, с. 67-77.*
43. Шалфеева, Е.А. Конструирование облачного медицинского сервиса по технологии IACPaas / Шалфеева, Е.А., Новоселов А.С. // *Евразийское Научное Объединение, 2016, № 5 (17). С. 25-29.*
44. Shalfeeva, E. Modeling of an intellectual problem solver by transformation of Semantic models // *International Journal "Information Models and Analyses. Vol.2, 2013, Number 3, p. 217-222.*
45. Модель управления качеством баз знаний с оценением / Грибова В.В., Шалфеева Е.А. // *сборник «Знания - Онтологии - Теории (ЗОНТ-2019)». Материалы VII Международной конференции. 2019. С. 138-147.*
46. Шалфеева, Е.А. Метод построения проектных представлений интеллектуального решателя задач по моделям начальных стадий жизненного цикла // *Искусственный интеллект («Штучный интеллект»). Донецк: Изд-во ИПИИ "Наука і освіта", 2013. № 4. С. 51-61.*
47. Шалфеева, Е.А. Решение задачи запроса дополнительной информации на основе декларативной темпоральной базы знаний для дифференциальной диагностики / Клещев А.С., Шалфеева Е.А. // *Сборник «Знания - Онтологии - Теории (ЗОНТ-2017). Материалы Всероссийской конференции с международным участием». 2017. С. 6-14.*
48. Шалфеева, Е.А. Инфраструктура IACPAAS для формирования интерпретируемых баз диагностических знаний по заболеваниям произвольной направленности / Грибова В.В., Окунь Д.Б., Петряева М.В., Шалфеева Е.А. // *Национальная конференция по искусственному интеллекту с международным участием. КИИ-2019. Сборник научных трудов: в 2-х томах. С. 81-89.*
49. Шалфеева, Е.А. Сервис подтверждения предварительного диагноза на основе формализованных знаний // *Петряева М.В., Шалфеева Е.А. // сборник «Материалы XII международной научной конференции "Системный анализ в медицине"» (САМ 2018). 2018. С. 50-53.*
50. Шалфеева, Е.А. Системы на основе онтологических баз знаний как основа для создания современных систем искусственного интеллекта // *Грибова В.В., Шалфеева Е.А. // Восемнадцатая Национальная конференция по искусственному интеллекту с международным участием КИИ-2020. Труды конференции. Под ред. В.В. Борисова, О.П. Кузнецова. Москва, 2020. С. 12-19.*

51. Шалфеева, Е.А. Платформа для разработки облачных интеллектуальных сервисов / Грибова В.В., Клещев А.С., Москаленко Ф.М., Тимченко В.А., Федорищев Л.А., Шалфеева Е.А. // Сборник «Пятнадцатая национальная конференция по искусственному интеллекту с международным участием». Труды конференции: в 3 томах. 2016. С. 24-32.
52. Шалфеева Е.А. Онтология диагностики процессов /Грибова В.В., Шалфеева Е.А. // Онтология проектирования. 2019. Том 9. № 4(34). С.449-461.

Личный вклад соискателя

Основные научные результаты, выносимые на защиту, получены автором лично. Личный вклад автора в публикациях с соавторами кратко характеризуется следующим образом: в статьях [20,32] соискателем выполнены выбор задач и принципа классификации, определение обрабатываемой в задачах информации; в публикациях [25,41,51] – принципы обеспечения жизнеспособности, модель и архитектура СБЗ, основанные на онтологии. В статьях [3,4,5,6,7] соискателем сделан вклад в технологию разработки компонентов СБЗ; в публикациях [12,19,21,28,50,52] автору принадлежат формальное представление онтологии знаний для решения задачи и описание онтологического подхода построения компонентов СБЗ; в [29,48] – формальное представление онтологии ПрОбл для создания портала знаний; в статьях [31,45,46] автору принадлежит архитектурно-технологическая модель развития компонентов СБЗ. В статьях [16,18,36,37] соискателем определена модель жизненного цикла СБЗ и требования к результатам представления начальных этапов разработки; в [8,12, 23,27] - механизмы монотонного усовершенствования БЗ и управления качеством, в [11,13, 30] - алгоритм решения задачи и модель использования вычислительных операций для реализации метода решения; в [24,35] - метод архитектурного проектирования решателей через композицию из повторно-используемых модулей и операций; в [14,33] - классификация свойств онтологий, определение графовых моделей и составление каталога свойств, в [9,22] – модели мониторинга информационных ресурсов интеллектуальной системы. В инструменте [38] соискателем разработаны требования к организации коллективной разработки. В работах [26,39] автором определены графовые структуры для оценивания качества компонентов СБЗ и представления результатов тестирования, а также выполнено архитектурное проектирование инструментов тестирования.

Шалфеева Елена Арефьевна

Методы, модели и технология обеспечения жизнеспособности интеллектуальных систем с декларативными базами знаний

Автореферат
диссертации на соискание ученой степени
доктора технических наук

Подписано в печать 16.09.21
Формат 60x84/16

Усл.п.л. 2.1
Тираж 100

Уч.-изд.л. 1.8
Заказ 2

Федеральное государственное бюджетное учреждение науки
Институт автоматизации и процессов управления
Дальневосточного отделения
Российской академии наук (ИАПУ ДВО РАН)

Отпечатано участком оперативной печати ИАПУ ДВО РАН
690041, г. Владивосток, ул. Радио, 5.
Email: director@iacp.dvo.ru
Сайт: www.iacp.dvo.ru