

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ФГБОУ ВО НГТУ

На правах рукописи



МАКАРОВА ЕКАТЕРИНА СЕРГЕЕВНА

**МОДЕЛИ И АЛГОРИТМЫ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ
ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ ИТ-КОНСУЛЬТИРОВАНИЯ НА
ОСНОВЕ МЕТОДА ПРЕЦЕДЕНТОВ**

Специальность 05.13.10 – Управление в социальных и экономических системах
(технические науки)

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель: д.т.н., профессор, Авдеенко Татьяна Владимировна

Новосибирск – 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1. УПРАВЛЕНИЕ ЗНАНИЯМИ И МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ.....	11
1.1. Управление знаниями в социальных и экономических системах	11
1.1.1. Управление социальными и экономическими системами.....	11
1.1.2. Актуальные проблемы в области управления знаниями	18
1.1.3. Концепция управления знаниями в социальных и экономических системах	21
1.2. Управление знаниями в области ИТ-консультирования	25
1.2.1. Управление знаниями для качественной информационной поддержки пользователей в области ИТ-консультирования.....	26
1.2.2. Обзор существующих систем информационной поддержки в области ИТ-консультирования	28
1.3. Модели и методы инженерии знаний	36
1.3.1. Обзор развития моделей представления знаний.....	36
1.3.2. Прецедентная модель.....	42
1.3.3. Онтологическая модель.....	44
1.4. Выводы	48
ГЛАВА 2. ГИБРИДНАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ ИТ-КОНСУЛЬТИРОВАНИЯ.....	49
2.1. Построение онтологии в области ИТ-консультирования	49
2.1.1. Формальная модель онтологии предметной области.....	49
2.1.2. Реализация онтологии в Protégé OWL	55
2.2. Описание класса Precedent и его реализация	70
2.3. Интеграция прецедентов с концептами онтологии	75
2.4. Выводы	81

ГЛАВА 3. МЕТОД ИЗВЛЕЧЕНИЯ РЕЛЕВАНТНЫХ ПРЕЦЕДЕНТОВ НА ОСНОВЕ ГЕНЕРИРОВАНИЯ СИСТЕМЫ НЕЧЕТКИХ ПРАВИЛ КЛАССИФИКАЦИИ	82
3.1. Цикл управления знаниями на основе интеграции прецедентной и нечеткой моделей	82
3.2. Интеграция прецедентной и нечеткой модели	87
3.3. Алгоритм классификации прецедентов на основе генерирования системы нечетких правил	91
3.4. Экспериментальные исследования алгоритма классификации прецедентов	104
3.5. Выводы	112
ГЛАВА 4. РЕАЛИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИИ В ОБЛАСТИ ИТ- КОНСУЛЬТИРОВАНИЯ	114
4.1. Описание деятельности ИТ-подразделения	114
4.2. Структура и описание СППР для консультантов ИТ-отдела	120
4.3. Описание прецедента в СППР	125
4.4. Экспериментальные исследования гибридной модели на наборе прецедентов в области ИТ-консультирования	128
4.5. Выводы	136
ЗАКЛЮЧЕНИЕ	138
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	140
ПРИЛОЖЕНИЯ	160
ПРИЛОЖЕНИЕ А	161
ПРИЛОЖЕНИЕ Б	165
ПРИЛОЖЕНИЕ В	171
ПРИЛОЖЕНИЕ Г	176

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, АББРЕВИАТУР И ТЕРМИНОВ

CBR – Case-based reasoning

БЗ – база знаний

СОЗ – система, основанная на знаниях

ИИ – искусственный интеллект

ИС – информационная система

ИСППР – информационная система поддержки принятия решений

СППР – система поддержки принятия решений

ЭО – экономический объект

ТАУ – теория автоматического управления

ИТ – информационные технологии

ФЗ – формализованные (явные) знания

НФЗ – неявные знания

НД – набор данных

ПрО – предметная область

ПО – программное обеспечение

ОВ – обучающая выборка

ВВЕДЕНИЕ

В связи с частыми изменениями законодательства, быстрым появлением новых интерфейсов и сервисов сотрудники организаций испытывают проблемы с использованием программных продуктов при выполнении профессиональных задач. Для помощи в решении подобных проблем традиционно используются ИТ-отделы, в которых работают аналитики, консультанты и программисты, занятые поддержкой пользователей в различных предметных областях. Для того чтобы корректно ответить на вопросы пользователей, консультанту необходимо определить сферу возникновения ошибки, проанализировать первичную информацию и, используя личный опыт и (или) справочные материалы, сформулировать ответ на вопрос. Проведенный нами анализ показывает, что среднее время, затрачиваемое на принятие решения консультантом-новичком и опытным экспертом отличается в 2-3 раза при одинаковой сложности задачи. В то же время использование даже очень простых средств записи и извлечения знаний о решении подобных проблем в прошлом (бумажные носители, Microsoft Word) позволяет значительно повысить эффективность неопытного консультанта. Именно поэтому актуальным является построение системы управления знаниями, помогающей накапливать, систематизировать и эффективно использовать опыт консультантов по решению ИТ-проблем сотрудников организации.

Концепция «управление знаниями» в последнее время является одной из активно обсуждаемых тем среди теоретиков и практиков управления в социальных и экономических системах. Управление знаниями обеспечивает интегрированный подход к созданию, организации, доступу и использованию информационных ресурсов организации, что дает компании непосредственный финансовый результат, если знания удастся применить для создания более эффективных и рациональных процессов.

Эффективное управление знаниями в информационном обществе предполагает использование программных систем. В области ИТ-консультирования для автоматизации обработки запросов клиентов в настоящее время используются системы HelpDesk (ServiceDesk), основным

компонентом которых является система управления инцидентами. Кроме того, системы типа HelpDesk могут включать базу клиентов, базу договоров на обслуживание, базу продуктов, и даже базу знаний для готовых решений. Однако основная задача таких систем - контроль исполнения заявок, а не помощь консультантам в решении вопросов пользователей. Их база знаний направлена на передачу клиентам инструкций, поиск готовых решений с учетом контекста проблемы и особенностей предметной области не организован, а проблему по сохранению знаний консультантов и их использованию в нужной ситуации системы HelpDesk не решают. Следовательно, создание системы управления знаниями в области ИТ-консультирования на основе разработки и использования методов инженерии знаний, является актуальной целью исследования.

Цель работы – разработка моделей и алгоритмов для создания информационной технологии интеллектуальной поддержки принятия решений в области ИТ-консультирования на основе эффективной организации и использования ретроспективных, текущих и экспертных знаний.

Для достижения поставленной цели были сформулированы и решены следующие **задачи**:

1. Провести анализ используемой технологии и проблем принятия решений, сделать выбор адекватных моделей представления ретроспективной, текущей и экспертной информации, в области ИТ-консультирования.

2. Разработать гибридную модель представления знаний, интегрирующую базу прецедентов принятия решений с онтологией предметной области, что позволяет определить меру близости прецедента к решаемой проблеме на основе их семантической близости к терминальным концептам онтологии.

3. Разработать подход к извлечению множеств семантически близких прецедентов из базы знаний, интегрированной с онтологией предметной области, на основе алгоритма генерирования нечетких правил классификации. Исследовать эффективность предложенного метода машинного обучения.

4. Разработать интеллектуальную систему поддержки принятия решений в области ИТ-консультирования с использованием базы знаний прецедентов, индексированных концептами онтологии.

Объектом исследования является процесс поддержки принятия решений в сфере ИТ-консультирования, а **предметом исследования** являются методы, модели и алгоритмы для создания новой информационной технологии, а также их практическое применение для поддержки принятия решений.

В качестве методологической основы использовались исследования отечественных и зарубежных учёных в области управления в социальных и экономических системах, прецедентного подхода, онтологического инжиниринга и семантического поиска: Xiong N., Aamodt A., Plaza E., Nilsson M., Sollenborn M., Gruber T.R., Guarino N., Карпова Л.Е., Юдина В.Н., Варшавского П.Р., Новикова А.М., Новикова Д.А., Гавриловой Т.А., Захаровой И.В., Загоруйко Н.Г., Загорулько Ю.А., Когаловского М.Р., Тузовского А.Ф., Массель Л.В., Хабарова В.И., Хорошевского В.Ф., Решты И.В. и др.

Научная новизна исследования состоит в следующем:

1. Предложен подход к созданию новой информационной технологии интеллектуальной поддержки принятия решений в области ИТ-консультирования на основе интеграции метода прецедентов и онтологии с возможностью адаптации методов машинного обучения для извлечения семантически близких прецедентов.

2. Разработана гибридная модель представления ретроспективных, текущих и экспертных знаний, интегрирующая онтологию предметной области с прецедентами принятия решений. Предложенная архитектура связи прецедентов с концептами онтологии позволяет реализовать эффективный метод извлечения прецедентов на основе их семантической близости к решаемой проблеме.

3. Разработан алгоритм машинного обучения на основе генерирования системы нечетких правил классификации из данных, полученных в результате формального описания прецедентов, а также их интеграции прецедентов с онтологией предметной области.

4. Создана интеллектуальная система поддержки принятия решений для реализации предлагаемой информационной технологии, позволяющей накапливать, систематизировать, интегрировать и эффективно использовать опыт экспертов.

Положения, выносимые на защиту:

1. Гибридная модель организации базы знаний, интегрирующая онтологию предметной области с базой прецедентов принятия решений (соответствует п.10 паспорта специальности 05.13.10).

2. Алгоритм извлечения нечетких правил классификации из данных о семантической близости прецедентов с концептами онтологии (соответствует п.6 паспорта специальности 05.13.10).

3. Информационная технология интеллектуальной поддержки принятия решений в области ИТ-консультирования, построенная в результате интеграции метода прецедентов, онтологии предметной области и разработанного метода извлечения семантически близких прецедентов на основе нечеткой логики (соответствует п.12 паспорта специальности 05.13.10).

4. Интеллектуальная система поддержки принятия решений, реализующая предлагаемую технологию (соответствует п.5 паспорта специальности 05.13.10).

Теоретическая значимость работы состоит в развитии гибридных подходов к представлению знаний и соответствующих им моделей, разработке и исследовании методов машинного обучения на основе нечеткой логики, а также создании на их основе информационной технологии сопровождения пользователей программного обеспечения.

Практическая значимость работы. Внедрение разработанной интеллектуальной системы поддержки принятия решений позволило повысить эффективность работы ИТ-подразделений, снизить затраты времени на консультации пользователей при сохранении качества принимаемых решений и удовлетворенности пользователей.

Методы исследования, используемые в работе, базируются на теории управления организационными системами для изменения технологии работы ИТ-отдела, теории принятия решений для решения задачи оказания эффективной консультационной поддержки пользователей, инженерии знаний для создания СППР в области ИТ-консультирования, нечеткой логики для решения задачи классификации прецедентов.

Обоснованность и достоверность научных положений, выводов и рекомендаций определяется использованием моделей и алгоритмов, которые показывают высокую точность как на тестовых, так и на реальных данных; повышением эффективности решения проблем ИТ-пользователей после внедрения разработанного математического и программного инструментария.

Исследования по тематике диссертационной работы велись в рамках гранта Министерства образования и науки «Разработка интеллектуальных технологий, средств компьютерного моделирования и эффективных методов оптимизации, как функционального наполнения информационно-аналитических систем поддержки принятия решений»(№ 01201256088, 2012-13гг.); гранта Министерства образования и науки в рамках проектной части государственного задания «Интеграция моделей представления знаний на основе интеллектуального анализа больших данных для поддержки принятия решений в области программной инженерии» (проект № 2.2327.2017/4.6, 2017-2019 гг.).

В отделах кадров и расчетных группах ФГУП «Охрана» Росгвардии, ООО «ПрофТехУслуги» ЭУ №054 приняты к использованию предлагаемые автором модели, методы и алгоритмы, с помощью которых консультант осуществляет поиск подходящего прецедента для решения проблемы пользователя, что подтверждается актами внедрения.

Интегрированная модель представления знаний, а также принципы создания новой информационной технологии поддержки принятия решений были внедрены в учебный процесс в рамках дисциплин «Системы поддержки принятия решений», «Инженерия знаний и интеллектуальные системы», «Интеллектуальные технологии в моделировании бизнес-процессов».

Апробация результатов диссертации. Основные результаты работы докладывались и обсуждались на конференциях различного уровня: Всероссийской НКМУ «Наука. Технологии. Инновации» (г. Новосибирск, 2014); Международной конференции «Информатика: проблемы, методология, технологии» (г. Воронеж, 2013, 2014); Международной НПК «Будущее науки» (г. Курск, 2014, 2015, 2016); Международной МНК «Молодежь и XXI век» (г. Курск, 2015, 2016); Международной НПК «Современные инструментальные

системы, информационные технологии и инновации» (г. Курск, 2015); Международной НПК «Теоретический и практический взгляд на современное состояние науки» (г. Кемерово, 2015); Всероссийской НПК студентов, аспирантов и молодых ученых (г. Юрга, 2015); International forum on strategic technology (IFOST, г. Новосибирск, 2016); Международной НТК «Актуальные проблемы электронного приборостроения» (АПЭП, г. Новосибирск, 2016); Международной конференции «Информационные технологии в бизнесе и промышленности» (г. Томск, 2016); XII International symposium intelligent systems INTELS 2016 (г. Москва, 2016); III международной конференция и молодёжной школе ИТНТ-2017, приуроченной к 75-летию Самарского университета в СНИУ им. Академика С.П. Королева (г. Самара, 2017); 2nd International Conference on Data Mining and Big Data DMBD 2017 (Fukuoka, Japan 2017).

Публикации: по теме диссертации автором опубликовано 22 работы, в том числе 3 публикации в ведущих рецензируемых научных журналах и изданиях, входящих в перечень рекомендованных ВАК [1,3,41], 5 публикаций в изданиях, индексируемых Scopus и Web of Science [113,114,115,116,156], 2 свидетельства о регистрации программы для ЭВМ [87,88], 12 публикации в сборниках материалов международных научно-практических конференций [43,44,45,46,47,48,49,50,51,52,55,89].

Личный вклад автора. Автором была разработана онтология предметной области, структура представления прецедента, реализована гибридная модель, интегрирующая онтологию с базой прецедентов, разработан и исследован алгоритм классификации с использованием нечеткой логики, создана интеллектуальная СППР, предложена и апробирована новая информационная технология работы с заявками пользователей.

Структура и объем работы. Диссертация состоит из введения, четырех глав, выводов по каждой главе, заключения, библиографического списка из 192 наименований, 4 приложений. Основной текст изложен на 139 страницах, в том числе 62 рисунка и 20 таблиц.

ГЛАВА 1. УПРАВЛЕНИЕ ЗНАНИЯМИ И МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В главе рассматриваются различные подходы к понятию управление, описана область управления знаниями в экономических системах, представлен аналитический обзор моделей представления знаний, существующих информационных систем принятия решений в области ИТ-консультирования. Обоснована необходимость разработки интеллектуальной системы поддержки принятия решений в области ИТ-консультирования.

1.1. Управление знаниями в социальных и экономических системах

1.1.1. Управление социальными и экономическими системами

С древнейших времен люди хотели использовать предметы и силы природы в своих собственных целях, то есть управлять ими. Задачи управления связаны с техническими системами – станками, самолетами, автомобилями. Например, нужно поддерживать заданный курс самолета, обороты двигателя и т.п. Если задачи решаются без участия человека, то говорят об автоматическом управлении.

Теория управления отвечает на вопрос «каким образом нужно управлять?». Наука об управлении зародилась в XIX веке, хотя системы автоматического управления существовали и до XIX века, например, ветряные мельницы. Активное развитие теории управления началось в период промышленной революции. Сначала теория управления в науке разрабатывалась для решения задач регулирования, то есть поддержания заданного значения в технических устройствах (например, в паровых машинах). Отсюда происходит название «теория автоматического регулирования».

Основу теории автоматического управления заложили четыре фундаментальные работы: Максвелл Дж. «О регуляторах» (механизмах наведения телескопа) [160]; Вышнеградский И.А. «Об общей теории регуляторов» [57]; Вышнеградский И.А. «О регуляторах прямого действия» [57]; Ляпунов А.М. «Общая задача об устойчивости движения» [38]. Большой вклад в теорию автоматического управления внесли отечественные ученые:

Петров Б.А., Попов Е.П., Красовский А.А., Пospelов Г.С., Шаталов А.С., Солодовников В.В., Кухтенко А.И., Фельтбаум А.А., Куневич В.М., Пугачев В.С., Болтянский В.Г. и многие другие.

Теория автоматического управления (ТАУ) – это один из разделов кибернетики, связанный с техническими системами. В этой науке было доказано, что управление является свойством любой системы, поэтому теорией в данной науке является общая теория управления, которая не связана непосредственно ни с одной предметной областью, и в то же время может применяться к любой области [30]. Объектом исследования является система. Кибернетики считают, что система – это любой комплекс динамически связанных элементов.

Было сформировано несколько концепций общей теории систем. Их можно разделить на 4 основные группы:

1. К первой группе относятся определения, в которых понятие «система» рассматривают как материальный объект, представляющий единство, т. е. целостность, части которой взаимодействуют друг с другом. Такого взгляда придерживаются Черри К., Дистофано С., Уилсон М. и др.

2. Ко второй группе относятся определения системы как материального объекта, которая представляет собой целостность, состоящую из частей, взаимодействие которых оценивается посредством некоторых входов информации, вещества, энергии и выходов – преобразованной информации, вещества, энергии. Например, «система – это собрание сущностей или вещей, одушевлённых или неодушевлённых, которое воспринимает входы и действует согласно им для производства некоторых выходов, преследуя при этом цель максимизации определённых функций входов и выходов» [100]. В этом случае исследователями не рассматривается процесс преобразования информации, вещества, энергии. Такого подхода придерживаются Клир Дж., Валах и другие.

3. В третьей группе система представляется как математическая модель. Авторами такого типа определений являются Кребер Г., Фриман Н., Месаревич

М., Раппопорт А., Заде Л., Дезоер Ч., Тяхтин В.С., Урсул А.Д., Садовский В.Н., Урманцев Ю.А. и др.

4. В четвёртой группе система рассматривается как совокупность действий, а не как совокупность материальных объектов. Такого взгляда придерживаются Сенгупта С., Акофф Р., Эмери Ф.Х

В работе [62] даны лаконичные определения понятия «система». Система – это упорядоченно-устойчивая самоуправляемая и управляемая целостность, или система – это целостность, которая поддерживает упорядоченно устойчивое состояние своей и вышестоящей системы посредством самоуправления и управления.

Управление – это процесс планирования, организации, мотивации и контроля, необходимый для того, чтобы сформулировать и достичь цели организации [60].

Управление – процесс систематического, сознательного, целенаправленного воздействия управляющей системы на объект управления в целом или его отдельные звенья на основе познания и использования присущих этому объекту закономерностей и прогрессивных тенденций в интересах обеспечения его эффективного функционирования и развития [70]. Существует и множество других определений, в соответствии с которыми управление определяется как: процесс, воздействие, реализация, действие и т.п.

Управление экономическими системами – это управление сложными экономическими объектами (ЭО) в условиях неопределенности [103]. Под ЭО будем понимать объект любой природы (предприятие, организация, фирма, подразделение), который представляет собой сложную, динамичную и управляемую систему [79].

Управление ЭО строится на применении математического моделирования и компьютерных технологий. Повышение эффективности сложных ЭО возможно за счет оптимизации и автоматизации поведения ЭО, т.е. управления ЭО. Автоматизация управления осуществляется за счет создания и внедрения автоматизированных систем для оптимального управления ЭО.

Проблемой управления экономическими системами занимались: Пирсон Б., Мескон М.Х., Новиков Д.А., Новиков А.М., Квятковская И.Ю., Астахова Н.И., Москвитина Г.И.

Система управления представляет собой совокупность объекта управления и субъекта управления [4,27,68]. Приведенная на рис. 1.1 так называемая входо-выходная структура является типичной для теории управления, изучающей задачи управления системами различной природы. Субъектом управления является управленческий аппарат, который объединяет сотрудников, разрабатывающих планы деятельности, требования к управленческим решениям, а также контроль их выполнения, который осуществляют начальники филиалов, подразделений, направлений, проектов и т.п. Под объектом управления подразумевается какое-либо подразделение, филиал, на которое оказывается управленческое воздействие. В задачу объекта управления входит реализация деятельности, для которой создавалась система управления, т.е. выполнение планов, которые разработал управленческий аппарат.

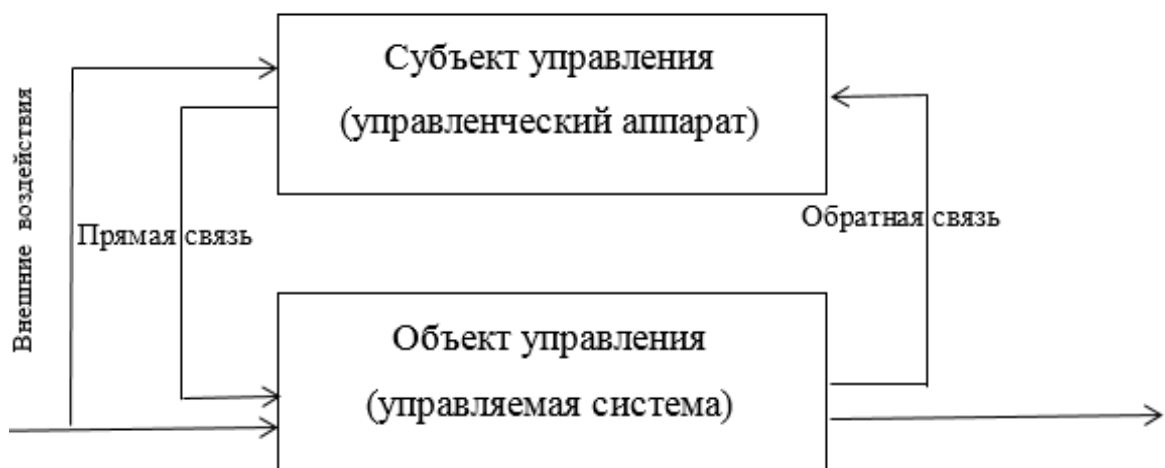


Рисунок 1.1. Структура системы управления

Субъект и объект управления связаны прямой и обратной связью. Прямая связь выражается директивной информацией, которая направляется от субъекта к объекту управления (ежемесячные планы, планы развития и т.п.). Обратная связь выражается отчетной информацией, выполненными принятыми

решениями, идущими в обратном направлении (отчеты за месяц, квартал, год и т.д.).

Информация формируется объектом управления и отражает внутреннюю ситуацию объекта, а также отражает степень влияния на информацию внешней среды. Внешняя среда влияет и на объект управления (поставляет информацию) и на субъект управления (управленческий аппарат), решения которого зависят от внешних факторов (состояние рынка, изменения законодательства, налоговая политика и т. д.).

Новиков Д.А. в своих работах [8,68,69,70,71,72,73] рассматривает управление как вид практической деятельности (см. рис.1.2), описывает субъект и объект системы управления, как набор компонентов практического вида деятельности.

Данная структура применяется для любого вида управленческой деятельности. Управляющий орган спускает необходимые планы развития, планы оптимизации, а, непосредственно, процесс управления выполнением этих планов ложится на управляемый объект. Иными словами, исполнитель сам занимается планированием работы, постановкой целей, разработкой технологий выполнения заданий, оценивает результаты работы и периодически отправляет отчеты в управляющий орган.

Систему управления можно рассмотреть с точки зрения системного анализа. Применение системного анализа для исследования социально-экономических объектов представлено в работах Акоффа Р., Клиланда Д., Берталанфи Л., Кинга В., Форрестера Дж., а также в трудах отечественных исследователей Блауберга И., Богданова А., Волковой В., Глушкова В., Денисова А., Мосейко В., Садовского В., Черняка Ю., Юдина Э.

С точки зрения системного анализа любая система задается перечислением следующих её компонент: состава, структуры и функций. Значит и любая организация определяется заданием:

- состава (сотрудники и подразделения организации);

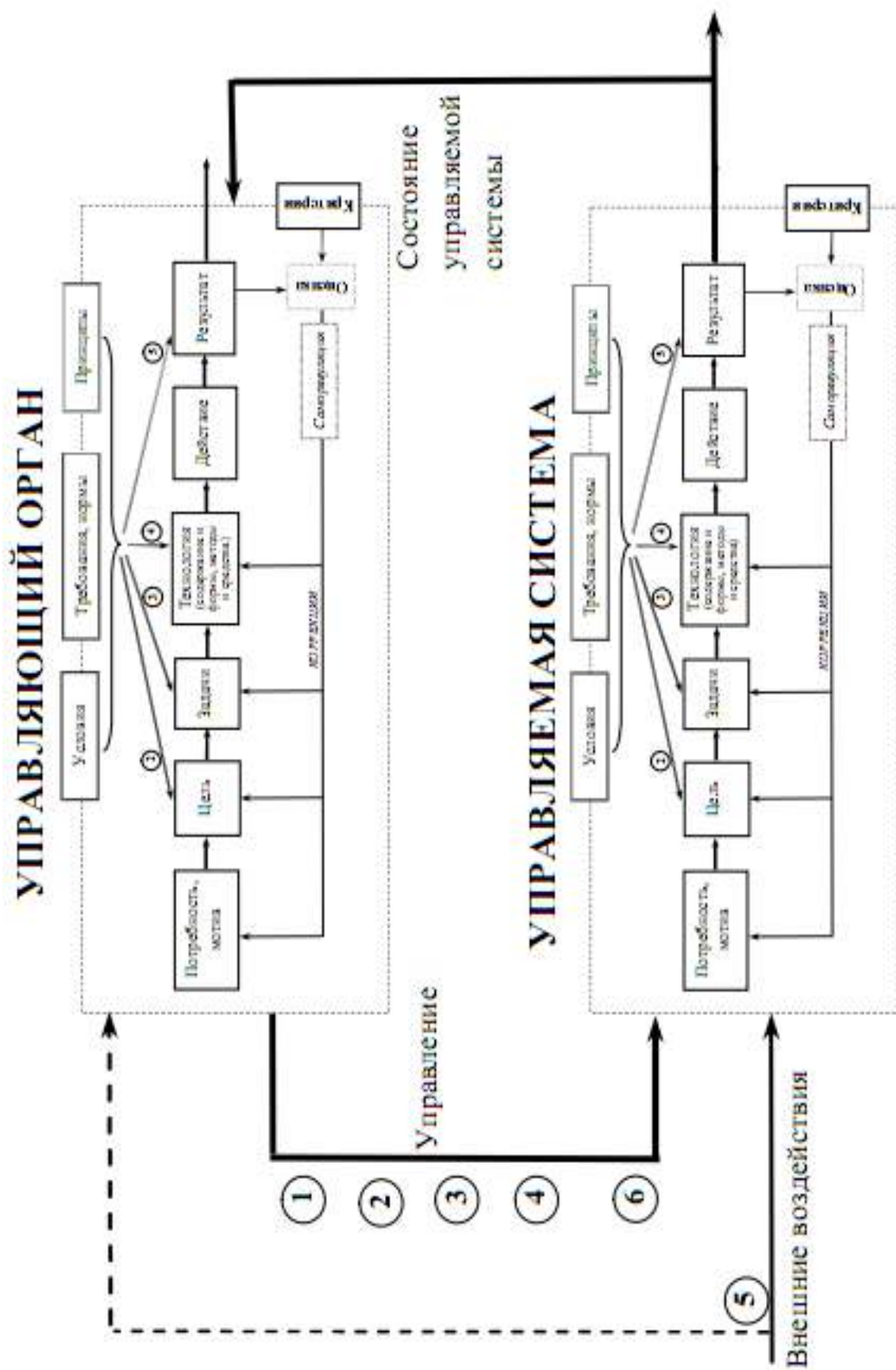


Рисунок 1.2. Структура управленческой деятельности (по Новикову)

- структуры (совокупности различных видов связей между участниками организации, например, информационные, технологические связи);
- функции (различные функции, которые необходимо выполнять организации для достижения поставленных целей).

В задачи управления входят: управление составом, управление структурой, институциональное управление, мотивационное управление, информационное управление (управление информацией, которой имеют право пользоваться сотрудники) (см. рис.1.3) [68].



Рисунок 1.3. Классификация задач управления

Рассмотрим каждую задачу управления в отдельности. *Управление составом* – это определение необходимого состава сотрудников для достижения поставленных целей организации. Эта задача решается вместе с задачей *управления структурой*, т.е. на этапе решения задачи создается или изменяется структура предприятия. Каждому сотруднику определяется должность и набор функций, которые он будет выполнять в ходе своей работы. *Институциональное управление* является наиболее жестким и заключается в том, что центр целенаправленно ограничивает множества возможных действий и результатов деятельности агентов. Такое ограничение может осуществляться явными (правовые акты, распоряжения, приказы) или неявными (морально-

этические нормы, корпоративная культура) воздействиями. Необходимо продумать мотивацию сотрудников, методики стимулирования качественного выполнения работы, и поощрять стремление сотрудников к развитию, как материальными, так и нематериальными способами. Решение такого типа задач происходит на этапе *мотивационного управления*. Важным является решение задач информационного управления, одной из главных задач является задача по ограничению прав доступа к определенному типу информации. Чем крупнее компания, тем тщательнее нужно продумывать права доступа каждого сотрудника к определенной информации.

В настоящее время, одной из ключевых задач управления на предприятии становится задача управления знаниями. Управление знаниями (англ. knowledge management) – это систематические процессы, благодаря которым знания, необходимые для успеха организации, создаются, сохраняются, распределяются и применяются [16,84]. Сегодня база знаний – это самый важный актив современной организации [97-99].

Проблемой управления знаниями занимались ученые: Румизен М.К., Глушкова В.М., Шрейдер Ю.А., Вииг К., Гиляревский Р.Ф., Козачкова Л.С., Ньюэлл К., Смит Д., Гапоненко А.Л., Нонаки И., Такеучи Х., Мильнер Б.З., Тузовский А.Ф., Чириков С.В., Ямпольский В.З.

Проблема управления знаниями в настоящее время активно исследуется учеными, что подтверждается большим количеством публикаций по этой теме [9,13,25,58,75,93,95,96,101]. В следующих подразделах опишем результаты исследований по управлению знаниями в экономических системах.

1.1.2. Актуальные проблемы в области управления знаниями

Учеными было выделено три этапа развития человеческого общества: доиндустриальная экономика, индустриальная экономика, постиндустриальная экономика. В 70-е годы XX века была описана теория постиндустриальной экономики, которая в XXI веке активно уточняется на основе теории информационного общества. Теория информационного общества базируется на необходимости формирования на основе возросшей или возрастающей роли

знаний, используя такие термины как «the knowledgeable society», «knowledge society» или «knowledge-value society». Таким образом, в настоящее время наблюдается переход от постиндустриальной экономики к экономике, основанной на «производстве» знаний. Приведем проблемы в экономике, которая основана на знаниях.

1. В стоимости товаров и услуг знание формирует большую часть этой стоимости. Например, стоимость автомобиля до 70 % сформирована интеллектуальными компонентами: дизайн, электроника и т.п. В стоимости программного обеспечения «степень участия» знаний еще выше.

2. Формирование, хранение и использование знаний в деятельности организаций приобретает большое значение. В связи с этим, системе образования отводится особая роль. Инвестиции в образование – вклад в человеческий капитал. Профессиональное образование формирует большую долю в структуре затрат государства.

3. Знания быстро устаревают. Прикладные знания, полученные выпускником вуза, особенно в области инновационных технологий становятся неактуальными к моменту получения диплома. В связи с этим, актуальным становится требование работодателей – обладать навыками профессиональной и социально-психологической адаптации в быстро меняющемся мире. Таким образом, образование должно стать процессом, длящимся в течение всей трудовой жизни человека.

4. Количество работников, занятых производством, хранением, передачей и использованием знаний в структуре населения возрастает. В обществе знаний возрастает конкуренция. Количество людей, которые могут добиться успеха, увеличивается за счет доступности знаний. Для выполнения большого числа работ требуется высокий уровень квалификации и соответствующий уровень знаний. Владение знаниями создают конкурентные преимущества и работодателям, и самим работникам, т.к. основаны на понимании собственного значения и способности превращать знания в материальную выгоду.

5. Предприятия, функционирующие в начале XXI века, приспосабливаются к новым тенденциям в экономике и организации производства, препятствующим чрезмерной специализации. Роль малого и среднего бизнеса позволяет в максимальной степени адаптироваться к меняющимся индивидуальным запросам конкретных потребителей, и позволяет создавать инновационные товары и услуги до возникновения на них покупательского спроса. В связи с этим предъявляются высокие требования к уровню профессионализма работников, которые должны владеть не только знаниями для выполнения операций одновременно в нескольких специальностях, но и навыками анализа, прогнозирования, обсуждения проблем в группах и пр. В связи с этим задачей является развитие навыков преобразования неявных знаний в явные (для того, чтобы они стали достоянием организации), а затем – обратной трансформации сформированных в результате совместной деятельности «общих знаний» работников – в неявную (процедурную) форму, используемую для выполнения работниками конкретных профессиональных задач [22].

Успешное функционирование организаций в экономике знаний предполагает развитие и поддержку ими различных механизмов управления знаниями. Под организационно-экономическим механизмом управления знаниями в социально-экономических системах следует понимать совокупность частных механизмов управления знаниями – планирования, организации, мотивации и контроля процессов производства, распределения, использования и трансфера знаний [165]. Частные механизмы управления знаниями интегрируются в общий организационно-экономический механизм управления знаниями. В свою очередь, организационно-экономический механизм управления знаниями является частью системы управления знаниями – ее движущей силой, приводящей систему в эффективное функционирование и развитие.

В качестве организационной части механизма управления знаниями выступает совокупность функций, организационных методов, инструментов,

технологий, рычагов воздействия проектной группы по управлению знаниями на комплекс процессов производства, распределения, использования и трансфера знаний, позволяющий осуществить управление знаниями с учетом особенностей целей предприятия в рамках существующей системы управления.

В качестве экономической составляющей механизма управления знаниями могут быть финансовое обеспечение деятельности по управлению знаниями; мотивация сотрудников, участвующих в этом процессе; экономические методы воздействия; экономические критерии оценки эффективности организации управления знаниями. Система управления знаниями является одной из подсистем в общей системе управления предприятием как социально-экономической системой.

1.1.3. Концепция управления знаниями в социальных и экономических системах

Концепция управления знаниями (knowledge management) появилась в 1990-х годах. Управление знаниями – это организация и систематизация информации и знаний в компании [33]. Однако, это достаточно общее определение, которое не показывает границы этого термина. На этапе становления систем управления знаниями, Томас Давенпорт дал следующее определение «Управление знаниями – это процесс сбора, распространения и эффективного использования знаний» [33]. Несколько лет спустя Gartner Group предложила более подробное определение: «Управление знаниями – это система, которая предполагает интегрированный подход к поиску, сбору, оценке, восстановлению и распространению всех информационных активов предприятия. В состав таких активов могут входить базы данных, документы, политики, процедуры, а также знания и опыт отдельных работников, которые ранее не фиксировались» [34].

У этих определений очень организационная, корпоративная природа – ведь исторически системы управления знаниями формировались именно в организациях. Основная суть системы управления знаниями в сборе и фиксации

информации и знаний, которыми владеют сотрудники, и в распространении этих знаний между всеми членами коллектива.

В условиях быстрого развития экономики, основанной на знаниях, само понятие «знание» подвергается существенной трансформации. Поэтому при рассмотрении вопросов управления знаниями важным начальным этапом является прояснение сути концепции «знание».

Общепринятыми положениями, касающимися концепции «знание», рассматриваемыми в различных научных областях, являются: признание значения сознания для приобретения знания, неразделимость знания и его носителя, различие понятий знания и данных, обусловленность знания другими знаниями и т.п. Рассмотрим иерархическую структуру различных уровней знаний, представленную в табл. 1.1.

Таблица 1.1. Уровни знаний

Уровень	Пример (оформление приказа о приеме на работу сотрудника)	Цель
Данные	Элементы: Ф.И.О. сотрудника, дата приема, образование, подразделение, должность, ...	Ничего не знать
Информация	Данные (Ф.И.О. сотрудника, дата приема, образование, подразделение, должность) + запрос к базе знаний о молодых сотрудниках, принятых на работу в прошлом году	Знать - что
Знания	Автоматическое оформление приказа о приеме на работу	Знать - как
Мудрость	Какими навыками должен обладать сотрудник отдела кадров? Почему выводится ошибка программы при проведении приказа о приеме на работу? Как исправить ошибку? Все ли данные занесены в программу сотрудником отдела кадров?	Знать - почему

На первом уровне рассматриваемой иерархии – данные. Данные представляют свойства объектов или рассматриваемых событий. Им свойственно отсутствие какой-либо цели. Данные существуют сами по себе, и до определенного момента не включены в деятельность. Сбор данных также происходит вне зависимости от решения какой-либо задачи.

На втором уровне иерархии – информация. Информации свойственна более конкретная цель, для достижения которой данные объединяются с процедурой их обработки. Информация содержится в ответах на вопросы, начинающиеся со слов «что», «кто», «где», «когда», «сколько» и т.п. Пример – простые информационные запросы к базам данных.

На третьем уровне иерархии – знания. Знания передаются через обучение и отвечают на вопрос «как». Знания, в отличие от информации, соответствуют не только наличию ингредиентов и рецепта, но и умению объединить полученную информацию, скоординировать действия для получения конкретного продукта.

Четвертым уровнем иерархии является мудрость, соотносимая с ответом на вопрос «почему». Понятие мудрости можно трактовать как «метазнания», т.е. знания о знаниях. В когнитивной психологии используется более широкая интерпретация понятия «мудрость». Мудрость принято связывать со способностями человека: достигать положительный результат за счет баланса различных интересов; синтезировать новые знания с учетом противоположных точек зрения; осознавать собственные ошибки и ограниченность собственных знаний; выявлять и формулировать проблемы (в результате творческого мышления) [116].

Традиционные системы имеют дело с так называемым формализованным (явным) знанием ФЗ (знание что, знание о ...). Формализованные знания доступны всем членам организации, они представлены в письменной форме, их можно передавать другим людям в форме документов, инструкций, учебных пособий и т.д. Примеры явных знаний: модели, математические формулы, научные статьи, формализованный технологический процесс, правила техники безопасности, должностные инструкции, патенты, лицензии, данные в компьютерных базах данных и т.п.

Современный этап развития экономики знаний характеризуется все возрастающей ролью неформализованных, так называемых «неявных», знаний НФЗ. Неявные знания – это знания, полученные с опытом и являющиеся значимыми для решения индивидом практических проблем. Неявное (скрытое)

знание основано на личном опыте, практических приемах, интуиции, включает личные ноу-хау и объяснительные субъективные модели, и поэтому с трудом поддается формализации. Неформализованное знание создается путем систематизации собственного личного опыта человека, его целенаправленных усилий в окружающем мире.

В отличие от традиционного формального образования, когда человек получает информацию в виде готовых формальных схем, неявные знания формируются в проблемных ситуациях, в которые человек поставлен в связи с принятием решения по конкретной проблеме. В отличие от формализованных (декларативных) знаний, отвечающих на вопрос «что», неформализованные (процедурные) знания отвечают на вопрос «как». Процедурные знания не только помогают в достижении целей, но и стимулируют субъекта на приобретение новых знаний, способствующих решению жизненно важных проблем.

Взаимодействие между сотрудниками организации по решению важных проблем порождает процесс создания нового знания. В этом процессе трансформации знания задействованы обе его формы: неформализованное и формализованное знание. В процессе создания нового знания неформализованное и формализованное знание взаимодействуют четырьмя способами: социализация; экстернализация, комбинация, интернализация Нонака и Такеучи (Nonaka and Takeuchi) [165], как показано на рис.1.4.

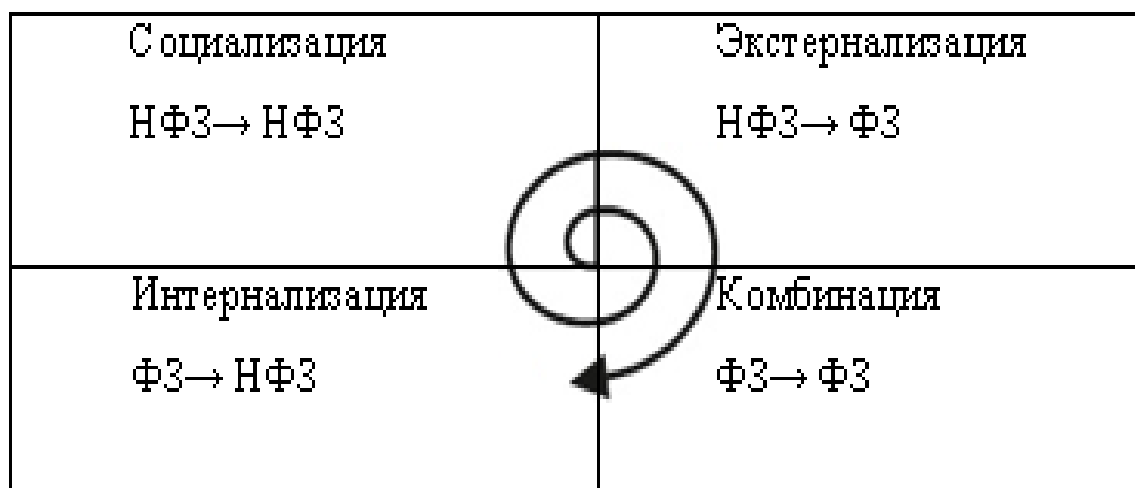


Рисунок 1.4. Спираль знаний Nonaka и Takeuchi

Создание знаний есть непрерывный процесс динамического взаимодействия явного и неявного знаний. Данный процесс поддерживается спиралью знаний, последовательно проходя через четыре указанные стадии.

Социализация – процесс восприятия человеком опыта от других людей без использования слов. Социализация, осуществляется, например, в результате наблюдения за деятельностью профессионала.

Экстернализация – процесс оформления неформализованного знания в формализованное (модели, гипотезы, инструкции, технологии, и т.д.). Подобное знание может быть сформировано, например, аналитиком (инженером по знаниям) с применением методов извлечения знаний из экспертов и других источников. Подобное знание может быть полуформализованным (тексты, графики, поле знаний), а может быть полностью формализованным (представленным на языке представления знаний в базе знаний интеллектуальной системы).

Комбинация – процесс встраивания полученных знаний в существующую модель формальных знаний.

Интернализация – формализованное доступное всем знание преобразуется в неформализованную модель знаний человека, и тем самым приобретает для него ценность.

Последовательное чередование четырех процессов – социализация, экстернализация, комбинация, интернализация – создает спираль знаний. Центральная задача менеджеров, таким образом, заключается в том, чтобы обеспечить эффективное функционирование этой спирали. Нам представляется перспективной разработка поддержки спирали знаний на основе интеллектуальных информационных технологий.

1.2. Управление знаниями в области ИТ-консультирования

В современном мире деятельность любой ИТ-компании можно рассматривать как сложный процесс, который может быть максимально эффективен только благодаря качественному управлению знаниями. Во всех компаниях имеется огромный набор данных и практического опыта, но эта

информация рассредоточена в различных источниках и эффективное ее использование без современных информационных технологий невозможно. Создание интеллектуальных информационных систем, способных накапливать знания и использовать их при принятии решений является ключевой составляющей управления знаниями в области ИТ-консультирования.

1.2.1. Управление знаниями для качественной информационной поддержки пользователей в области ИТ-консультирования

В связи с частыми изменениями законодательства, быстрым появлением новых интерфейсов и сервисов сотрудники организаций испытывают проблемы с использованием программных продуктов. Для помощи в решении данных проблем традиционно используются ИТ-отделы, в которых работают консультанты и программисты, занятые поддержкой пользователей в различных областях и по различным программным продуктам.

Основную роль в обеспечении качественной ИТ-поддержки играет управление знаниями консультанта. Во время ежедневного консультирования пользователей задача консультанта сводится к идентификации вопроса от пользователя, анализу наработанной базы знаний исходя из опыта работы с системой, и предложению варианта ответа пользователю. Для того чтобы корректно ответить на вопросы пользователей, необходимо определить сферу возникновения ошибки, проанализировать первичную информацию и, используя личный опыт и/или справочные материалы, сформулировать ответ на вопрос, а затем подтвердить его используя тестовую базу (рис.1.5).

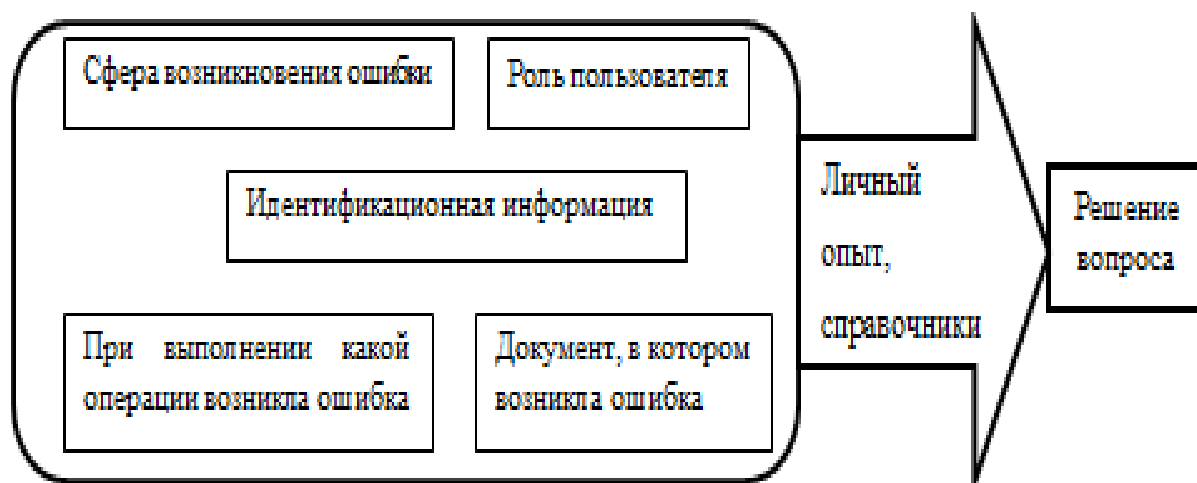


Рисунок 1.5. Схема ответа на вопрос пользователя

Вопросы, которые поступают консультанту от пользователей можно разделить на следующие типы:

1. Простые вопросы – вопросы, на которые при наличии определенного опыта работы с системой (около года) консультант может отвечать сразу. Затраты времени на решение вопроса составляют от 15 до 25 минут.

2. Вопросы средней сложности – вопросы, которые требуют анализа сложившейся ситуации, обращение к системе, анализ работы пользователя, восстановление последовательности действий. Затраты времени на решение вопроса составляют от 25 до 60 минут.

3. Сложные вопросы – вопросы, которые требуют детальной проработки, анализа сложившейся ситуации, не исключено, что решением вопроса в данном случае будет доработка системы с помощью программистов. Затраты времени на решение вопроса составляют от 60 минут до 120 минут.

4. Технические задания (простые) на разработку системы. В связи с изменениями в области законодательства, изменениями во внешнем мире возникает необходимость дорабатывать определенные модули системы. В этом случае, затраты времени варьируются от степени сложности доработки системы, это может быть и 24 часа, и несколько дней, а иногда и месяц.

Написание технических заданий связано с необходимостью совершенствования системы или оптимизации процессов, либо в целом с внедрением нового модуля, т.е. изучением запросов пользователей, описанием бизнес-процессов и их реализацией для решения определенных видов задач.

Среднее время, затрачиваемое консультантом, зависит от опыта его работы и сложности задачи. В табл.1.2 приведены усредненные результаты экспертного оценивания аналитиками времени, затрачиваемого на решение проблем пользователей. Согласно данным таблицы, если неопытный консультант использует предшествующие свои или чужие наработки, то в среднем он затрачивает меньше времени на решение проблемы.

Таблица 1.2. Затраты времени на решение вопроса

Категория	Опытный консультант	Неопытный консультант		
		Самообучение	Использование прецедентов на бумажных носителях без функции поиска	Использование прецедентов с функцией поиска (MS Word)
Простые вопросы	7–10	15–25	12–20	10–17
Вопросы средней сложности	10–25	25–60	20–50	17–45
Сложные вопросы	25–45	60–120	50–110	45–100
Технические задания	> 45	> 120	> 110	> 100

Использование даже очень простых средств записи и извлечения знаний о решении проблем позволяет приблизить эффективность неопытного консультанта к эффективности опытного специалиста [115]. Проанализировав затраты времени на решение вопросов, можно прийти к выводу, что для обеспечения качественной работы ИТ-отдела необходимо сократить время выполнения заявки пользователя. Для этого целесообразно разработать интеллектуальную систему поддержки принятия решений в области ИТ-консультирования, которая будет содержать наработки и опыт консультантов, а также выдавать готовые решения, релевантные запросу пользователя.

В следующем подразделе проанализируем существующие системы информационной поддержки, которые в настоящее время используются в области ИТ-консультирования.

1.2.2. Обзор существующих систем информационной поддержки в области ИТ-консультирования

В настоящее время для информационной поддержки консультантов используются системы по обработке заявок (HelpDesk или ServiceDesk). Такие системы позволяют выявить проблемные участки ИТ-инфраструктуры и оценить эффективность работы ИТ-подразделения.

Системы информационной поддержки позволяют ИТ-подразделениям обеспечить качественное выполнение следующих функций:

- единую точку обращения к службе поддержки;
- стандартный способ регистрации и выдачи заданий специалистам;
- контроль за последовательностью исполнения работ, потраченным временем и ресурсами;
- назначение приоритетов запросам в зависимости от типа запроса, конкретного пользователя или других обстоятельств;
- эскалация запросов и инцидентов, оповещение соответствующих администраторов;
- отчётность по затратам времени и средств на выполнение запросов [90].

Одним из компонентов в системах информационной поддержки может быть база знаний, которая представляет собой инструмент, позволяющий фиксировать полезную информацию (инструкции, руководства по работе и т.д.) и предоставлять к ней общий доступ для всех пользователей системы.

Полезная для бизнеса база знаний в системе информационной поддержки должна обладать следующей функциональностью:

- Хранение и отображение для всех пользователей системы информационной поддержки (включая контактных лиц клиентов в клиентском портале) публичных статей и разделов — для быстрого и самостоятельного решения типовых проблем.
- Хранение и отображение закрытых статей — доступных только для сотрудников сервисной компании (например, внутренние регламенты работы службы технической поддержки);
- Возможность ведения персональных клиентских разделов, в которых будет храниться специфичная информация относительно заказчика (например, схема локальной сети, конфигурация информационной системы или поэтажный план).

Перечисленными функциями не обладает большинство существующих систем информационной поддержки. База знаний в классической системе

информационной поддержки представляет собой иерархию папок с некоторой информацией, как правило, поиск ограничивается выбором соответствующей папки и длительным ручным поиском внутри этой папки необходимой информации, которая может помочь решить проблему пользователя программного продукта. Рассмотрим основные системы информационной поддержки (HelpDesk системы), которые используются в области ИТ-консультирования.

Система информационной поддержки OTRS – это открытая система обработки заявок является веб-приложением, которое можно запускать в окне любого HTML-совместимого веб-браузера. Чтобы система одинаково правильно работала в веб-браузерах мобильных телефонов и других мобильных устройств, веб-интерфейс OTRS не содержит динамического контента. Для успешной работы с OTRS не нужно устанавливать никаких специальных операционных систем; единственное что потребуется – HTML браузер. В табл. 1.3. представлены основные достоинства и недостатки OTRS системы[82].

Таблица 1.3. Достоинства и недостатки системы информационной поддержки OTRS

Достоинства	Недостатки
<ul style="list-style-type: none"> • возможность создавать очереди обращений 	<ul style="list-style-type: none"> • недостаточно проработанный функционал Process Management
<ul style="list-style-type: none"> • личный кабинет для пользователей, для клиентов 	<ul style="list-style-type: none"> • сложность понимания статусов заявок из-за гибкой системы формулирования статусов;
<ul style="list-style-type: none"> • гибкая настройка прав пользователей 	<ul style="list-style-type: none"> • сложность настройки системы
<ul style="list-style-type: none"> • русская локализация (по факту доступно 34 языка) 	<ul style="list-style-type: none"> • недостаточно информации по системе
<ul style="list-style-type: none"> • гибкая система отчетов 	<ul style="list-style-type: none"> • сложность внедрения системы
<ul style="list-style-type: none"> • существует документация и русское сообщество 	
<ul style="list-style-type: none"> • бесплатное ПО 	

Система информационной поддержки GLPI – это система для работы с заявками и инцидентами, система для инвентаризации компьютерного оборудования и ПО, система написана на PHP. Система GLPI может быть

развернута как на открытых операционных системах типа Linux и BSD, так и на коммерческих системах типа Windows.

Основные возможности: развитый механизм управления заявками, обращениями и инцидентами, возможно с привязкой к конкретным конфигурационным единицам; учет компьютеров, периферийного оборудования, сетевых устройств и т.п.; четкое разграничение полномочий пользователей в пределах различных типов активов и организаций на основе профилей; назначение задач, планирование; управление лицензиями, документами, договорами; управление статусом работы оборудования; использование нескольких интерфейсов; резервирование оборудования; генератор отчетов[139]. В табл. 1.4. представлены основные достоинства и недостатки GLPI системы.

Таблица 1.4. Достоинства и недостатки системы информационной поддержки GLPI

Достоинства	Недостатки
• возможность создания шаблона, расширенные подключения устройств	• невозможность закрыть заявку пользователем
• возможно осуществлять поиск в системе по различным параметрам	• сложность со вводом в эксплуатацию системы
• внесение информации происходит быстро	• поиск данных в базе знаний только иерархический
• гибкая система отчетов	• нет поддержки каталога услуг,
• гибкая настройка прав пользователей	• сложность со вводом в эксплуатацию системы;
• существует документация, русифицированный ПП	• поиск данных в базе знаний только иерархический;
• наличие базы знаний	• нет поддержки каталога услуг,
• бесплатное ПО	• планирование и контроль релизов необходимо осуществлять в стороннем ПП;
	• работа в системе требует некоторого изучения специфики системы;
	• невозможно заложить заранее KPI с целью автоматического контроля

Система информационной поддержки vsDesk – это российская разработка, которая имеет как платную версию без ограничений, так и бесплатную с ограничениями. Сейчас существует 3 редакции системы, доступных для скачивания или приобретения: «Старт», «Про» и «Корп».

Версия «Старт» включает в себя ограниченный функционал и предлагает следующие возможности: управление инцидентами, управление сервисами, управление уровнями сервиса, база знаний (без иерархии с категориями), e-mail уведомления и т.п. Версия «Про» включает в себя версию «Старт» плюс добавлены: SMS уведомления, планировщик заявок, простой чат, тонкая настройка ролей, отчеты. Версия «Корп» включает в себя версию «Про» плюс добавлены: AD интеграция, Исходные коды. [182]. В табл. 1.5. представлены основные достоинства и недостатки vsDesk системы.

Таблица 1.5. Достоинства и недостатки системы информационной поддержки vsDesk

Достоинства	Недостатки
<ul style="list-style-type: none"> • простой интерфейс на основе Twitter Bootstrap 	<ul style="list-style-type: none"> • есть ошибки в работе программы, но разработчики быстро исправляют
<ul style="list-style-type: none"> • коробочная версия на русском языке, техподдержка тоже на русском языке 	<ul style="list-style-type: none"> • отсутствие сортировок отображаемых элементов в табличных частях
<ul style="list-style-type: none"> • возможность самостоятельных доработок (Yii в основе и исходники в комплекте) 	<ul style="list-style-type: none"> • в БЗ отсутствует возможность создавать вложенные категории, поиск предусмотрен только ручной
<ul style="list-style-type: none"> • удачная реализация процесса «Управление инцидентами» 	<ul style="list-style-type: none"> • резервное копирование делается вручную
<ul style="list-style-type: none"> • есть самые необходимые для контроля за исполнением отчеты с возможностью экспорта в Excel 	<ul style="list-style-type: none"> • ручное обновление
<ul style="list-style-type: none"> • реализация процесса «Управление проблемами» 	<ul style="list-style-type: none"> • нет возможности редактировать статусы заявок
<ul style="list-style-type: none"> • есть простой конструктор типов активов 	<ul style="list-style-type: none"> • нет возможности редактировать роли пользователей
<ul style="list-style-type: none"> • есть авторизация при помощи учетных записей 	<ul style="list-style-type: none"> • платное ПО

1С:Битрикс – это модуль для системы bitrix. Данный модуль предназначен для организации службы техподдержки и консультирования пользователей на

сайте и по электронной почте. Пользователь может обратиться в службу технической поддержки, заполнив специальную форму на сайте, отправив сообщение по электронной почте, позвонив по телефону, или может создать обращение прямо из сообщений клиента, а к сообщениям прикрепить файлы разных форматов.

Основной функционал: обработка запросов через сайт, e-mail, через модуль почты, форум, по телефону, классификация запросов по категориям, возможность настроить необходимое число статусов обращений, возможность настроить оценки ответов, уровни критичности обращения, выбрать типовой (шаблонный) ответ, назначение ответственных, оценка ответов, история обращений, совместная работа, гибкое распределение прав доступа к модулю техподдержки, гибкий контроль работы службы поддержки [63]. В табл. 1.6. представлены основные достоинства и недостатки системы 1С:Битрикс.

Таблица 1.6. Достоинства и недостатки системы информационной поддержки 1С:Битрикс

Достоинства	Недостатки
• простой понятный интерфейс	• ограниченный функционал
• совместимость с бухгалтерским софтом «1С: Предприятие»	• работа осуществляется через рабочий стол
• возможность закрытия сообщения администратором	• отсутствует база знаний
• возможность оценивания работы консультантов	• достаточно высокая цена на ПП
• справочники категорий критичности с возможностью добавления новых позиций	• требователен к техническим ресурсам пользователей
• разные отчеты для контроля работы консультантов;	
• ставится достаточно быстро, и сразу можно начать пользоваться.	

ServiceDesk «Итилиум» – первое российское решение в области управления ИТ услугами на платформе 1С:Предприятие 8. Итилиум реализует автоматизацию процессов управления ИТ, а также функций службы ServiceDesk. «Итилиум» разработан для использования с платформой 1С:Предприятие 8.

Код системы открыт. «Итилиум» является конфигурацией для платформы 1С: Предприятие. Внедрение ServiceDesk «Итилиум» позволяет: структурировать деятельность, выполняемую ИТ-подразделением представив ее в виде каталога услуг [91]. В табл. 1.7. представлены основные достоинства и недостатки системы информационной поддержки Итилиум

Таблица 1.7. Достоинства и недостатки системы информационной поддержки Итилиум

Достоинства	Недостатки
• низкая стоимость владения;	• достаточно высокая цена на ПП;
• глубокая методологическая проработка продукта;	• требователен к техническим ресурсам пользователей;
• открытый код – возможность самостоятельной адаптации;	• необходима обязательное наличие платформы 1С;
• платформа 1С:8 – самая распространенная платформа;	• база знаний ограниченного функционала;
• продукт постоянно и стремительно развивается;	• качественное внедрение требует хорошего профессионального уровня специалистов по внедрению.
• методическая поддержка;	• достаточно высокая цена на ПП;
• поддержка самостоятельного внедрения;	
• самая продаваемая российская система класса ServiceDesk;	
• лучшее решение по соотношению цена/качество;	
• продукт создан специально для отечественного рынка;	
• сообщество пользователей.	

IntraService – это универсальная российская ServiceDesk система с веб-интерфейсом, позволяющая упростить и автоматизировать учет и обработку заявок. Система может быть настроена только для внутренней работы или с возможностью принимать заявки извне, от клиентов.

Основной функционал: создание заявок пользователем, история всех операций по заявкам, поиск, фильтрация, сортировка, сохранение фильтров, экспорт в Excel заявок, расписание и настройка времени реакции и выполнения заявок по сервисам, автоматическая эскалация заявок, база знаний, модуль

активов, автоматическое наполнение базы данных активов, история изменений, статистика, отчеты [92]. Достоинства и недостатки представлены в табл. 1.8.

Таблица 1.8. Достоинства и недостатки системы информационной поддержки IntraService

Достоинства	Недостатки
<ul style="list-style-type: none"> • настройка бизнес-процессов обработки заявок разного типа 	<ul style="list-style-type: none"> • высокая цена на ПП
<ul style="list-style-type: none"> • учет полного цикла обработки всех типов заявок 	<ul style="list-style-type: none"> • требователен к техническим ресурсам пользователей
<ul style="list-style-type: none"> • контроль и управление качеством обработки заявок 	<ul style="list-style-type: none"> • не предусмотрено резервное копирование
<ul style="list-style-type: none"> • возможность мониторинга трудозатрат по обработке заявок 	<ul style="list-style-type: none"> • система предназначена для работы в пределах одной компании
<ul style="list-style-type: none"> • интерфейсы интуитивно понятные 	<ul style="list-style-type: none"> • механизм поиска в БЗ не позволяет искать по прикрепленным файлам
<ul style="list-style-type: none"> • возможность создание заявки в системе IntraService с произвольной формы на любом стороннем WEB сайте 	
<ul style="list-style-type: none"> • поиск по комментариям к заявке 	
<ul style="list-style-type: none"> • наличие базы знаний 	
<ul style="list-style-type: none"> • гибкие отчеты 	

Итак, мы рассмотрели наиболее используемые системы информационной поддержки в области ИТ-консультирования. Эти системы позволяют упростить и автоматизировать учет и обработку заявок пользователей в ИТ-подразделениях. В продвинутых версиях систем есть раздел «база знаний», который представляет собой иерархический набор директорий, в которых расположена полезная информация для пользователей. Поиск информации во всех системах осуществляется через поиск подходящей директории. Как правило, отсутствует поиск внутри прикрепленных документов. Таким образом, база знаний в существующих системах представляет собой набор документов с большим объемом информации, который не может решить проблему, возникшую у пользователя.

Следовательно, целесообразно разработать интеллектуальную систему поддержки принятия решений в области ИТ-консультирования, которая будет

выдавать готовые решения, которые когда-то ранее были приняты опытным экспертом при решении проблемы, схожей с текущей проблемой пользователя. В основе такой системы должна лежать адекватная модель представления знаний, развиваемая в научной области инженерии знаний. В следующем подразделе представлен обзор моделей и методов представления знаний, проанализированы их достоинства и недостатки.

1.3. Модели и методы инженерии знаний

В настоящее время говорить об эффективном управлении знаниями можно только на основе использования информационных технологий. Однако традиционно используемые на предприятиях информационные системы, построенные на основе накопленных данных, не справляются с такой высокоинтеллектуальной задачей, как управление знаниями. Здесь необходимо применять новые методы и подходы, разрабатываемые в научной области искусственного интеллекта. Модель представления знаний – способ описания знаний в базе знаний. Для реализации интеллектуальной системы поддержки принятия решений в области ИТ-консультирования необходимо выбрать подходящие модели представления знаний.

1.3.1. Обзор развития моделей представления знаний

С середины XX века большие надежды возлагались на исследования, проводимые в области искусственного интеллекта (ИИ) [78,81]. В истории становления ИИ как научной области периоды оптимизма и ожидания быстрых успехов регулярно сменялись застоями, а то и падениями, когда получаемые результаты начинали казаться неубедительными, а вложения в работы по ИИ резко сокращались. Однако возникновение в 1970-е годы экспертных систем и систем, основанных на знаниях (СОЗ), определенно связывают с успехом в данной научной области, когда методы ИИ стали использоваться в реальном секторе экономики.

С момента внедрения первых, так называемых классических систем, основанных на модели представления знаний в виде правил (rule-based reasoning), прошло более 40 лет, однако архитектура подобных систем мало

изменилась. При внедрении таких систем разработчики сталкивались с такими проблемами, как сложность извлечения знаний из экспертов, требование высокой квалификации персонала, сложность обработки больших объемов информации, трудности сопровождения. Для решения этих проблем проводились исследования с целью нахождения лучших методов и инструментов [122,162], в результате чего были разработаны различные оболочки и среды, улучшена методология разработки [133,183,185], разработаны языки моделирования знаний, онтологии [110,123,124,185], методы и средства для систем поддержки принятия решений [119,184].

Однако в 1980-е годы, формируется альтернативная парадигма, которая привлекает все больше и больше исследователей. Рассуждения на основе прецедентов (Case-based reasoning, CBR) – метод, в котором решаются новые задачи, путем адаптации ранее успешного решения аналогичных проблем в прошлом [163].

Работа [175], по распространенному мнению, является истоком прецедентного подхода. В этой работе было предложено обобщить знания о ситуациях, и записывать их в виде сценариев, на основании которых в дальнейшем можно было бы формировать выводы. Сценарии были предложены, в качестве модели мышления для описания событий, стереотипных ситуаций, таких как поход к врачу или в ресторан. Однако эксперименты со сценариями показали, что теория моделей мышления до конца еще не сформирована и направление дальнейших исследований лежит в области философии и психологии. Позднее Шенк продолжал исследовать роль воспоминаний о предыдущих ситуациях (т.е. случаях) и исследовал структуру процесса организации и сохранения предыдущего опыта в виде контейнера знаний, который играл важную роль, как в процессе решения задач, так и в процессе обучения [174]. В это же время, Гентнер [138] разрабатывает теоретическую структуру аналогий, которые играют важную роль в прецедентном подходе. В исследованиях [186] можно найти ссылки о том, что естественные понятия (концепты), такие как стол, стулья, не могут быть

классифицированы по простому набору свойств, а должны быть описаны с помощью более сложной структуры (прецедентов). Эта работа впоследствии упоминалась как философская основа прецедентного подхода.

Группа ученых, которую возглавлял Шенк в Йельском университете, разработала первые приложения, основанные на этой модели. Колоднер разработал первую прецедентную систему под названием CYRUS [105]. Эта система содержала знания в виде прецедентов о путешествиях и встречах бывшего госсекретаря США Сайруса Венса. CYRUS была реализацией динамической модели памяти Шенка. Эта модель в дальнейшем послужила основой для ряда других систем: MEDIATOR [178], CHEF [147], PERSUADER [179], CASEY [153] и JULIA [148].

Альтернативный подход был описан в работах Портера в университете Техаса, который заключался в эвристической классификации и машинном обучении, в результате чего была разработана система PROTOS [105]. Данная система объединила общие знания предметной области и конкретные знания о проблеме в единую модель прецедента. Эти наработки были в дальнейшем использованы в системе GREBE, которая работала в правовой сфере [121], где понятие прецедента является базовым. Группа Edwina Rissland в университете штата Массачусетс в Амхерсте разработала систему NYPO [105]. В NYPO случаи, представляющие юридические прецеденты, используются для интерпретации ситуации и подготовки аргументов как защиты, так и для обвинения. Позднее эта система была объединена с системой, основанной на правилах, результатом такого объединения стала система CABARET [173].

Прецедентный подход развивался не только учеными из США, упомянутыми ранее, но и европейскими [106,107,111,151,161,172, 176,191], израильскими [169,170], индийскими [181], японскими [152] учеными. В России также проводятся исследования в этой области, однако гораздо менее интенсивно. Перечислим основные работы в этой области. Карпов Л. Е., Юдин В. Н. [31,32] предложили метод принятия решений, основанный на интеграции методов добычи данных и вывода по прецедентам, где методы добычи данных

используются для автоматического отбора из большой базы прецедентов и адаптации решения. Также они предложили метод выявления наиболее удачных прецедентов на основе предложенной локальной контекстно-зависимой метрики, базирующейся на предварительном разбиении базы прецедентов на классы эквивалентности и модифицированном методе распознавания (оценки) объектов в нефиксированном пространстве признаков [31,32,104].

Васильев В. И. и Белков Н. В. исследовали системы поддержки принятия решений (СППР) по обеспечению безопасности персональных данных. Формирование базы знаний происходит на основе правил и прецедентов с использованием онтологического анализа предметной области [12].

В [10,11] рассматривались вопросы, связанные с моделированием временных зависимостей в интеллектуальных системах поддержки принятия решений (СППР) на основе прецедентов. Описаны особенности применения систем на основе прецедентов (CBR-систем) для поддержки принятия решений в открытых и динамических предметных областях при наличии достаточно жестких временных ограничений. В [19,20] исследованы преимущества применения систем на основе темпоральных прецедентов для поддержки принятия решений в открытых и динамических предметных областях в условиях наличия неопределенности в исходных данных и достаточно жестких временных ограничениях.

За последнее десятилетие прецедентный подход был применен в различных областях. Области применения прецедентного подхода для решения практических задач можно сгруппировать в несколько основных классов: строительная отрасль, медицинская диагностика, SCM/CRM системы, стратегическое планирование, и другие.

Известны следующие основные исследования прецедентного подхода в строительной области: Cirovic и Sekic [131] применяют CBR для поддержки разработчиков в области строительных проектов на этапе предварительного проектирования. Разработчики в соответствии с критериями текущей задачи

извлекают из интегрированной базы данных проектов (IPDB) подходящие прецеденты, применяют проектные решения к текущей задаче, и затем сохраняют в базу знаний (СКБ) проектное решение для качественного проектирования в будущем. Juan, Shin, and Perng [149] интегрировали CBR с генетическим алгоритмом (ГА) для проектирования в области жилищного строительства. Их система посвящена изготовлению под заказ дизайнерского проекта внутренней отделки жилых помещений.

Ученые, которые занимались исследованиями в области медицинской диагностики: Guiu, Ribe, Mansilla, Fabrega [145] и Galobardes, Llorca, Salamo, Martí [140] использовали CBR для диагностики рака молочной железы. Chang C.L. [125] применял CBR для выявления детей с задержкой развития. Диагностирование этого заболевания на ранней стадии позволяет пациенту иметь шанс на выздоровление. Задержка развития у детей может быть диагностирована по определенному набору симптомов, поэтому CBR хорошо подходит для решения подобных задач.

Исследования в области SCM/CRM систем также показывают эффективность прецедентного подхода. Исследователи Choy and Lee [130], используют CBR для выбора поставщиков, оказывающих аутсорсинговые услуги. Ученые оценивали поставщиков в соответствии с их техническими возможностями, оценкой качества оказанных услуг и профилем организации. Choy, Fan, and Lo [129] расширили исследование, приведенное выше, и разработали интеллектуальную систему клиент-SRM. Исследователи Cheng and Cheng [127] разработали систему для предсказания качества пряжи из хлопка, на основании свойств волокон с использованием CBR. В прошлом, эти проблемы обычно решались с помощью математической модели, например, регрессионного анализа, но для этого необходимо было сформулировать слишком много гипотез, что затрудняло применение математической модели на реальных объектах. Прецедентный подход решил эту проблему.

Известны следующие исследования прецедентного подхода в области стратегического планирования: Changchien and Lin [126] создали

автоматизацию разработки плана по маркетингу для персонала телекоммуникационной компании. С помощью CBR было проанализировано состояние оказываемых услуг по отраслям, например, проведен внешний экологический аудит, аудит промышленности и т.д., и выявлены ключевые маркетинговые стратегии, относительно продуктов, цены, места и продвижения. Исследователи Lim, Ahn, and Lee [154] применили CBR для разработки стратегий для конкурентного управления. После анализа, проведенного акционерами, RDAP (реакция, защита, адаптация, и действие) стратегии извлекаются, перерабатываются и реализуются с помощью механизма CBR.

Кроме перечисленных выше областей прецедентный подход с успехом применяли в других сферах: исследователи Belecheanu et al. [118] ссылаются на использование прецедентов для снижения неопределенности информации, например, в области промышленных потребностей. Ученый Gwen [146] предложил применение CBR для управления знаниями, Choobineh and Lo [128], предложили использовать CBR для улучшения баз данных по концептуальному проектированию.

Хотя прошло много времени с момента появления прецедентного подхода, интерес исследователей к нему не ослабевает. Международная конференции по прецедентному подходу (ICCBR) проводится ежегодно с 2009 года. ICCBR 2016 охватила широкий спектр тем по прецедентному подходу от основ CBR, новых методов извлечения и повторного использования, до различных подходов к применению CBR в конкретных областях знаний. Например, в работе [108] предложено CBR–решение по планированию маршрута для групп людей в музее истории информатики, расположенном на факультете информатики университета Комплутенсе в Мадриде. В работе [117] представлены два подхода (только на основе правил и, одновременно, на основе правил и прецедентов) для поиска подходящего архитектурного дизайна. Ряд работ был посвящен развитию электронной коммерции [135,192] и электронного правительства [134,155,187] на основе прецедентной системы

поддержки принятия решений. В следующем подразделе рассмотрим подробно прецедентную модель.

1.3.2. Прецедентная модель

Модель рассуждений на основе прецедентов (Case-Based Reasoning, CBR) используется для представления знаний в социально-экономических системах. Прецедент – это описание проблемы или ситуации в совокупности с подробным указанием действий, предпринимаемых в данной ситуации или для решения данной проблемы [104]. Значит, модель на основе прецедентов – это модель принятия решений, в котором используются знания о предыдущих прецедентах. При рассмотрении новой проблемы отыскивается похожий прецедент в качестве аналога. Вместо того чтобы искать решение каждый раз сначала, можно попытаться использовать решение, принятое в сходной ситуации, возможно, адаптировав его к изменившейся ситуации текущего случая. После того как текущий случай будет обработан, он вносится в базу прецедентов вместе со своим решением для его возможного последующего использования [120].

Метод рассуждений на основе прецедентов включает четыре этапа, формирующие так называемый CBR-цикл (рис. 1.6), или цикл 4R (Retrieve, Reuse, Revise, Retain) [47,53]:

-*извлечение* (retrieve) наиболее адекватного (подобного) прецедента для сложившейся ситуации из базы знаний;

-*повторное использование* (reuse) извлеченного прецедента для попытки решения текущей проблемы;

-*пересмотр и адаптация* (revise) в случае необходимости полученного решения в соответствии с текущей проблемой;

-*сохранение* (retain) вновь принятого решения как части нового прецедента.

Проанализировав CBR подход, сформулируем типовой алгоритм работы СППР по решению проблемы пользователя:

- идентификация проблемы у пользователя;
- поиск в базе знаний подходящего прецедента;

- выполнение набора операций на тестовой базе;
- оценка результата (помогло/не помогло);
- доработка набора действий в прецеденте;
- сохранение нового прецедента в базу.

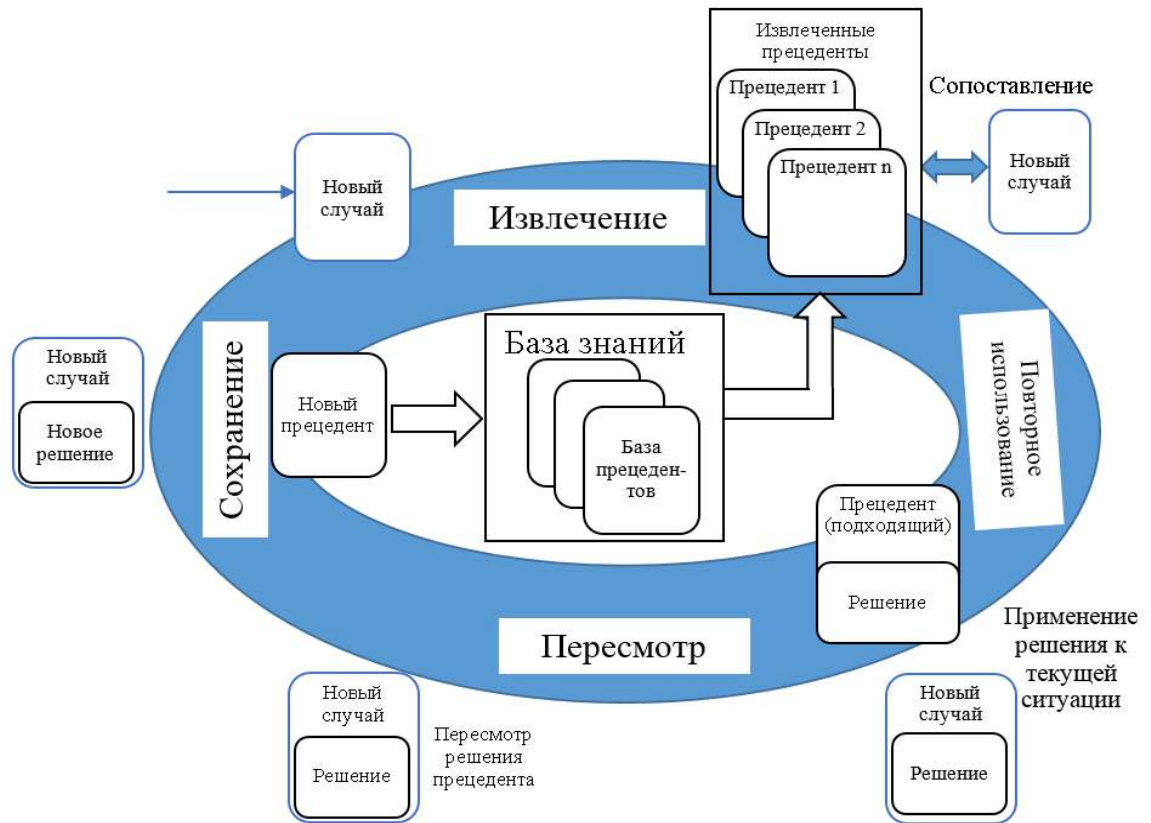


Рисунок 1.6. CBR-цикл

Можно сформулировать следующие основные достоинства CBR подхода:

- метод рассуждений на основе прецедентов не требует явной модели предметной области, поэтому извлечение знаний превращается в простую задачу сбора прецедентов;
- можно использовать технологию баз данных для хранения больших объемов прецедентов;
- CBR-системы самообучаемы, новые прецеденты можно получать, применяя методы машинного обучения.

В то же время есть некоторые недостатки CBR-систем:

- при описании случаев, обычно не принимаются во внимание более глубокие знания о предметной области;

- большое количество прецедентов в базе приводит к снижению производительности системы;
- трудно определить, хорошие критерии для индексирования и сравнения прецедентов.

Несмотря на указанные недостатки, прецедентный подход является признанным и заслуживающим доверия для построения баз знаний. Этот подход органично подходит для поддержки принятия решений в области ИТ-консультирования. Однако нам представляется, что в данной сфере целесообразно интегрировать метод прецедентов с моделью представления знаний в виде онтологии предметной области, в которой решается проблема пользователя. В [176] удалось повысить релевантность поиска документов в области ИТ-технологий на основе индексирования документов концептами онтологии в отличие от обычного индексирования ключевыми словами, хотя в этой работе и не использовался прецедентный подход. Нам представляется что интеграция прецедентной модели представления знаний с онтологической даст новые перспективы в области управления знаниями при решении задач ИТ-консультирования.

1.3.3. Онтологическая модель

В области ИТ-консультирования ключевым компонентом являются знания о предметной области. Мы предлагаем интеграцию прецедентной модели представления знаний с онтологической моделью для улучшения качества взаимодействия пользователя и консультанта. Такая интеграция позволит разговаривать на едином профессиональном языке (использовать одинаковую терминологию) пользователю, являющемуся специалистом в предметной области, и консультанту, являющемуся экспертом в ИТ-проблемах, в результате чего улучшится качество взаимодействия а, следовательно, скорость и качество решения проблем. Рассмотрим основные понятия онтологического инжиниринга.

В изучении понятия онтология существует множество вопросов. До сих пор нет четкого определения для этого понятия. Само понятие онтология

происходит от др.-греч. «онтоc» – сущее, «логос» – учение, понятие, т.е. это раздел философии, изучающий бытие [21,168].

Для специалистов в области информационных технологий, классическим определением онтологии является определение: «онтология – это эксплицитная спецификация концептуализации» данное Грубером в 1993 г. [141].

Грубер в своей работе [142] рассматривал взаимодействие интеллектуальных систем между собой и с человеком. Под интеллектуальной системой понимаются программы, которые моделируют некоторые аспекты интеллектуальной деятельности человека. Можно сказать, любая программа автоматизирует некоторые аспекты деятельности человека. В этом и состоит ценность внедрения компьютерных технологий в деятельность организации – помогать решать рутинные задачи и облегчать работу человеку, но в этом смысле, знания, которые заложены в алгоритме программы, являются статичными. Интеллектуальная же система в этом смысле универсальна, в ней знание о том, что надо делать в процессе исполнения программы, не заложено в программу раз и навсегда, а может меняться. Если так, то эти знания необходимо передавать программе как данные, т.е. возникает необходимость их описания.

Идея Грубера состояла в том, чтобы позволить интеллектуальным системам обмениваться между собой заложенными в них знаниями о предметных областях задач. Если внутри интеллектуальной системы знания о предметной области могут быть закодированы как угодно, то для обмена этими знаниями с другой интеллектуальной системой необходимо предоставить описание этих знаний. Это описание должно быть в достаточной степени формальным, чтобы быть понятным другой системе, а также должен быть известен язык этого описания. Кроме того, описание должно быть понятно также и человеку.

Таким образом, онтология по Груберу представляет собой описание декларативных знаний, сделанное в виде классов с отношением иерархии между ними. К этому описанию, предназначенному для чтения человеком,

присоединено описание в канонической форме, которое предназначено для чтения машинами.

Также известен ряд расширенных определений Грубера, среди которых можно выделить такие:

- онтология – это эксплицитная спецификация концептуализации, где в качестве концептуализации выступает описание множества объектов предметной области и связей между ними [141];
- онтология – это знания, формально представленные на базе концептуализации. Формально онтология состоит из терминов, организованных в таксономию, их определений и атрибутов, а также связанных с ними аксиом и правил вывода [142];
- онтология – база знаний, описывающая факты, которые предполагаются всегда истинными в рамках определенного общества на основе общепринятого смысла используемого словаря [143].

Существуют сложности с определением самого понятия «онтология», также возникают сложности и с описанием модели формальным языком. Классическим считается, такое формальное описание онтологии:

$O = \langle X, R, F \rangle$, где X – конечное множество понятий предметной области; R – конечное множество отношений между понятиями; F – конечное множество функций интерпретации [66].

В проектировании онтологий условно можно выделить два направления, до некоторого времени развивавшихся отдельно. Первое связано с представлением онтологии как формальной системы, основанной на математически точных аксиомах (т.е. это ресурсы онтологического типа, созданные в различных областях математики). Второе направление развивалось в рамках компьютерной лингвистики и когнитивной науки. Здесь онтология понималась как система абстрактных понятий, существующих только в сознании человека, которая может быть выражена на естественном языке (или

средствами какой-то другой системы символов). При этом обычно не делается предположений о точности или непротиворечивости такой системы.

Таким образом, существует два альтернативных подхода к созданию и исследованию онтологий. Первый (формальный) – основан на логике (предикатов первого порядка, дескриптивной, модальной и т.п.). Второй (лингвистический) – основан на изучении естественного языка (в частности, семантики) и построении онтологий на больших текстовых массивах, так называемых корпусах [142].

Онтологический инжиниринг имеет уже достаточную историю исследований и разработок [5,16,18,21,30,35,59,83,94,144,180], которая уже привела к построению широкого спектра различных онтологических моделей, требующих определенной классификации. В рамках классификации [18] онтологии различаются: по степени формальности; наполнению, содержанию; цели создания. В рамках классификации онтологий по цели создания часто выделяют четыре уровня: онтологии представления, онтологии верхнего уровня, онтологии предметных областей и прикладные онтологии. При этом целью создания онтологий представления является описание области представления знаний и разработка языков для спецификации более низких уровней онтологий. Примером онтологии этого уровня является описание понятий языка OWL средствами RDF/RDFS.

В настоящее время существует несколько серьезных проектов, направленных на создание онтологий верхнего уровня. Например, Cyc, DOLCE, SUMO, Sowa's Ontology и других, но в целом попытки создать онтологию верхнего уровня для всех предметных областей, которая бы признавалась всеми специалистами, пока не привела к ожидаемым результатам.

Назначение онтологий предметных областей схоже с назначением онтологий верхнего уровня, но область их охвата ограничена определенной предметной областью (консультирование, медицина и т.п.), а прикладные онтологии создаются для описания концептуальных моделей конкретных задач и/или приложений.

Таким образом, разработка онтологической модели предметной области, а также ее интеграция с прецедентной моделью ИТ-консультирования, представляется полезной и перспективной для улучшения взаимодействия консультанта с пользователем.

1.4. Выводы

В данной главе мы рассмотрели проблему управления деятельностью ИТ-подразделения организаций как в общем контексте управления социальными и экономическими системами, так и в более узком контексте проблемы управления знаниями. Для эффективного управления знаниями в области ИТ-консультирования оказывается актуальным создание информационной технологии поддержки принятия решений, базирующейся на современных интеллектуальных методах проектирования баз знаний, содержащих как знания предметной области, так и ретроспективные знания о прошлом опыте принятия решений в подобных ситуациях. Однако существующие информационные системы поддержки деятельности ИТ-консультантов (системы HelpDesk) не обладают необходимым модельным и программным инструментарием.

Таким образом, целью диссертационной работы является разработка новой информационной технологии поддержки принятия решений в области ИТ-консультирования на основе эффективного управления знаниями в этой сфере.

Для достижения указанной цели в последующих главах диссертации предлагается разработка соответствующего модельного, методического и программного инструментария для интеллектуальной поддержки деятельности консультантов. В главе 2 мы разрабатываем гибридную модель представления знаний на основе интеграции онтологии предметной области с прецедентами принятия решений. В главе 3 предлагается эффективный метод извлечения прецедентов на основе генерирования нечетких правил классификации. В главе 4 описана созданная интеллектуальная система, базирующаяся на основе предлагаемого модельного и алгоритмического инструментария, а также новая информационная технология поддержки принятия решений в области ИТ-консультирования пользователей, построенная на основе использования интеллектуальной системы.

ГЛАВА 2. ГИБРИДНАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ ИТ-КОНСУЛЬТИРОВАНИЯ

Данная глава посвящена построению онтологии предметной области, для которой решаются задачи ИТ-консультирования, проектированию структуры прецедента, а также интеграции прецедентов принятия решений с онтологией предметной области. Подобная интеграция призвана помочь улучшению взаимодействия консультанта-аналитика с пользователем, являющимся специалистом предметной области, в которой возникает проблема при использовании ИТ. Полученная в результате интеграции гибридная модель представления знаний является основой для организации эффективного способа извлечения семантически близких прецедентов. Полученные в этой главе результаты отражены в публикациях [3,43,45,46,50].

2.1. Построение онтологии в области ИТ-консультирования

2.1.1. Формальная модель онтологии предметной области

Онтологии – формальные явные описания терминов предметной области и отношений между ними [142]:

$$O = \langle C, R, S, G, T, E \rangle \quad (2.1)$$

где $C = \{c_i \mid i = \overline{1, n}\}$ – конечное непустое множество классов (концептов), описывающих основные понятия предметной области; $R = \{r_i \mid i = \overline{1, m}\}$ – конечное множество бинарных отношений, заданных на классах, $R \subseteq C \times C$, являющееся объединением двух подмножеств $R = \{R_{ISA}\} \cup \{R_{ASS}\}$, где R_{ISA} – антисимметричное, транзитивное, нерефлексивное отношение иерархии «класс-подкласс», задающее частичный порядок на множестве классов; R_{ASS} – ассоциативное отношение, которое будет использовано для установления связи между прецедентами и концептами онтологии; $S = \{s_i \mid i = \overline{1, k}\}$ – конечное множество слотов (свойств класса); $G = \{gs_i \mid i = \overline{1, l}\}$ – конечное множество

фасетов (свойств слота); $E = \{e_i \mid i = \overline{1, u}\}$ – конечное множество экземпляров классов; T – конечное непустое множество, определяющее контролируемый словарь терминов предметной области, построенное на множестве базовых терминов $B = \{b_i \mid i = \overline{1, n}\}$, составляющих множество имен классов онтологии:

$$T = \bigcup_{i=1}^n T_i, T_i = \{b_i\} \cup Eq(b_i), \bigcap_{i=1}^n T_i = \emptyset, Eq(b_i) \text{ – множество синонимов}$$

базового термина.

Структура класса определяется следующим образом:

$$c = \langle Name, (isa\ c_{parent}), (s_1, \dots, s_{n(c)}) \rangle \quad (2.2)$$

где $c, c_{parent} \in C$ – классы онтологии, связанные отношением иерархии R_{ISA} , $s_i \in S$ – слоты фрейма, $Name \in B$ – имя класса, являющееся базовым термином словаря T . Таксономия классов образуются посредством указания в подклассе связи «is-a» и имени класса-родителя c_{parent} . Назовем классы, не имеющие потомков, терминальными концептами онтологии (терминалами). Терминалы будут играть роль ключевых слов при индексировании прецедентов.

Структура слота определяется следующим образом:

$$s_C = \langle Name_{S,C}, (gs_1, gs_2, \dots, gs_{k(S,C)}) \rangle \quad (2.3)$$

где $s_C \in S$ – слот класса C , $gs_i \in G$ – фасеты (свойства слота), $Name_{S,C}$ – имя слота.

Онтология включает наиболее востребованные у пользователей области - бухгалтерский учет, описываемый концептом *Accounting*, расчеты с персоналом, описываемый концептом *Payroll* и договорной блок, описываемый концептом *ContractUnit*. Иерархия концептов самого верхнего уровня представлена на рис. 2.1. Таким образом, построенная онтология предметной области представляет собой объединение трех таксономий, внутри которых концепты связаны иерархическими отношениями. Допускается, что потомки

родителя имеют неодинаковое влияние на родительский концепт. Для этого каждому концепту, имеющему родителя, добавляется слот – вес концепта, участвующий в определении семантической связи прецедентов с терминальными концептами онтологии.

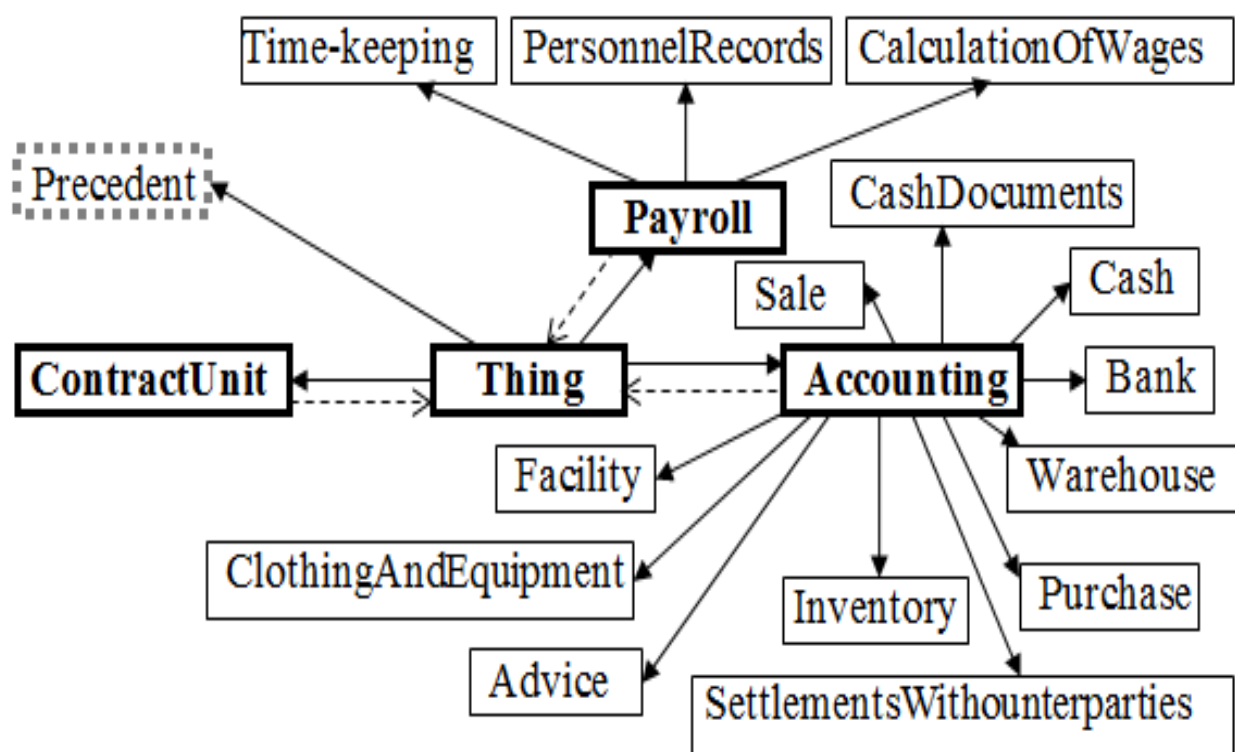


Рисунок 2.1. Иерархическая структура концептов верхнего уровня

Основными концептами верхнего уровня онтологии являются *Accounting*, *Payroll* и *ContractUnit*.

Концепт *Accounting* описывает основные подразделы бухгалтерского учета. Бухгалтерский учет представляет собой упорядоченную систему сбора, регистрации и обобщения информации в денежном выражении об имуществе, обязательствах организаций и их движении путем сплошного, непрерывного и документального учета всех хозяйственных операций. *Accounting* образует таксономию, которая формируется двенадцатью подчиненными концептами (рис.2.2). Концепты *Bank* и *Cash* отражают ведение операций с денежными средствами, за безналичные и наличные расчеты соответственно. Концепт *Sale* отражает оформление операций по продаже товаров и оказанию услуг клиентам, этот концепт является одним из основных для ведения деятельности предприятия. Концепт *Purchase* предназначен для учета ведения операций по

покупкам товаров и услуг у поставщиков и имеет подчиненные концепты. Концепт *Warehouse* отражает учет движения материалов на складе и т.д. Данные и подчиненные концепты помогают выразить смысл вопросов, возникающих у пользователей. Например, вопрос пользователя «В поступлении товара ставка в номенклатуре проставляется без НДС, почему?» целесообразно связать с концептом *Purchase*.

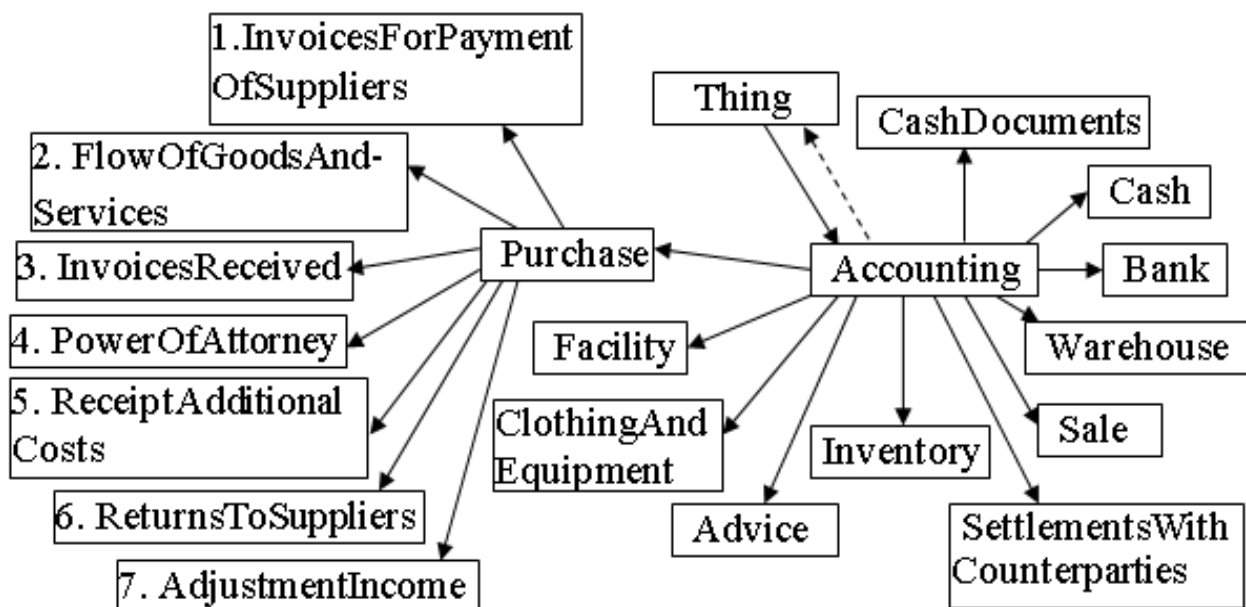


Рисунок 2.2. Иерархическая структура концепта *Accounting*
(бухгалтерский учет)

Концепт *Payroll* описывает основные подразделы таксономии «Расчеты с персоналом». В этой таксономии решаются задачи автоматизации деятельности как менеджеров, принимающих решения по зарплате персонала, так и бухгалтеров-расчетчиков зарплаты; обеспечивается также ведение взаиморасчетов с работниками предприятия, осуществляется учет затрат на оплату труда в составе себестоимости продукции и услуг, начиная с ввода документов о фактической выработке, об оплате больничных листов и отпусков, вплоть до формирования документов на выплату зарплаты и отчетности в государственные надзорные органы. У пользователей могут возникать различные вопросы, связанные с данными концептами. Например, у сотрудника отдела кадров могут возникнуть следующие вопросы: «При создании сотрудника возникает ошибка, что физическое лицо уже существует,

что делать?», «Как внести больничный лист?». Эти вопросы можно отнести к концепту *PersonnelRecords*. Пользователи, которые могут задавать вопросы, связанные с таксономией *Payroll*, – это бухгалтеры по заработной плате, кадровики, табельщики, т.е. пользователи отдела расчетов с персоналом. Фрагмент онтологии концепта *Payroll* представлен на рис. 2.3.

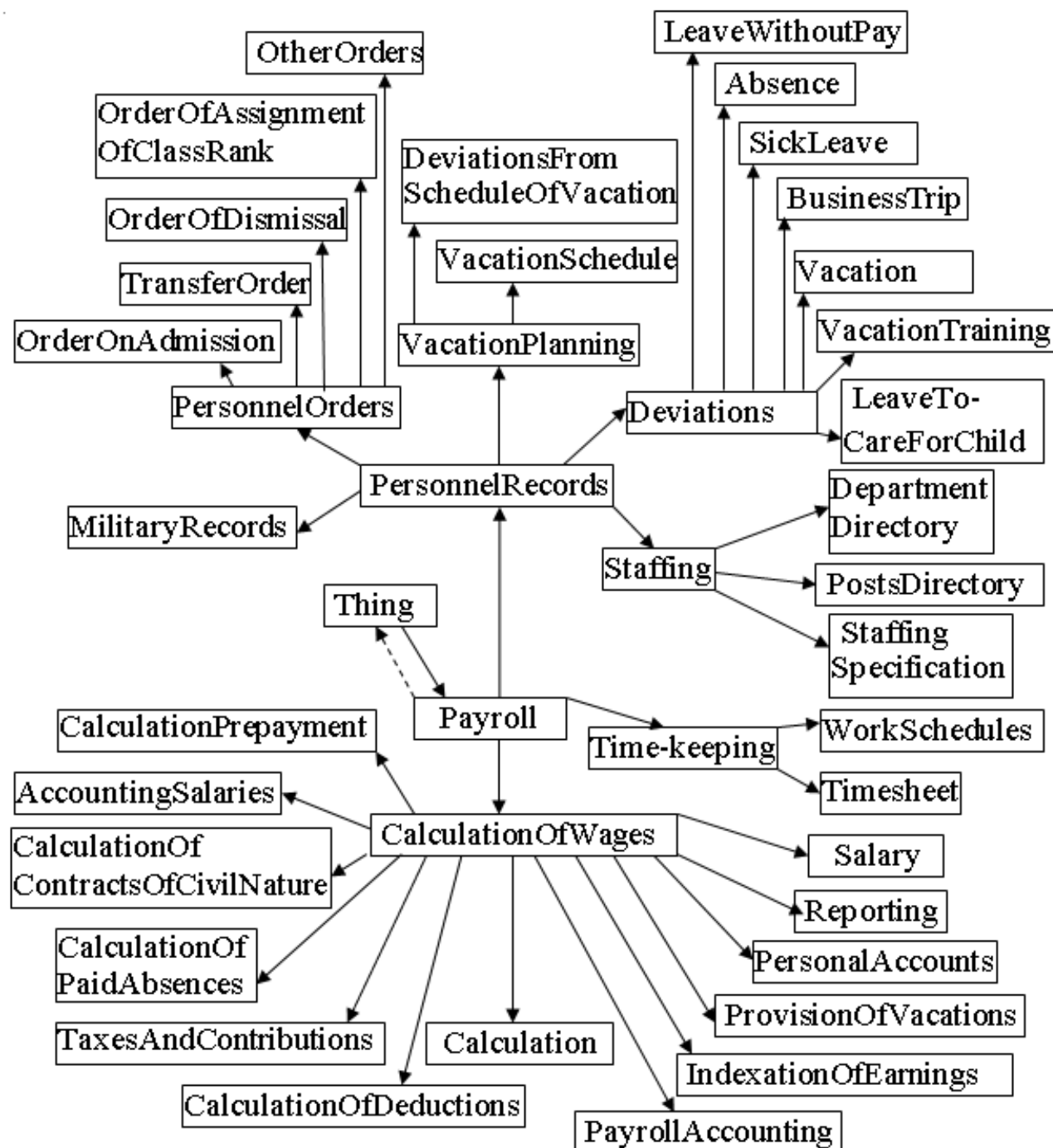


Рисунок 2.3. Иерархическая структура концепта *Payroll*
(расчеты с персоналом)

Концепт *ContractUnit* (фрагмент онтологического графа представлен на рис. 2.4) описывает основные подразделы предметной области «Договорной блок». Договорной блок предназначен для автоматизации работы пользователей в сфере оформления и ведения договоров контрагентов. Договорной блок состоит из концептов *Contractors* – описание основных реквизитов контрагентов, *ContractCounterparties* – оформление договоров контрагентов, *AdditionalAgreements* – учет дополнительных соглашений для введения в действие договоров контрагентов и автоматического формирования пакета документов для оформления оказания услуги контрагенту, *AmendmentOfContract* – изменение реквизитов договоров, *ChangeConsoleRooms* – изменение пультовых номеров, *Rates* – настройка тарифов контрагентов, *Reports* – формирование отчетов по оказанию услуг контрагентам. Например, у договорника могут возникнуть следующие вопросы: «Как ввести договор в действие?», «Почему нет начислений по договору?». Эти вопросы относятся к концепту *ContractCounterparties*. Пользователи, которые могут задавать вопросы по концепту *ContractUnit*, – это бухгалтеры по договорам или договорники.

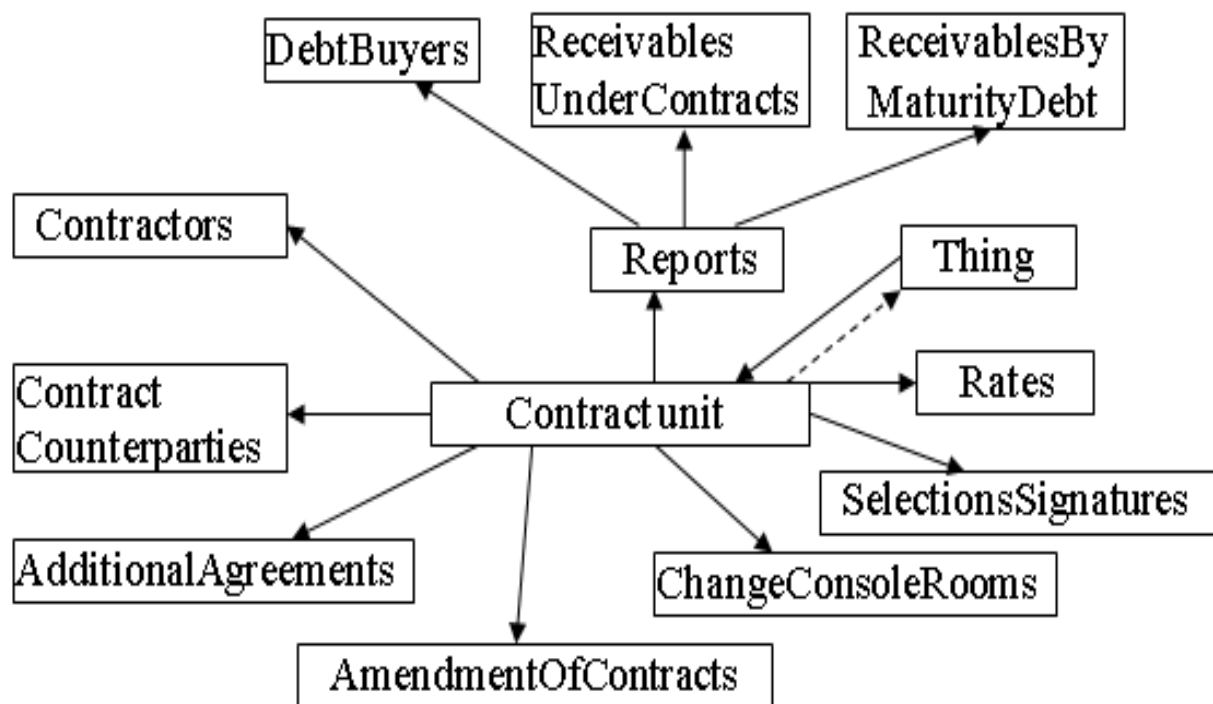


Рисунок 2.4. Иерархическая структура концепта *ContractUnit*
(договорной блок)

Созданная в Protégé OWL иерархия концептов содержит 71 концепт таксономии *Payroll*, 82 концепта таксономии *Accounting* и 11 концептов таксономии *ContractUnit*.

Таким образом, вышеописанная онтология предметной области представляет собой объединение трех таксономий концептов, связанных иерархическими отношениями. В принципе, если потомки некоторого родителя имеют неодинаковое влияние на родительский концепт, то возможно введение в таксономию весовых коэффициентов. Для этого каждому концепту, имеющему родителя, добавляется слот – вес концепта.

2.1.2. Реализация онтологии в Protégé OWL

Онтология была создана в редакторе Protégé OWL – это свободный, открытый редактор онтологий для построения баз знаний. Protégé OWL – программный продукт, разработанный Стэнфордским университетом. Онтологии, построенные в этом редакторе, экспортируются во множество форматов: RDF (RDF Schema), OWL и XML Schema. Этот редактор имеет открытую и легко расширяемую архитектуру.

Любая онтология содержит заголовок и тело. На рис. 2.5 отображен заголовок онтологии, добавлена аннотация, можно добавить комментарии, и настроить импорт онтологий. Тело онтологии содержит описание классов, свойств и индивидов.

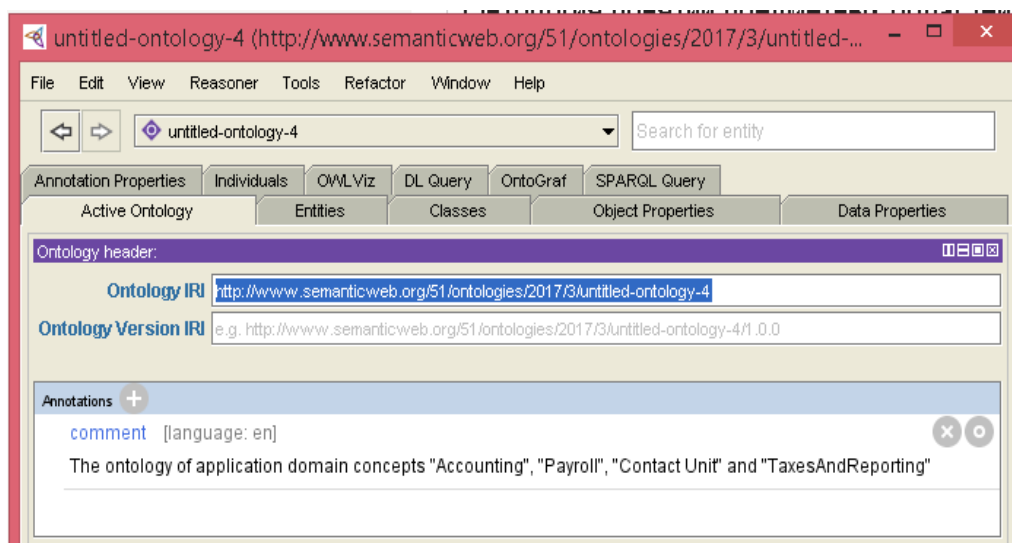


Рисунок 2.5. Заголовок онтологии, созданной базы знаний

В язык OWL введен класс «owl:Class». Классы организуются в иерархию с помощью свойства «rdfs:subClassOf». Важное место занимают два взаимодополняющих класса: «owl:Thing» – надкласс любого класса OWL; «owl:Nothing» – подкласс любого класса OWL. Экземпляр любого класса OWL входит экстенционал класса «owl:Thing». Экстенционал класса «owl:Nothing» является пустым множеством. Описания класса являются строительными блоками для определения классов посредством аксиом. Для определения классов и формирования различных аксиом существует ряд конструкций: «rdf:ID» – определение именованного класса; «rdfs:subClassOf» – экстенционал одного класса полностью входит в экстенционал другого; «owl:equivalentClass» – экстенционалы двух классов совпадают; «owl:disjointWith» – экстенционалы двух классов не пересекаются.

Классы являются основным компонентом OWL-онтологии. Классы бывают именованные, непересекающиеся и иерархические. При создании классов в редакторе Protégé первоначально уже содержится класс «Thing» – это класс, представляющий набор, содержащий все объекты предметной области.

Концепт *Accounting* был реализован в редакторе онтологий Protégé (рис.2.6). Одним из основных подразделов в предметной области бухгалтерского учета является концепт *Purchase*. *Purchase* образует таксономию, которая формируется семью подчиненными концептами. Концепт *InvoicesForPaymentOfSuppliers* отражает учет счетов на оплату, выданных поставщикам. Концепт *FlowOfGoodsAndServices* отражает операции по поступлению товаров и услуг. Концепт *InvoicesReceived* отражает счет-фактуры полученные от контрагентов. Концепт *PowerOfAttorney* отражает учет доверенностей, выданных сотрудникам организации. Концепт *ReceiptAdditionalCosts* отражает поступление дополнительных расходов. Концепт *ReturnsToSuppliers* отражает возвраты денежных средств поставщикам. Концепт *AdjustmentIncome* отражает корректировку поступления, когда операции по контрагенту были некорректно отражены и возникла потребность внести изменения в документы.

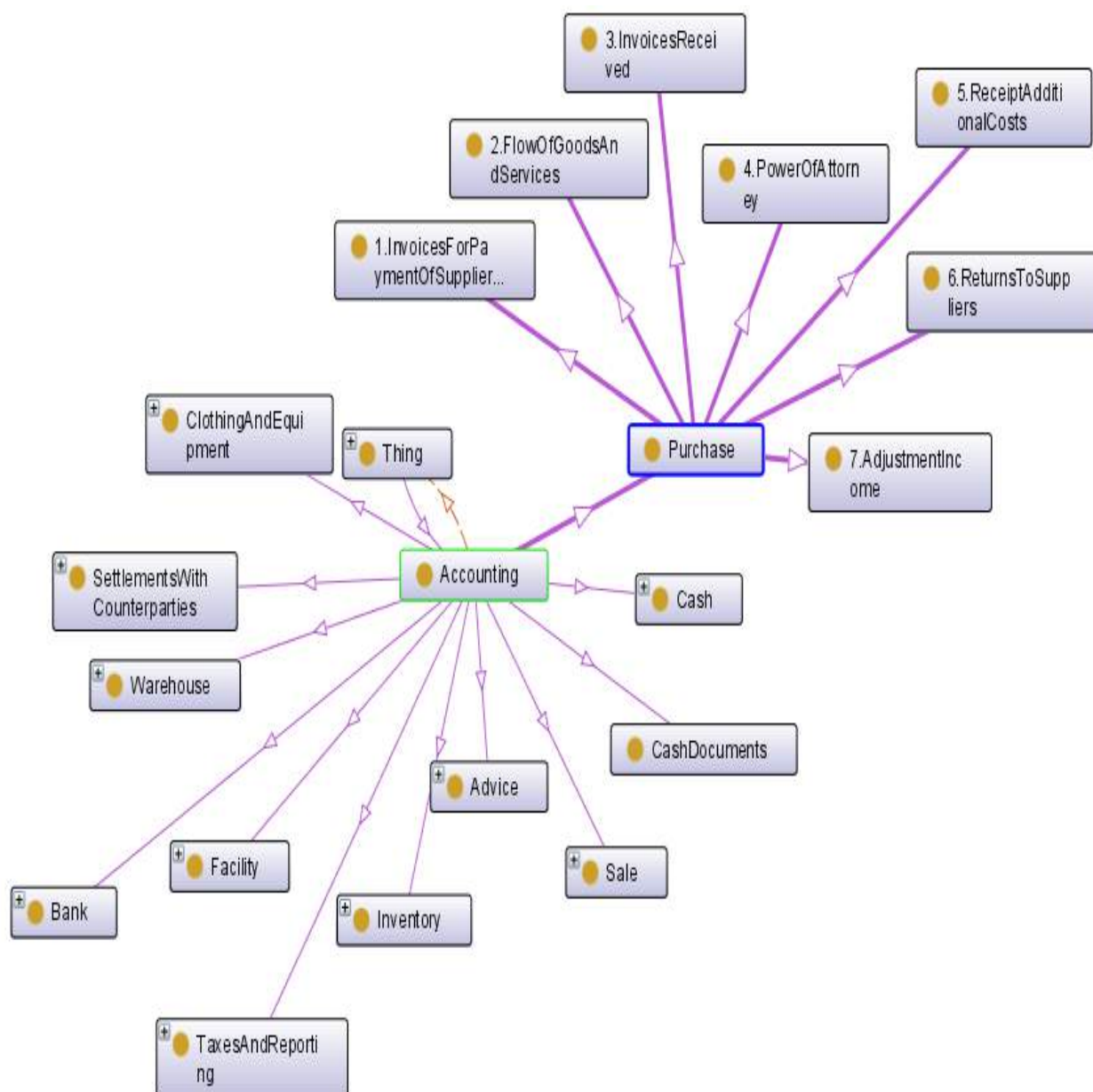
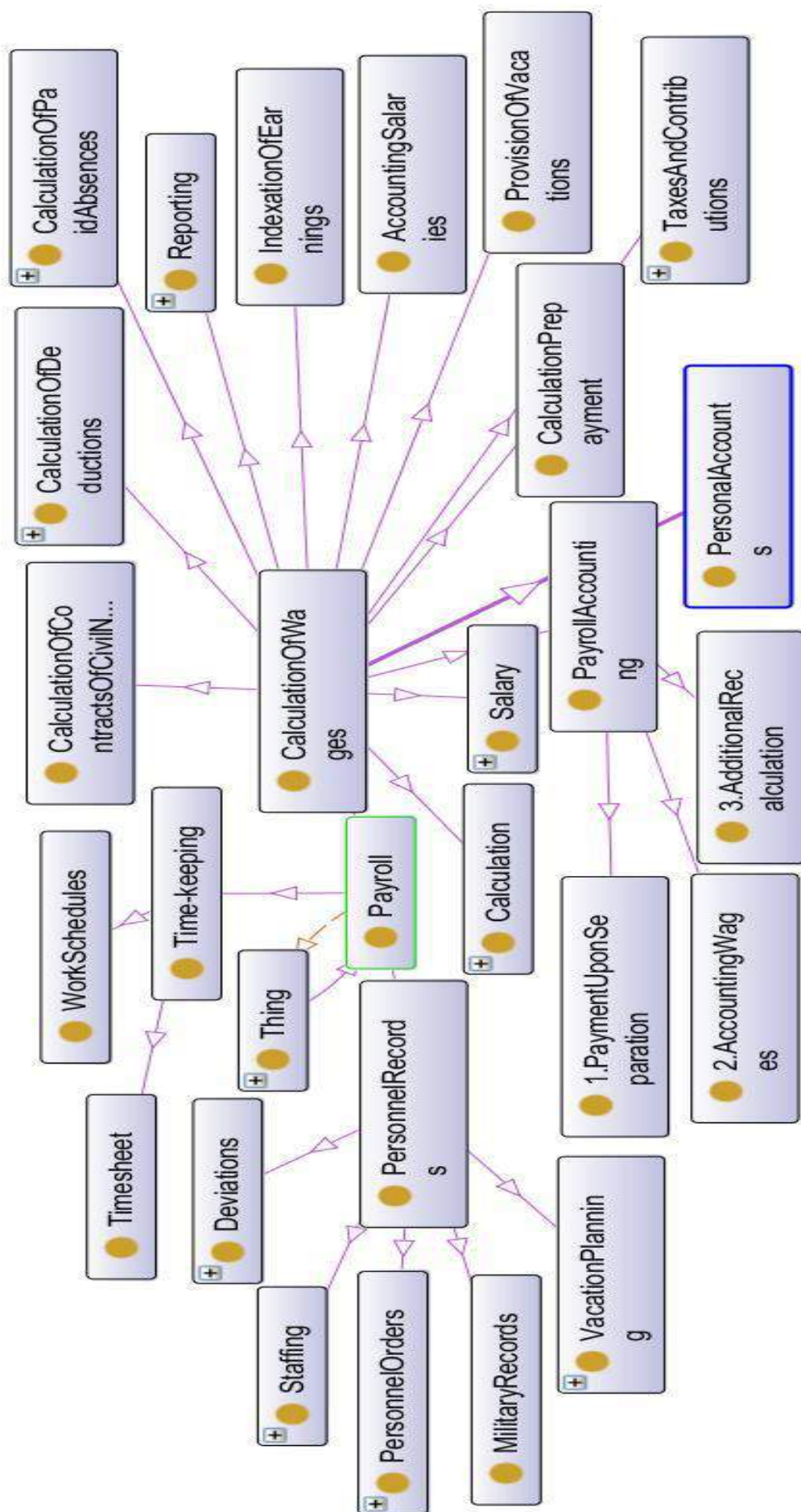


Рисунок 2.6. Онтологический граф концепта *Accounting*

Концепт *Payroll* был реализован в редакторе онтологий Protégé. Одним из основных подразделов в предметной области расчеты с персоналом является концепт *PayrollAccounting*. *PayrollAccounting* образует таксономию, которая формируется тремя подчиненными концептами. Концепт *PaymentUponSeparation* отражает расчет по сотруднику при увольнении. Концепт *AccountingWages* отражает все операции по начислению заработной платы сотруднику организации. Концепт *Additional\recalculation* отражает операции по доначислению и перерасчет у сотрудникам. Фрагмент онтологии концепта *Payroll* представлен на рис. 2.7.

Рисунок 2.7. Онтологический граф концепта *Payroll*

Также в онтологии указаны непересекающиеся классы (любой объект не пересекающегося класса может быть экземпляром только одного класса). Например, класс «Order on admission» является непересекающимся классом к классам «Order of dismissal», «Transfer order», «Order of assignment of class rank» и «Other order». Для указания, что класс «Order on admission» не пересекается с другими классами, есть свойство «Disjoint with», расположенное в группе «Description».

Концепт *ContractUnit* был реализован в редакторе онтологий Protégé (рис. 2.8).

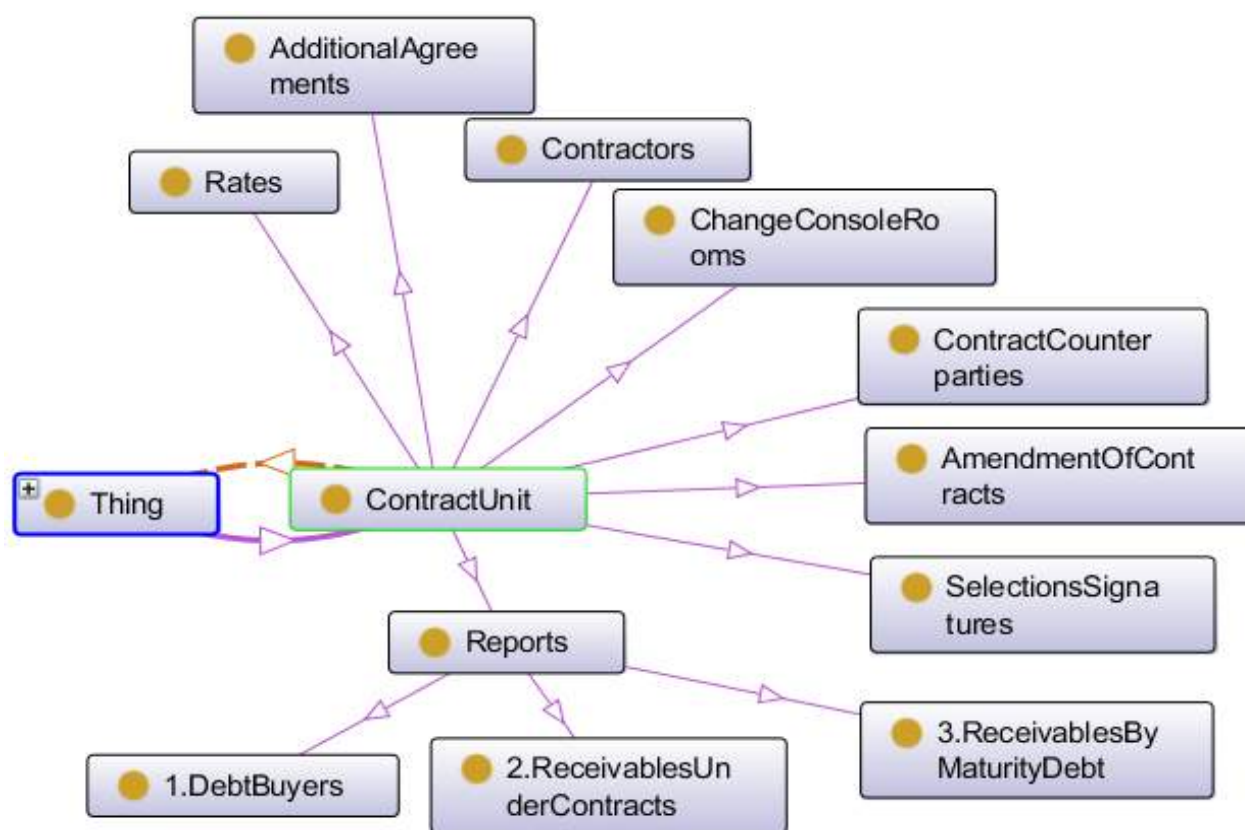


Рисунок 2.8. Онтологический граф концепта *ContractUnit*

Одним из подразделов в предметной области договорной блок является концепт *Reports*. *Reports* образует таксономию, которая формируется тремя подчиненными концептами. Концепт *DebtBuyers* отражает отчет по задолженности покупателей. Концепт *ReceivablesUnderContracts* отражает отчет по задолженности покупателей по договорам. Концепт *ReceivablesByMaturityDebt* отражает отчет по задолженности покупателей по срокам долга.

Иерархия концептов из Protégé 4.2. представлена на рис. 2.9 [50].

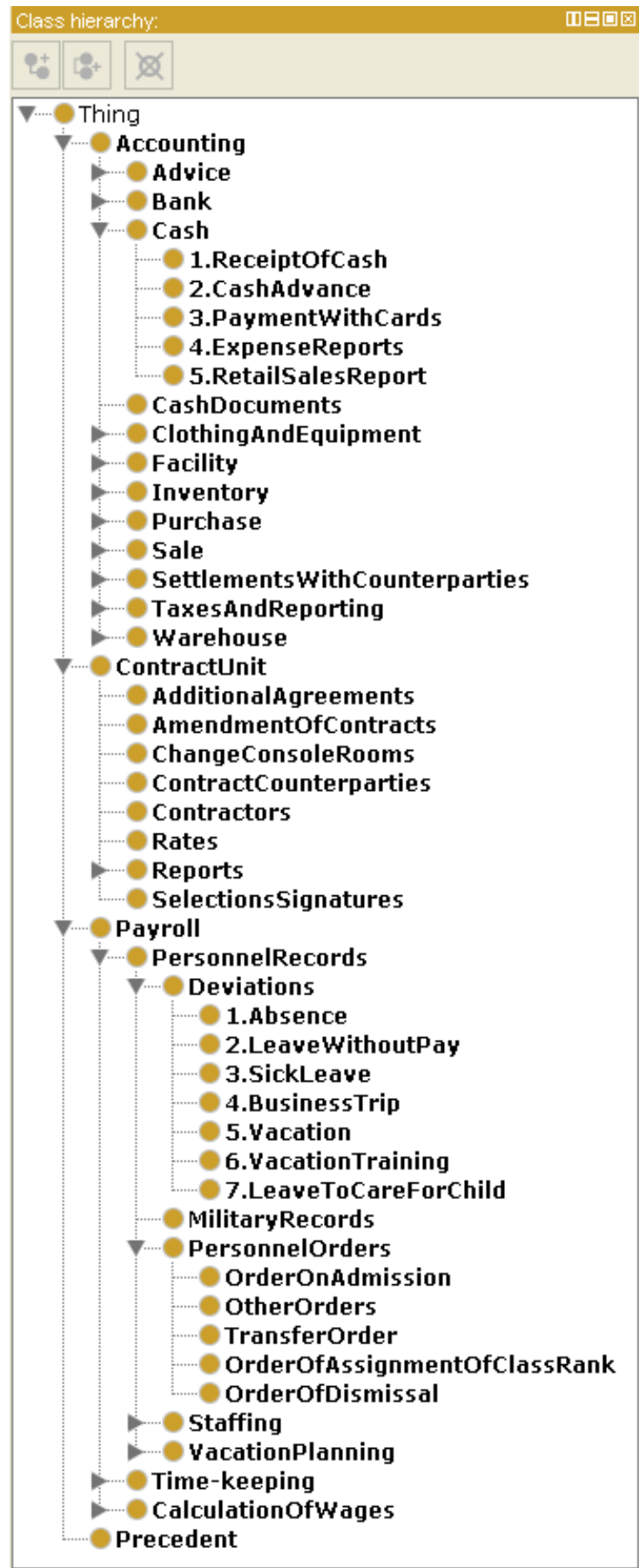


Рисунок 2.9. Иерархия концептов

Созданные концепты нашей предметной области также представлены в приложении А на рис. А.1.

Свойства в языке OWL представляют собой отношения. Существует два основных типа свойств: свойства объектов и свойства типов данных. Свойства объектов – это отношения между двумя индивидами. Свойства типов данных – это отношения между индивидами со значениями данных. Оба класса свойств являются подклассами класса «*rdf:Property*». Для определения новых свойств как экземпляров «*owl:ObjectProperty*» или «*owl:DatatypeProperty*» используются аксиомы свойств.

В языке OWL свойства имеют различные характеристики: функциональные свойства, обратные функциональные свойства, транзитивные свойства, симметричные свойства, ассиметричные свойства, рефлексивные свойства, иррефлексивные свойства. Задать характеристику свойства можно через панель характеристик, представленную в приложении А на рис. А.2.

Рис. 2.10 показывает, что свойство *Keyword1* свяжет индивида «*A_return_to_his_former_place_of_work_during_temporary*», наполняющего класс *OrderOnAdmission* с классом *TransferOrder*.

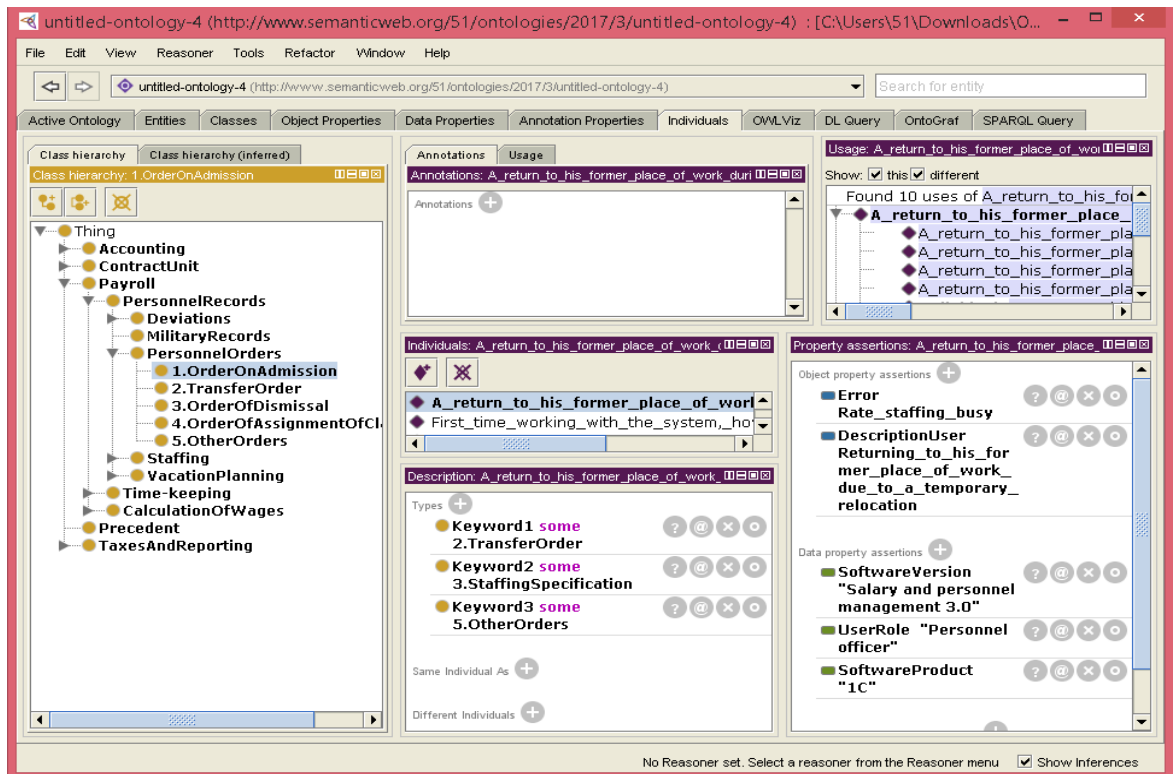


Рисунок 2.10. Панель описания класса «Order on admission»

При создании индивиды имеют только имя, в дальнейшем в процессе описания им задаются свойства.

Свойства Datatype связывают индивида со значением типа данных схемы XML или rdf. Другими словами, они описывают связи между индивидом и значениями данных. Свойство datatype может использоваться для индивида, относительно конкретных значений данных, которые могут быть типизированными и не типизированными. Будем использовать свойства datatype для описания программного продукта, версии программы и пользовательской роли. Создадим свойство типа данных *SoftwareVersion*, которое будет использоваться для описания версии ПП (см. рис.2.11).



Рисунок 2.11. Свойства типа данных (Datatype)

Создадим несколько примеров экземпляров программных продуктов с конкретными значениями. Так свойство *SoftwareVersion* используется для описания версий возможных программ (Salary and personnel management 3.0, Enterprise accounting 2.0, Enterprise accounting 3.1). На рис. 2.12 фрагмент описания индивида «How_to_make_sick_leave_and_maternity_leave?» свойством типа данных *SoftwareVersion*= Salary and personnel management 3.0

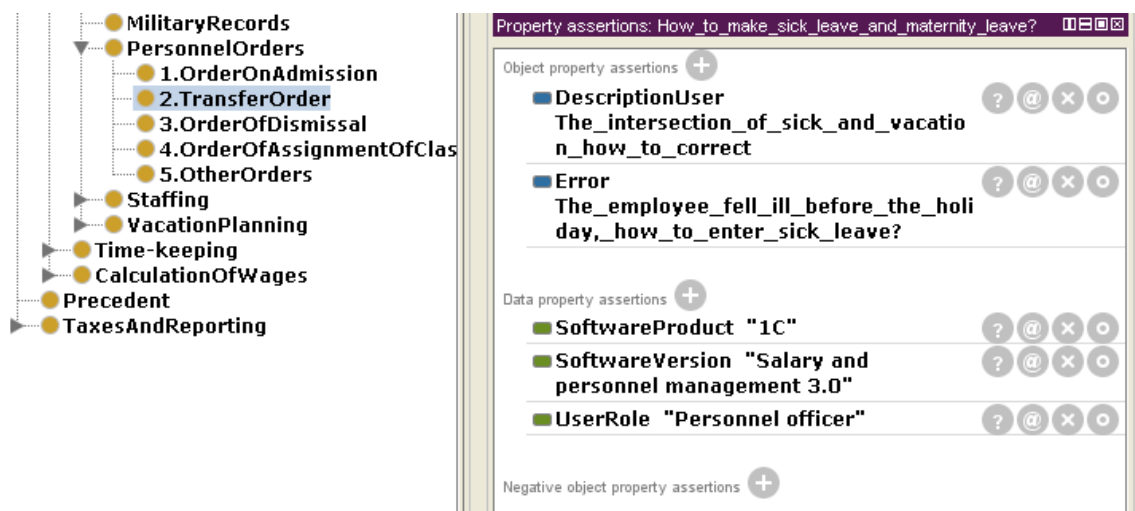


Рисунок 2.12. Свойство SoftwareVersion

В языке OWL можно создавать аннотации с различной информацией (описание класса, автор, дата модификации онтологии и т.п.), и метаданными классов, свойств, индивидов и онтологии (онтологии заголовка). Добавим различные свойства аннотаций в онтологии. Теперь добавим аннотацию к ограничениям, выбрав команду «@» – изменить свойства аннотации, находящуюся на панели описания классов в поле выбранного ограничения (рис. 2.13).

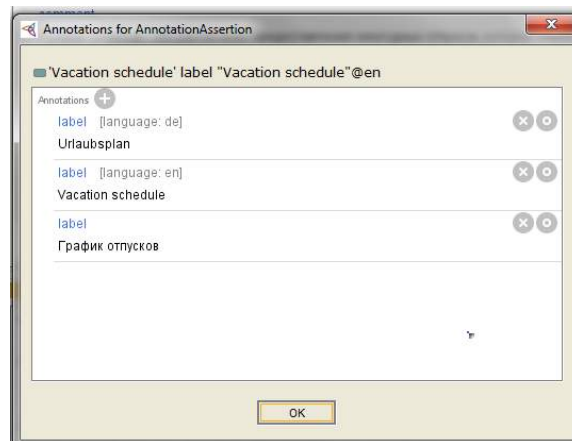
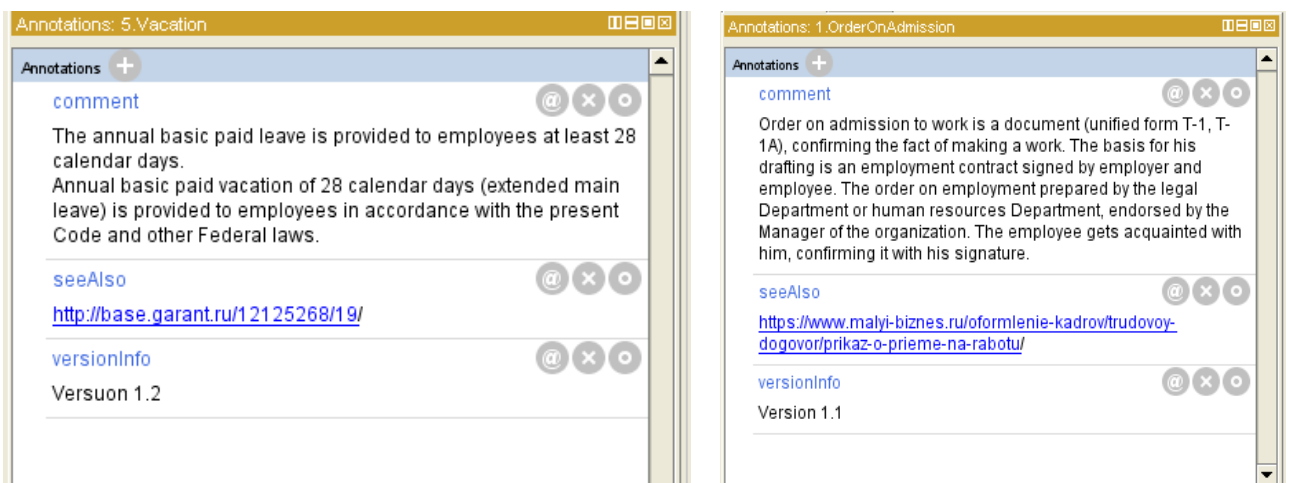


Рисунок 2.13. Панель, описывающая многоязычные имена класса *Vacation*

На рис. 2.14а изображена панель аннотации к классу *Vacation*, на рис. 2.15б – к классу *OrderOnAdmission*.



а)

б)

Рисунок 2.14. Панель описания аннотации к классам *Vacation* и *OrderOnAdmission*

На рис. 2.15 показан пример описания аннотации свойства «*rdfs:seeAlso*», определенного для индивида «*How_to_make_a_new_employee?*», который

наполняет класс *OrderOnAdmission*. Панель описания индивида «How_to_make_a_new_employee?» включает в себя: дерево классов, свойства объекта и аннотацию.

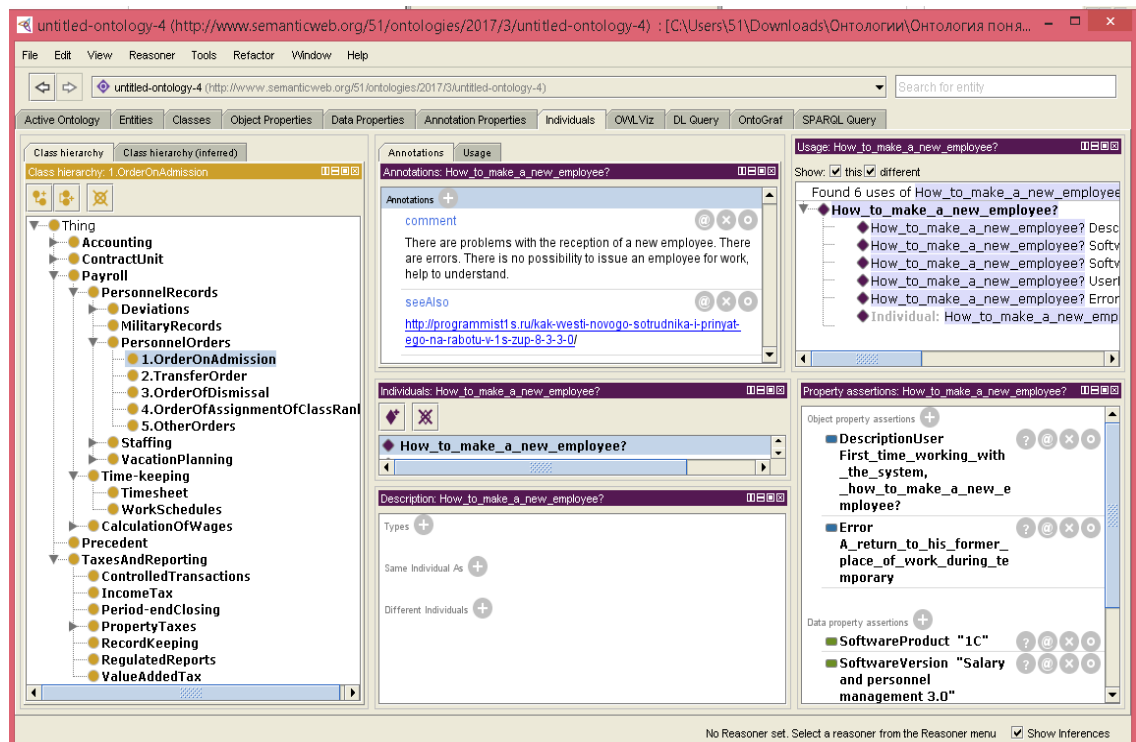


Рисунок 2.15. Описание индивида «How_to_make_a_new_employee?»

При нажатии ссылки, определяющей свойство «*rdfs:seeAlso*», пользователь автоматически попадает на web-ресурс с информацией о приказах о приеме на работу (см. рис. 2.16).



Рисунок 2.16. Ресурс сайта, определенный через свойство «*seeAlso*»

Под индивидами понимаются экземпляры классов. В Protégé индивиды определяются при помощи аксиом индивидов (фактов). Существует два вида

фактов. Первый – факты членства индивидов в классах и факты о значениях свойств индивидов. Второй – факты идентичности или различности индивидов.

Пример аксиом первого вида:

<Order of dismissal rdf:ID= «The order of dismissal of employee №12»>

<HaveKeyword1 rdf:resource= «Dismissal»>

</Order of dismissal >

Данная аксиома содержит два факта:

- существует некоторый индивид класса *OrderOfDismissal*, имеющий имя «The order of dismissal of employee №12»;
- этот индивид связан свойством *Keyword1* с индивидом *Dismissal* (который определен где-то в другом месте).

Первый факт говорит о членстве в классе, второй – о значении свойства индивида.

Аксиомы второго вида необходимы для суждения об идентичности индивидов. В языке OWL не делается никаких предположений ни о различии, ни о совпадении двух индивидов, имеющих различные идентификаторы URI. Подобные утверждения выражаются аксиомами идентичности с помощью следующих конструкций: «owl:sameAs» постулирует, что две ссылки URI ссылаются на один и тот же индивид; «owl:differentFrom» постулирует, что две ссылки URI ссылаются на разные индивиды; «owl:AllDifferent» предоставляет средство для определения списка попарно различных индивидов.

Для создания индивидов в Protégé 4.2 используется закладка Individuals. В панели описания индивидов были определены объекты, которые наполняют классы, а также заданы им свойства. Для примера, в приложении А на рис. А.3 показано, что индивид

«A_return_to_his_former_place_of_work_during_temporary» является экземпляром подкласса *OrderOnAdmission* класса *PersonnelOrders*. Ему присвоены следующие свойства: Error = Rate staffing busy, DescriptionUser = Returning_to_his_former_place_of_work_due_to_a_temporary_relocation, SoftwareVersion = Salary and personnel management 3.0, UserRole=Personnel

officer, Software Product = 1C, Keyword1= *TransferOrder*, Keyword2= *StaffingSpecification*, Keyword3= *OtherOrder*, на рис. А.4 представлен пример еще одного индивида.

SPARQL (Protocol and RDF Query Language) – язык запросов к данным, представленным по модели RDF, а также протокол для передачи этих запросов и ответов на них [80].

Рассмотрим некоторые типы запросов к данным:

1) *Выполнение запросов «по умолчанию»*

При открытии закладки SPARQL Query в редакторе Protégé 4.2 в окне формирования запросов уже представлен образец запроса «по молчанию»:

```
SELECT ?subject ?object
```

```
WHERE { ?subject rdfs:subClassOf ?object }
```

В результате выполнения запроса переменной ?subject в соответствие будут установлены классы, для которых характерно наличие свойства subClassOf. То есть классы, которые являются потомками для других классов. Результат запроса показан на рис. 2.17.

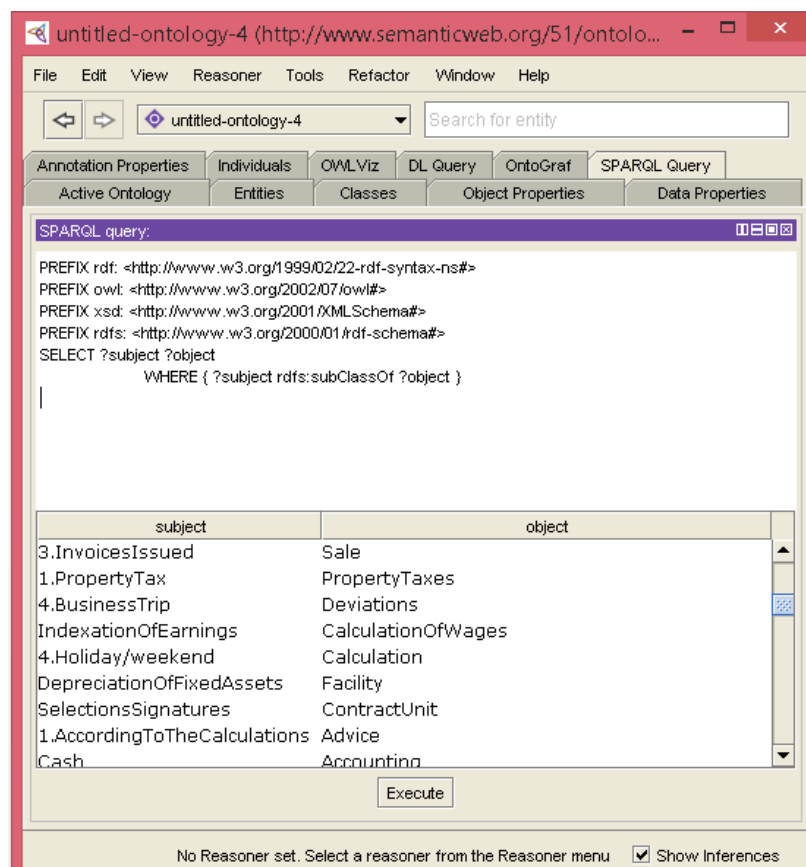


Рисунок 2.17. SPARQL Query «по умолчанию»

1) Запросы с применением данных о литералах

Выполним запрос, который вернет только те классы, которые являются подклассами «PersonnelRecords»:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT ?subject ?class
```

```
WHERE { ?subject rdfs:subClassOf ?class.
```

```
?class rdfs:label "PersonnelRecords" @en }
```

В результате выполнения запроса будут возвращены потомки класса, для которого установлена метка " PersonnelRecords " @en. Результат запроса показан на рис. 2.18.

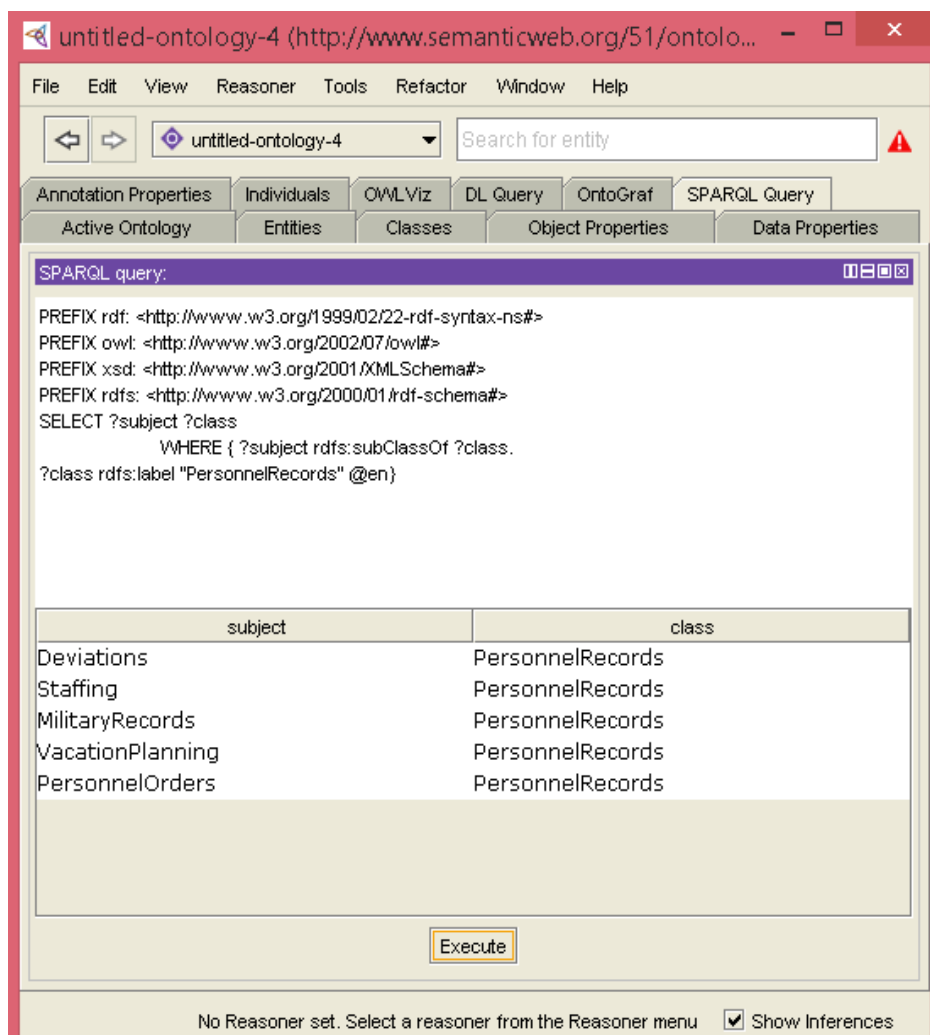


Рисунок 2.18. SPARQL Query с применением данных о литералах

Выполним запрос, используя информацию о комментарии (см. рис. 2.19):

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?subject ?class

WHERE { ?subject rdfs:subClassOf ?class.

?class rdfs:comment "Payroll — the process of calculation of payment of hired workers in accordance with the terms of the employment contract and deduction of taxes and other deductions under legislation of the Russian Federation, as well as their documentation" @en }

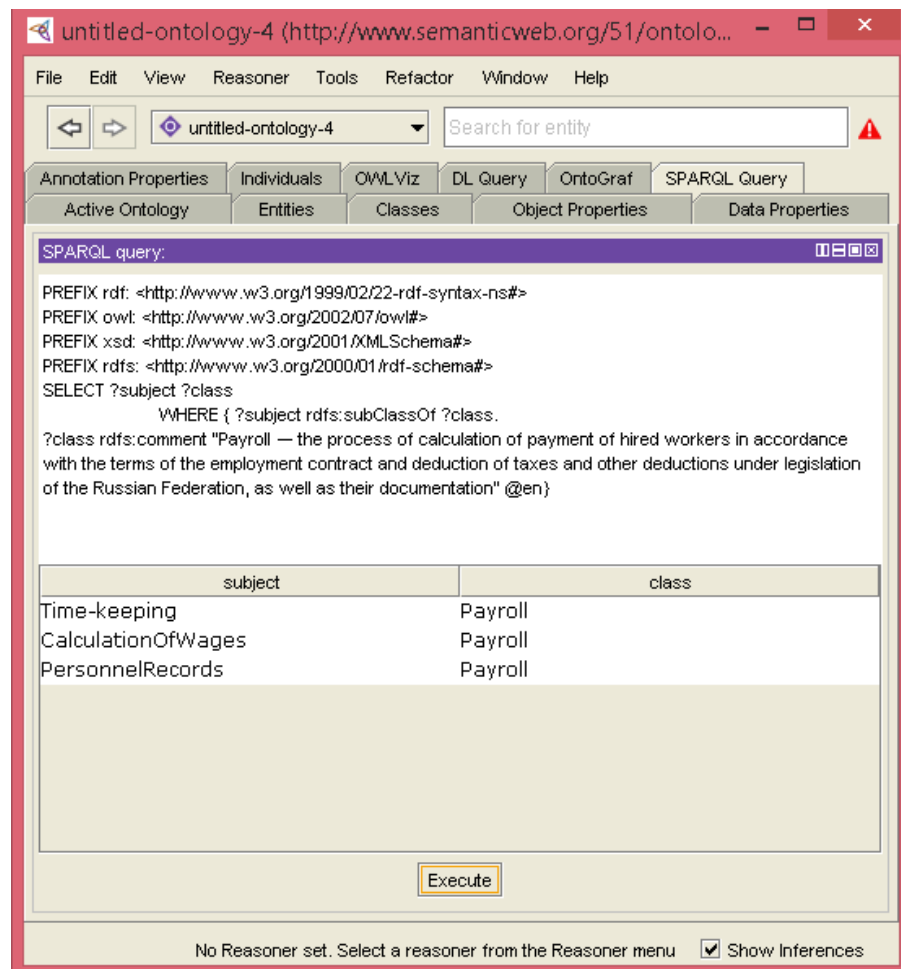


Рисунок 2.19. SPARQL Query с применением комментария

В результате выполнения запроса будут возвращены потомки класса, для которого установлен комментарий "Payroll — the process of calculation of payment of hired workers in accordance with the terms of the employment contract

and deduction of taxes and other deductions under legislation of the Russian Federation, as well as their documentation"@en.

Онтологический граф – двудольный граф, вершинами которого являются понятия предметной области, а дугами – отношения между ними. Двудольный граф – однонаправленный ориентированный граф, в одну вершину которого может входить и выходить несколько дуг [80].

Онтологический граф является информационной моделью предметной области, имеющий вид ориентированного графа, вершинами которого являются классы, а дугами отношения или связи. На рис. 2.20 представлен онтологический граф, полученный в результате построения онтологии предметной области в редакторе Protégé 4.2.

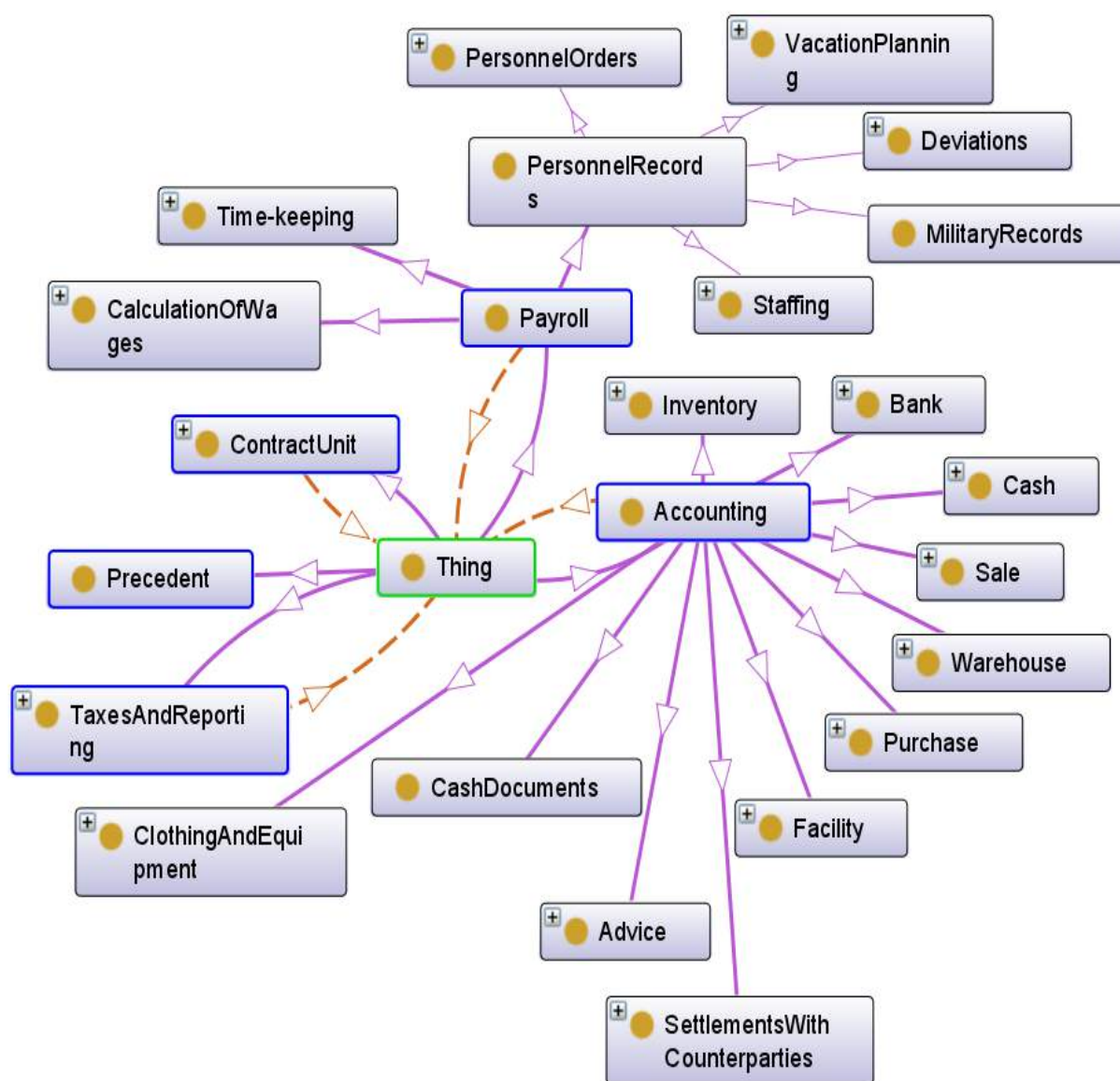


Рисунок 2.20. Онтологический граф

Онтологический граф, как модель предметной области, используется для представления декларативных знаний и обладает свойствами интерпретируемости и связности. Благодаря этим свойствам снижаются объемы хранимых данных, и вывод умозаключений осуществляется по ассоциативным связям.

В следующем подразделе представим описание класса Precedent и его реализацию.

2.2. Описание класса Precedent и его реализация

Рассуждения на основе аналогии, или метод прецедентов (CBR) – это подход, который позволяет решить новую проблему, используя или адаптируя решение, ранее уже принятое в аналогичной ситуации. Когда рассматривается новая ситуация, система находит подобный прецедент в базе знаний как аналог решаемой задачи и пытается использовать решение найденного прецедента. Если необходимо, близкий прецедент адаптируется к текущей ситуации. После применения решения, полученного на основе рассуждения по аналогии, к текущей проблеме, осуществляется анализ полученных результатов, после чего новый прецедент заносится в базу прецедентов для его использования в будущем.

Прецедент, как единица знаний включает следующие элементы:

- описание ситуации;
- решение, которое было принято в этой ситуации;
- результат применения решения [40].

Описание ситуации содержит по возможности всю информацию, которая необходима для достижения цели (выбора наиболее подходящего решения).

Решение содержит: набор операций, которые необходимо выполнить для получения успешного результата, т. е. для решения вопроса.

Результат применения решения – это обратная связь, которая возникает при применении решения к текущей ситуации. Прецедент должен содержать либо положительный, либо отрицательный исходы. Даже если проблему не удалось решить при использовании адаптированного прецедента в текущей

ситуации, эта информация (отрицательный исход) полезна и может быть использована для дальнейшего анализа. Удалять такой прецедент не следует, так как более глубокий анализ может показать направление дальнейших действий для решения вопроса. Достаточно поместить прецедент в отдельный класс прецедентов с отрицательными исходами [65].

Рассмотрим механизм принятия решения на основе прецедентов, представленный на рис. 2.21.



Рисунок 2.21. Механизм принятия решения на основе прецедентов

Поступил от пользователя некоторый запрос, с описанием текущей проблемы. Этот запрос сопоставляется с описанием прецедентов из базы знаний с помощью различных методов. Метод, с помощью которого осуществляется вычисление меры подобия (сходства) прецедентов, задается во время создания CBR-системы разработчиками. Наиболее популярным и часто используемым методом является поиск ближайшего соседа, в основе которого лежит способ измерения степени совпадения значений атрибутов (свойств), определяющих прецедент [1,56,89]. Решение из подходящего прецедента применяется к текущей ситуации и формируется результат применения решения, который также фиксируется в прецеденте. Как только принято решение на основе уже имеющихся (хранимых) прецедентов, соответствующая информация упаковывается в контейнер, называемый прецедентом, и

сохраняется в хранилище прецедентов для последующего использования. Ситуация, для которой был сохранен прецедент, считается опорной, или базовой.

Проблема представления прецедента – прежде всего проблема выбора информации, которую надо включать в описание прецедентов, нахождение соответствующей структуры для описания его содержания, а также определения, каким образом должна быть организована база знаний прецедентов для эффективного поиска и многократного использования.

Первоначальное описание прецедента может быть простым (линейным):

$$Case = (x_1, x_2, x_3, \dots, x_n, d) \quad (2.4)$$

где $x_1, x_2, x_3, \dots, x_n$ – значения атрибутов прецедента, идентифицирующих ситуацию; d – решение проблемы, определенное в прецеденте. В последующем, по мере углубления в проблемную область, возможно усложнение структуры прецедента, введение иерархических и других отношений между признаками.

Для интеграции онтологии предметной области с описанием прецедентов оказания ИТ-поддержки пользователям в онтологии был создан класс *Precedent*. Данный класс не обладает разветвленной иерархической структурой, как другие классы (концепты) онтологии, он вообще не имеет подклассов. Назначение класса *Precedent* – создать наиболее полную структуру для ввода информации о прецедентах консультирования (решения задачи пользователя), вводимой консультантом-аналитиком, а также установить связь с онтологией предметной области. Данный класс включает три группы свойств (слоты) – *Main*, *Changes* и *Files*, цель которых – структурно и содержательно разделить информацию, включенную в описание прецедента (см. рис. 2.22).

Слот *Main* имеет следующие подчиненные слоты:

- *Decision* (решение) – полное описание последовательности действий пользователя (технологии) для решения проблемы;
- *Descripton User* (описание пользователя) – информация о проблеме, которую пользователь дает консультанту при составлении запроса;

– *Error* (ошибка) – техническая ошибка, которая может быть решена только путем перепрограммирования (заполнено или нет);

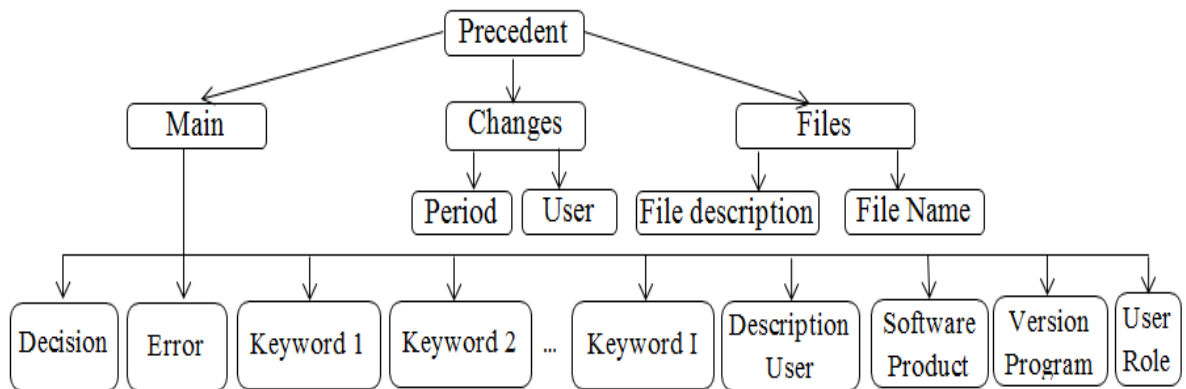


Рисунок 2.22. Свойства прецедентов

– *Keyword 1...3* (ключевые слова 1...3) – один или несколько слотов ключевых слов, характеризующих проблему. По этим слотам прецедент связан с онтологией;

– *Software Product* (программный продукт) – программный продукт, где возникла ошибка пользователя, выбор из списка (1С, Ахарт и т. д.);

– *User Role* (роль пользователя) – пользователь может быть сотрудником отдела кадров, бухгалтером, табельщиком, главным бухгалтером, заместителем главного бухгалтера, аудитором и т. д. Функциональность, которая может быть использована для решения проблемы пользователем, зависит от роли пользователя;

– *Version Program* (версия программного продукта) – релиз или версия. Программные продукты постоянно обновляются, разработчики исправляют ошибки, поэтому, прежде чем ответить на вопрос пользователя, необходимо понять, на каком релизе работает пользователь.

Слот *Changes* класса *Precedent* полезен для случая, когда несколько консультантов работают с базой данных. Всегда можно понять, кто и когда вносил изменения в прецедент. Данный слот имеет следующие подчиненные слоты:

– *Period* (период) – дата и время, когда был создан прецедент или были внесены изменения;

– *User* (консультант) – имя пользователя, внесшего изменения.

Слот *Files* имеет следующие подчиненные слоты:

- *Files Description* (описание файла) – краткое описание файла;
- *File Name* (имя файла) – путь к файлу, прикрепленному к прецеденту.

Это может быть файл с ошибкой, которая возникает в этом запросе, или файл с инструкцией по устранению неполадок.

Свойства онтологии для реализации СППР представляют собой свойства класса *Precedent*. Свойства объектов представлены на рис.2.23.

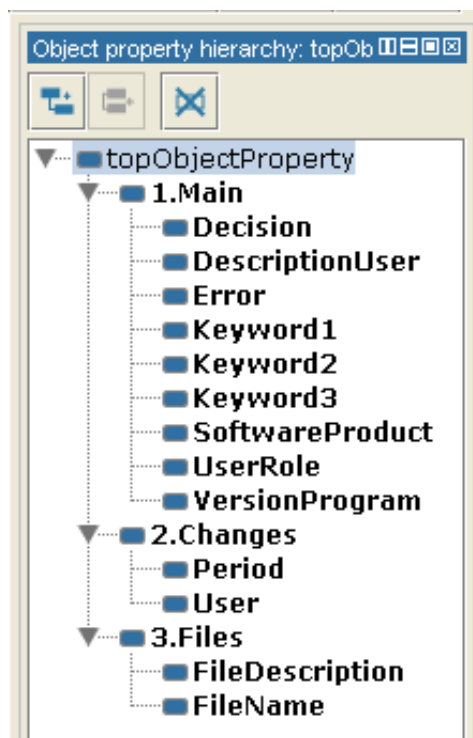


Рисунок 2.23. Свойства объектов

Структура прецедента, которая была описана выше, имеет необходимую полноту и избыточность, т. к. она описывает основные характеристики запроса пользователя: решение, описание пользователем, ошибка, набор ключевых слов, программный продукт, роль пользователя и версия программного продукта. Таким образом, мы определили место, где возникла проблема (программный продукт и его версии), кто столкнулся с проблемой (роль пользователя), как пользователь описывает свою проблему (описание пользователем, ошибки). Консультант дает профессиональное описание, характеризующее проблему пользователя, определяет место проблемы в онтологии с помощью ассоциативных связей с концептами онтологии. Прецедент также содержит сведения о внесении изменений в прецедент: дата,

когда изменения были сделаны и кем, так что есть возможность проанализировать внесенные изменения. К прецеденту можно прикрепить файл, который содержит инструкции по решению проблемы, или ошибки пользователя, которые могут быть приобщены к прецеденту. Этой информации достаточно, чтобы решить проблему пользователя и найти подходящий прецедент.

2.3. Интеграция прецедентов с концептами онтологии

В традиционном методе CBR для извлечения прецедентов используется мера близости (расстояние) в многомерном пространстве, задаваемом атрибутами прецедента. В этом случае в базе прецедентов производится поиск случая, наиболее близкого к текущей проблеме для заданного подмножества значений атрибутов, после чего найденный прецедент (прецеденты) адаптируется к текущей ситуации. Однако далеко не всегда наиболее близкий (в терминах расстояния в многомерном пространстве) прецедент является наиболее релевантным, когда достигается наилучшее семантическое соответствие найденного прецедента текущей проблеме. Нам представляется перспективным осуществлять сравнение между текущей ситуацией и прецедентами, оценивая степень их связи с концептами онтологии. Это означает, что близость прецедентов друг к другу оценивается степенью семантической близости связанных с этими прецедентами концептов. Для этого на этапе создания базы знаний при внесении каждого нового прецедента необходимо установить смысловые связи вновь вводимого прецедента с концептами онтологии.

Связь экземпляров класса *Precedent* с концептами онтологии осуществляется путем задания ассоциативного отношения R_{ASS} для слотов группы *Keywords* класса *Precedent* (слота *Main*). Отношение устанавливается путем явного указания в качестве значения слота класса *Precedent* имени, связанного с ним класса — концепта онтологии, а также типа связи, существующей между этими классами. Для реализации указанной

ассоциативной связи R_{ASS} в качестве типа группы слотов *Keywords* используется тип D_{class} (тип *Class*).

Задание типа D_{class} для каждого из I слотов *Keywords* предполагает указание дополнительного аргумента – ассоциированного класса, концепта онтологии. Если, например, i -й слот группы *Keywords* имеет тип D_{class} с ассоциированным классом c_i , то в качестве значений слота при создании экземпляров класса *Precedent* могут быть использованы классы множества $Tr(c_i)$ – транзитивного замыкания концепта c_i по отношению R_{ISA} , включающего класс $c_i = c_i^{(0)}$ и все его подклассы ниже по иерархии:

$$Tr(c_i) = \{c_i = c_i^{(0)}\} \cup ISA(c_i^{(0)}) \quad (2.5)$$

где $ISA(c^{(0)}) = \bigcup_{l=1}^L \{c^{(l)} \in C \mid \exists R_{ISA}(c^{(l-1)}, c^{(l)})\}$ L – максимальная глубина

потомков класса c_i . В этом случае классы *Precedent* и c_i связаны ассоциативным отношением $R_{ASS}(Precedent, c_i)$.

Устанавливая связи некоторого прецедента с онтологией, аналитик выбирает концепты, наиболее близкие по смыслу с данным прецедентом. Это могут быть как терминальные (не имеющие потомков), наиболее четко семантически оформленные концепты, так и нетерминальные (промежуточные) концепты, имеющие более общий смысл. Необходимость в нетерминальных концептах возникает в том случае, если возникающая проблема не может быть однозначно отнесена к терминальному концепту, или аналитик не имеет достаточного опыта и ему проще отнести прецедент к более общему по смыслу понятию.

Необходимо подчеркнуть, что в нашем подходе мы допускаем установление не одной, а нескольких связей прецедента с концептами онтологии. Это расширяет выразительные возможности нашего подхода и может быть использовано в случае, когда проблема возникает на стыке

нескольких понятий, и ее адекватное описание требует учета этой междисциплинарности.

Пусть в качестве фасета для i -го слота группы *Keywords* задается весовое значение $v_i, 0 \leq v_i \leq 1, \sum_{i=1}^I v_i = 1$, устанавливающее силу связи прецедента с соответствующим концептом онтологии. Вес v_i тем больше, чем ближе по смыслу прецедент к данному понятию предметной области.

Рассмотрим теперь, как можно организовать процедуру классификации и извлечения семантически близких прецедентов, используя вышеописанную интегрированную модель. Мы будем различать терминальные и нетерминальные концепты. Назовем терминальные концепты онтологии ключевыми словами. Пусть в онтологии имеется J ключевых слов $kw_j, j = \overline{1, J}$. Поставим в соответствие каждому прецеденту вектор размерности J весовых коэффициентов $w_j, j = \overline{1, J}$.

Процедуру вычисления весовых коэффициентов можно организовать следующим образом. Без ограничения общности предположим, что концепты онтологии, с которыми связан прецедент, не входят в транзитивное замыкание друг друга (т.е. они не должны быть расположены на одной иерархической ветке). Данное предположение вполне естественно, так как если мы можем отнести прецедент к более конкретному концепту-потомку, то отпадает необходимость в его связи с более общим концептом-предком. В этом случае процедура формирования вектора весовых коэффициентов $w_j, j = \overline{1, J}$, ключевых слов $kw_j, j = \overline{1, J}$, может быть представлена следующим образом.

*Процедура вычисления весов связи прецедента с терминальными
концептами онтологии*

Предположим, что рассматриваемый прецедент связан с концептами C_1, C_2, \dots, C_I посредством слотов $Keyword_1, Keyword_2, \dots, Keyword_I$.

1. Положим сначала $w_j = 0, \forall j = \overline{1, J}$.

2. Для всех связанных с прецедентом концептов $\overline{C_i}, i = \overline{1, I}$, определим значения весов терминалов kw_j , входящих в транзитивное замыкание $\overline{C_i}$:

$$w_j = \begin{cases} v_i, & \text{если } C_i \text{ является терминальным концептом } (kw_j = C_i = C_i^{(0)}) \\ v_i \cdot \prod_{l=1}^L v_i^{(l)}, & \text{не является терминальным концептом } (kw_j = C_i^{(L)}, L > 0) \end{cases} \quad (2.6)$$

где w_j - веса терминальных концептов онтологии $\sum_{j=1}^J w_j = 1$, $v_i^{(l)}$ - вес связи $R_{ISA}(c_i^{(l-1)}, c_i^{(l)})$ от концепта-родителя $c_i^{(l-1)}$ к концепту-ребенку $c_i^{(l)}$ на пути от ассоциированного с прецедентом концепта C_i к терминальному концепту kw_j . Иллюстрация последнего варианта вычисления представлена на рис. 2.24.

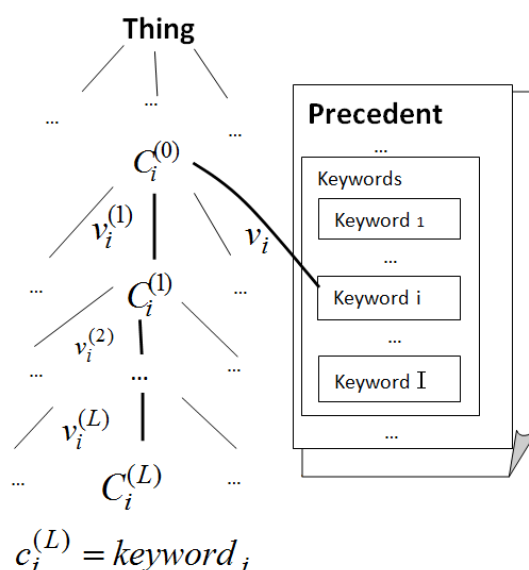


Рисунок 2.24. Вычисление силы связи прецедента с ключевыми словами. Веса концептов, подчиненных одному родителю онтологии, считаются одинаковыми.

Проиллюстрируем вычисление весов ключевых слов на конкретном примере. Предположим, что мы имеем в базе знаний конкретный прецедент, связанный только с концептами, принадлежащими транзитивному замыканию класса *PersonellOrders* по отношению R_{ISA} . Фрагмент онтологии с полностью развернутым классом *PersonellOrders* (до терминальных концептов) представлен на рис.2.25.

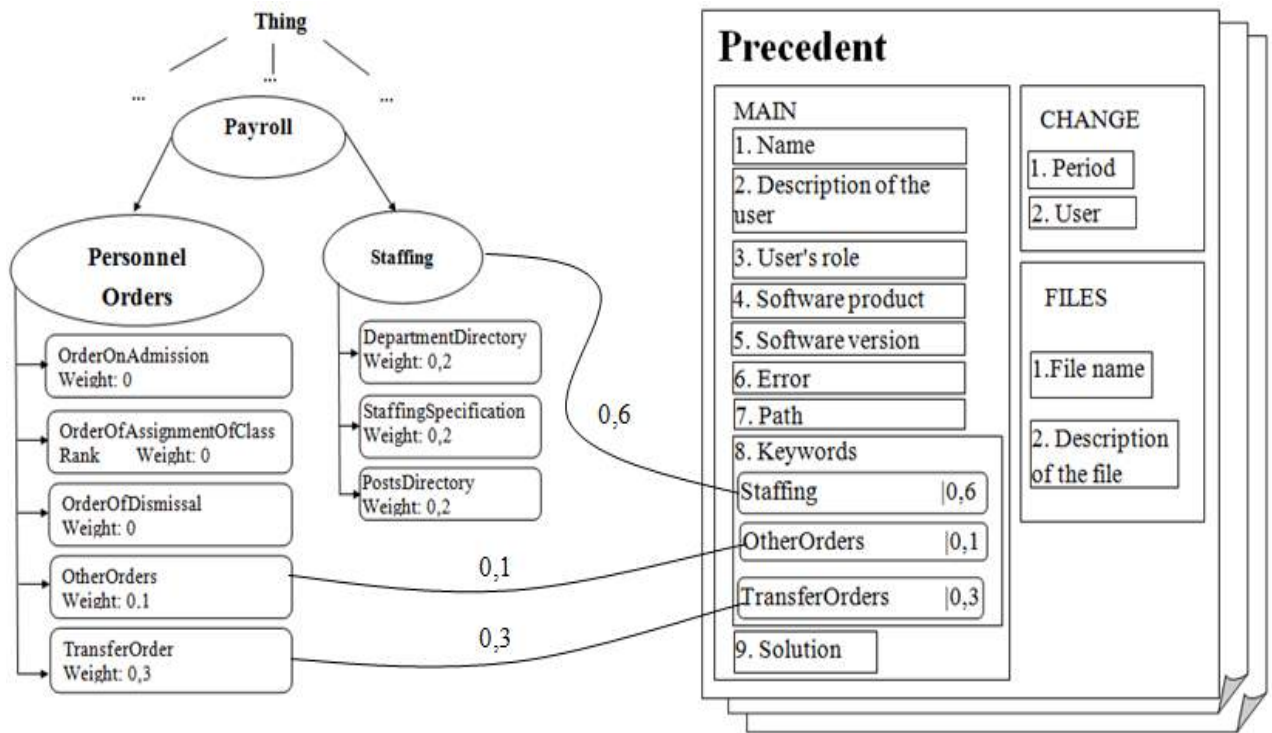


Рисунок 2.25. Пример вычисления силы связи прецедента с ключевыми словами

Эксперт-аналитик установил три ассоциативные связи от прецедента к онтологии. Две связи установлены к терминальным концептам *TransferOrder* и *OtherOrder*. Веса ключевых слов *TransferOrder* и *OtherOrder* соответственно полагаются равными соответствующим установленным аналитиком весам $v_1 = 0,3$ и $v_2 = 0,1$. Третья связь установлена к нетерминальному концепту *Staffing*, являющемуся родителем трех терминальных концептов – ключевых слов *DepartmentDirectory*, *PostDirectory* и *StaffingSpecification*. Соответственно и вес $v_3 = 0,6$, установленный аналитиком для связи прецедента с концептом *Staffing*, делится на данные три ключевых слова поровну. Соответствующие данным ключевым словам веса равны $w_j = 0,6 \cdot 1/3 = 0,2$.

Остальные ключевые слова остались несвязанными с прецедентом, и соответственно, имеют по отношению к нему нулевые веса. Таким образом, для подмножества из восьми терминальных концептов (ключевых слов) *OrderOnAdmission*, *TransferOrder*, *OrderOfAssignmentOfClassRank*, *OtherOrder*, *OtherOfDismissal*, *DepartmentDirectory*, *PostDirectory* и *StaffingSpecification*

представленных на рис. 2.4 имеем следующий подвектор весов, соответствующий рассматриваемому прецеденту

$\tilde{w} = (0 \ 0,3 \ 0 \ 0,1 \ 0 \ 0,2 \ 0,2 \ 0,2)^T$. Так как прецедент имеет связи только с концептами этого фрагмента онтологии, то остальные ключевые слова имеют нулевые веса. В приложении В (табл. В.1) представлен фрагмент набора прецедентов с их весами.

Фрагмент онтологии, реализованный в Protégé OWL с выделенным классом *Precedent*, списком хранящихся экземпляров прецедентов и концептам, соответствующим текущему прецеденту, представлен на рис. 2.26.

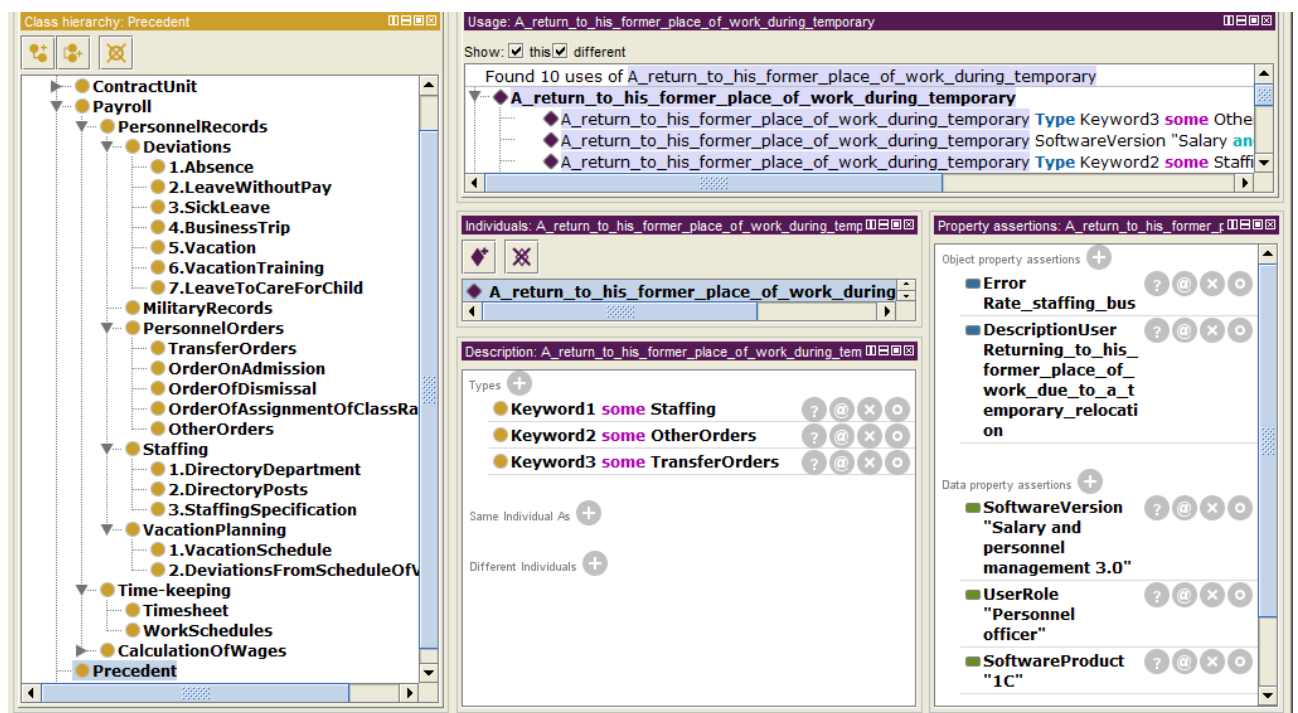


Рисунок 2.26. Фрагмент онтологии в Protégé OWL с классом Precedent

В результате применения вышеописанного алгоритма мы получаем таблицу данных большой размерности, число строк которой равно числу прецедентов в базе знаний, а число столбцов равно числу ключевых слов – терминальных концептов онтологии. Каждое ключевое слово включено в представление прецедента с весом, рассчитанным на основе ассоциативных связей между прецедентом и понятиями онтологии.

К данной таблице возможно применение методов анализа данных для сокращения размерности, кластерного анализа, с целью подготовки базы

знаний к реализации методов извлечения групп прецедентов, семантически наиболее близких к решаемой проблеме.

2.4. Выводы

Основными результатами главы 2 являются:

1. Построена онтология наиболее востребованных у ИТ-пользователей предметных областей, позволяющая установить более тесное взаимодействие консультантов-аналитиков и специалистов, пользующихся программным обеспечением предприятий. Онтологическое моделирование позволило описать рассматриваемые области знаний на концептуальном и логическом уровнях, а также графически в виде онтологического графа, в котором наглядно продемонстрированы все объекты предметной области и связи между ними. Концептуальное описание объектов, входящих в предметную область, позволило избавиться от противоречий, многозначности и избыточности информации.

2. Построена модель прецедента, позволяющая сохранить в базе знаний не только информацию о возникшей у ИТ-пользователя проблеме (инциденте), но также и о принятом в данном случае решении, и о последствиях этого решения. Произведено наполнение базы прецедентов экземплярами класса *Precedent* – конкретными ситуациями принятия решений при консультировании пользователей. Апробация созданной базы прецедентов показывает, что предложенная структура обладает свойствами полноты и неизбыточности.

3. Предложена гибридная модель интеграции прецедентов с концептами онтологии, являющаяся основой для разработки метода извлечения прецедентов, семантически близких к решаемой проблеме. Результатом предложенной интеграции является матрица коэффициентов семантических связей прецедентов с терминалами онтологии, позволяющая использовать методы интеллектуального анализа данных для извлечения и последующей адаптации прецедентов.

ГЛАВА 3. МЕТОД ИЗВЛЕЧЕНИЯ РЕЛЕВАНТНЫХ ПРЕЦЕДЕНТОВ НА ОСНОВЕ ГЕНЕРИРОВАНИЯ СИСТЕМЫ НЕЧЕТКИХ ПРАВИЛ КЛАССИФИКАЦИИ

Данная глава посвящена разработке и исследованию алгоритма машинного обучения на основе генерирования нечетких правил классификации, который используется для поддержания предлагаемой информационной технологии в системе поддержки принятия решений в области ИТ-консультирования. Кластерный анализ данных, полученных при индексировании прецедентов концептами онтологии, позволяет разделить все множество прецедентов на классы семантически близких прецедентов. Предложенный алгоритм извлекает прецеденты, семантически близкие к текущей проблеме пользователя. Полученные в этой главе результаты отражены в публикациях [1,41,51,52,113,114,116,156].

3.1. Цикл управления знаниями на основе интеграции прецедентной и нечеткой моделей

Классические подходы к управлению в социальных и экономических системах строятся на том предположении, что процесс управления можно описать строгой аналитической формулой, пусть даже она будет очень сложной. Однако применение таких подходов возможно только для управления простыми объектами. Управление знаниями – достаточно сложная область, в которой классические подходы не работают, так как принятие решений в этой области происходит в условиях неопределенности. В связи с этим, в настоящее время исследуются подходы к управлению объектами на основе интеграции моделей и методов интеллектуального анализа данных. Мы считаем, что наиболее подходящими методами для решения задачи поддержки принятия решений является интеграция прецедентной модели и нечеткой логики, так как при использовании такого интегрированного подхода появляется возможность извлечения из прецедентов нечетких правил, которые специалист может в дальнейшем интерпретировать.

В прецедентном подходе для извлечения релевантных прецедентов необходимо оценить все прецеденты в базе знаний. Эффективность прецедентного подхода явно связана с размером базы прецедентов. Таким образом, целый ряд исследований был посвящен уменьшению времени для оценки прецедентов. Алгоритм К-средних является наиболее популярным методом. Прецедентный подход, интегрированный с методом К-средних, означает что кластеры всех прецедентов оцениваются только в том случае, когда извлечен наиболее похожий кластер. Кроме того, Chang and Lai [126] применяли самоорганизующиеся карты (COM). В их исследованиях, было определено, что интеграция самоорганизующихся карт и прецедентного подхода (COM/CBR) лучше, чем интеграция метода К-средних с прецедентным подходом (K/CBR). Оба метода K/CBR и COM/CBR повысили эффективность прецедентного подхода. Однако производительность этих двух механизмов CBR тесно связана с представлением прецедентов и индексацией близких прецедентов. [177].

Прецедентный подход был широко интегрирован с другими технологиями. Marling, Sqalli, Rissland, Hector, and Aha [157] исследовали различные интеграции прецедентного подхода. Некоторые системы, например, ADIOP, CADRE, CADSYN, CHARADE, COMPOSER, IDIOM, JULIA, и Weigel's, интегрируют прецедентный подход с алгоритмами по удовлетворению ограничений. Некоторые системы, например, ANAPRON, AUGUSTE проект, CAMPER, GREBE, GYMEL и SAXEX, объединяют прецедентный подход с системами на основе правил. Существуют также гибриды прецедентного подхода и модели на основе рассуждений, например, CASEY, FORMTOOL и SOPHIST. Эта интеграция может улучшить и повысить эффективность прецедентного подхода.

Мы предлагаем интегрировать прецедентный подход с теорией нечеткой логики. Рассмотрим основные понятия нечеткой логики. Нечеткая логика – это обобщение традиционной логики, в случае, когда истинность рассматривается как лингвистическая переменная, принимающая значения типа «очень

истинно», «более-менее истинно», «не очень ложно» и т.п. [102]. Указанные лингвистические значения представляются нечеткими множествами.

Под лингвистической переменной будем понимать переменную, которая для своего задания использует лингвистические значения, которые являются качественными характеристиками. Например, температура воды и скорость – это лингвистические переменные; «холодная», «прохладная», «теплая», «горячая» - это лингвистические значения (характеристики); «примерно 45 градусов», «около 60 км/ч» - это значения нечетких чисел.

Для определения некоторой лингвистической переменной используется лингвистическое терм-множество, которое содержит в себе все лингвистические значения. Под областью значений переменной понимается множество всех числовых значений, которые могут относиться к определенному параметру изучаемой системы, а также множество значений, которые важны с точки зрения решаемой задачи. Лингвистические значения представляются нечеткими множествами (НМ), которые представляют собой совокупность элементов произвольной природы, относительно которых нельзя с полной определенностью утверждать, что какой-либо элемент принадлежит данному множеству [150].

Введем необходимые понятия теории нечетких множеств, необходимые для построения интегрированной модели. Пусть имеется некоторое обычное (универсальное, или универсум) множество X элементов x . Нечеткое множество A определяется как упорядоченное множество пар вида $\langle x, \mu_A(x) \rangle$, где $x \in X$ – является элементом некоторого универсального множества X (универсума), $\mu_A(x)$ – функция принадлежности $\mu_A: X \rightarrow [0, 1]$. При этом $\mu_A(x) = 1$ для некоторого x означает, что элемент x определенно принадлежит нечеткому множеству A , а значение $\mu_A(x) = 0$ означает, что элемент x определенно не принадлежит нечеткому множеству A .

Формально конечное нечеткое множество записывается в виде

$$A = \{ \langle x_1, \mu_A(x_1) \rangle, \langle x_2, \mu_A(x_2) \rangle, \dots, \langle x_n, \mu_A(x_n) \rangle \} \quad (3.1)$$

Таким образом, нечеткая логика основана на рассуждениях, что решение является не точным, а приблизительным. Нечеткие логические переменные принимают значения истинности, которые расположены между 0 и 1 и не ограничиваются всего двумя значениями истинности, как в традиционной логике. Нечеткая логика является формой гибких вычислений, допускающих неточность, которые имитируют принятие решений человеком [85].

В диссертационной работе предложена интеграция прецедентной и нечеткой моделей для эффективного управления знаниями в организации. Обе модели обладают своими преимуществами и недостатками. Поэтому на практике правильно было бы воспользоваться преимуществами обеих моделей. Сочетание двух моделей можно представить в виде циклического преобразования знаний из одной формы в другую, как представлено на рис. 3.1 [116].



Рисунок 3.1. Трансформация знаний в гибридной модели

При создании СППР для управления знаниями на первоначальных этапах, когда еще нет глубокого представления о предметной области, целесообразно использовать прецедентную модель представления знаний, в которой знаниями являются реализованные в той или иной сфере работы организации актуальные прецеденты принятия решений (этап I). Для создания прецедента в самом простом случае надо определить конечный набор признаков (атрибутов),

однозначно определяющих ситуацию, и конкретное решение, принимаемое в данной ситуации.

Уже на самых ранних этапах развития системы управления знаниями возможно извлечение, адаптация и использование прецедентов для решения текущих проблем принятия решений. После применения прецедента к текущей ситуации новый прецедент записывается в базу знаний для дальнейшего использования. Отметим, что с точки зрения цикла Нонака-Такеучи [165] в прецедентной модели мы не проводим формализацию знаний (превращение неявного знания в явное). При этом мы пока не пытаемся проанализировать, на основании чего принято это решение (т.е. превратить неявное знание в формальное), а просто фиксируем факт применения неявного знания в конкретном принятом решении.

По мере достаточного накопления прецедентов для принятия решений и углубления наших знаний на основе анализа базы прецедентов в рассматриваемой предметной области становится возможным проводить формализацию знаний, способствуя превращению неявных знаний в явные. Большой интеллектуальной поддержкой здесь может оказаться применение методов интеллектуального анализа данных для автоматического извлечения новых знаний в виде правил из выборки прецедентов (этап II). В этом случае множество прецедентов может быть представлено в виде таблицы данных объект-свойство, к которой применяется, например, метод деревьев решений [171]. Другой метод формирования правил из множества прецедентов использует преобразование выборки прецедентов в систему нечетких правил. Таким образом, формируется множество правил, включающее все прецеденты базы знаний [51,52].

Переход к модели знаний в виде правил означает получение явной формы знаний, способной объяснять причинно-следственные связи в предметной области. Подобные заключения могут быть предъявлены экспертам для анализа и вынесения экспертного мнения. При этом явные знания могут быть очищены, уточнены и проинтерпретированы экспертами (этап III). Немаловажным

моментом на этом этапе является разрешение конфликтов, возникающее в системе правил.

На этапе IV происходит накопление новых прецедентов при использовании полученного варианта системы поддержки принятия решений. Новые прецеденты получаются при использовании базы правил, а также в результате CBR-цикла, адаптируя прецеденты из базы знаний к новым проблемам. Получение новых прецедентов на основе CBR-цикла означает применение формальных (явных) знаний и превращение их в неявную форму.

Рассмотренный цикл трансформации знаний повторяется далее на новом уровне. При этом возможно усложнение структуры прецедентов – переход от параметрической формы представления к иерархической или более сложной логической форме. Предложенная модель позволит эффективно поддерживать высоко динамичный процесс постоянного движения знаний в ИТ-организации.

3.2. Интеграция прецедентной и нечеткой модели

Общее представление системы, основанной на прецедентном подходе и нечеткой модели, представлено на рис.3.2. Работа системы начинается с описания текущей ситуации Q , в которой надо принять решение, с использованием тех же признаков, которые использовались для описания прецедентов C в базе прецедентов CB . Для текущей ситуации система должна найти релевантное решение в базе прецедентов [132,133,159]. Релевантность между текущей ситуацией и прецедентами в базе прецедентов устанавливается с использованием нечеткого логического вывода, использующего множество нечетких правил, которые содержат знания для оценки релевантности между случаями. Выбираются только те прецеденты из базы прецедентов CB , степень релевантности которых выше заданного порогового уровня α [1,45].

Таким образом, множество прецедентов, извлеченных из базы формулируются как

$$M = \{C \in CB \mid \text{Relev}(C, Q) \geq \alpha\} \quad (3.2)$$

где $\text{Relev}(C, Q)$ – функция сходства прецедента C с текущей ситуацией Q , вычисляемая на основе нечетких рассуждений. Выбранные прецеденты в M

передаются на следующий блок «слияния решений», где решения полученных прецедентов объединяются и адаптируются для разрешения текущей ситуации.

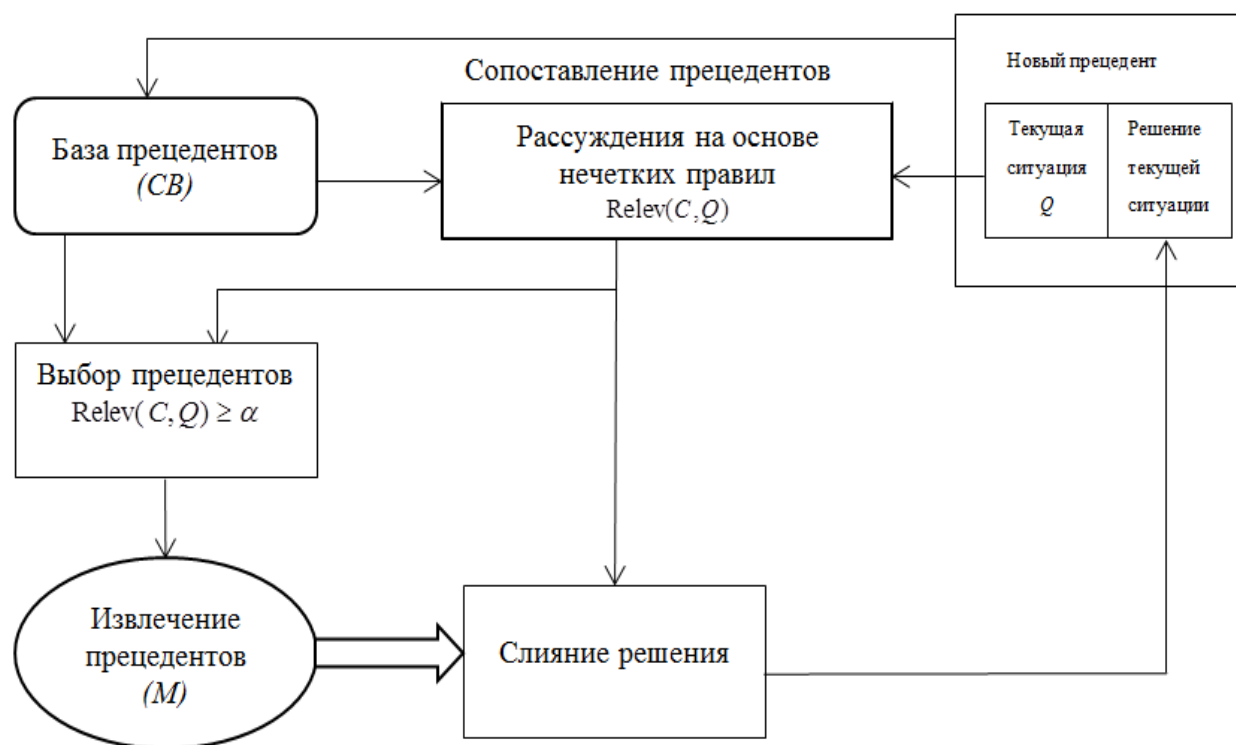


Рисунок 3.2. Схема работы системы на основе нечеткой логики

На этапе слияния решений происходит поиск новых решений для текущей проблемы путем модификации и обобщения известных решений, полученных из прецедентов. Подобие прецедентов определяется с помощью показателей полезности или целесообразности выбранных прецедентов для решения новой проблемы. Прецеденты с высоким значением меры подобия будут иметь больше влияния при определении окончательного решения.

Обобщенная схема работы нечеткого модуля представлена на рис. 3.3. Первоначально на вход системы поступает выборка прецедентов из базы прецедентов. Каждый из прецедентов характеризуется набором признаков. Для проверки работы алгоритма исходная выборка случайным образом делится на две под выборки – обучающую и контрольную. На первом этапе происходит настройка базового множества нечетких правил под имеющиеся объективные данные обучающей выборки прецедентов.

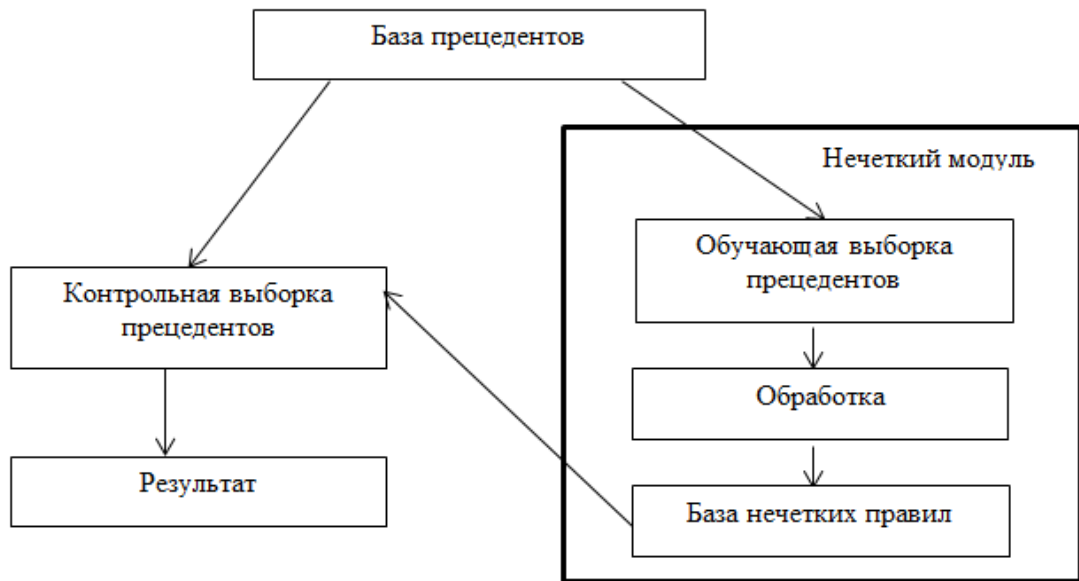


Рисунок 3.3. Обобщенная схема работы нечеткого модуля

На следующем этапе проводится проверка точности классификации прецедента на имеющейся контрольной выборке. Для этого каждый элемент контрольной выборки рассматривается как текущая ситуация, для которой на основе нечеткого вывода находится соответствующий прецедент из обучающей выборки. Далее полученное в результате такой процедуры решение сравнивается с решением прецедента, после чего делаются выводы о точности предсказания.

Интеграцию метода рассуждений на основе прецедентов с нечеткой логикой рассмотрим в следующей постановке. Предположим, что имеется выборка прецедентов, характеризующаяся рядом признаков (переменных), описывающих ситуацию, которые могут быть как количественными, так и качественными. В модели нечеткого вывода на рис.3.4 эти признаки описываются входными лингвистическими переменными A, B, C, \dots . Выборка содержит качественный признак, характеризующий решение прецедента. В нашей постановке данный признак будет относить прецедент к определенному классу решений (классообразующий признак). В модели нечеткого вывода на рис.3.4 классообразующий признак описывается выходной лингвистической переменной y . Таким образом, мы рассматриваем задачу классификации прецедентов – отнесения прецедентов к тому или иному классу решений.

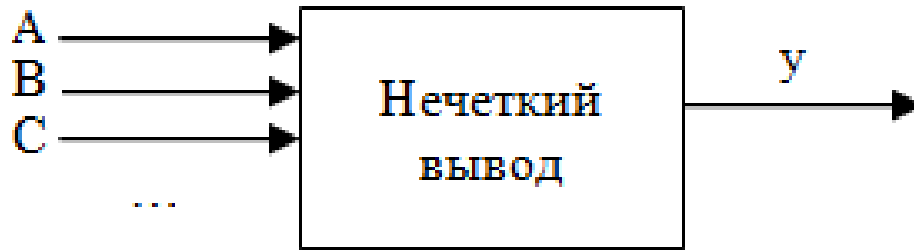


Рисунок 3.4. Модель нечеткого вывода

На рис. 3.5 представлена гибридная модель, полученная в результате интеграции метода рассуждений на основе прецедентов и нечеткой логики [1].

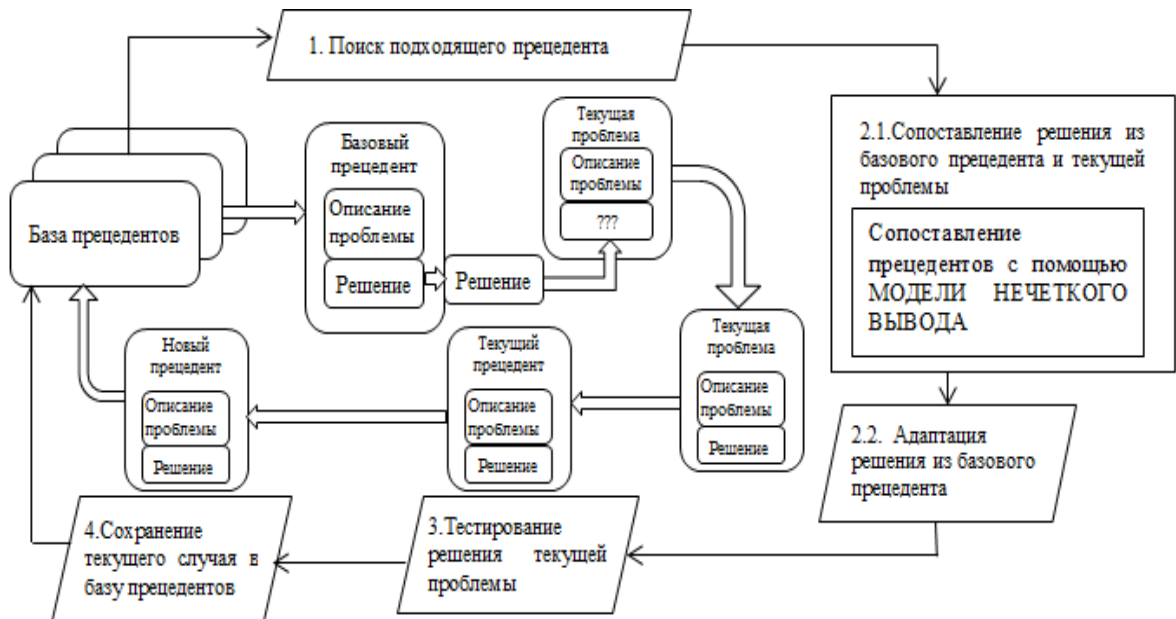


Рисунок 3.5. CBR и нечеткая модель

Кластерный анализ данных, полученных при индексировании прецедентов концептами онтологии, позволяет разделить все множество прецедентов на классы семантически близких прецедентов, и представить прецедент в виде

$$Case = (x_1, x_2, x_3, \dots, x_n, d) \quad (3.3)$$

где $x_1 \in X_1, x_2 \in X_2, \dots, x_n \in X_n$ – свойства, определяющие проблему пользователя, определенные на множествах $X_i, i = \overline{1, n}$, соответственно, $d \in D$ – класс семантически близких прецедентов из множества $D = \{d_1, d_2, \dots, d_K\}$.

В качестве свойств x_1, \dots, x_n могут выступать числовые признаки – веса ключевых слов, определяющих связи прецедента с понятиями предметной

области, а также дополнительные параметры из описания прецедента, имеющие в основном качественный тип.

Известным методом машинного обучения для решения задачи классификации является метод деревьев решений. В отличие от других методов классификации данный подход позволяет получать знания более высокого уровня – правила, применяемые для отнесения того или иного объекта к определенному классу. Таким образом, деревья решений позволяют автоматизировать задачу получения знаний в виде правил из прецедентов, интегрируя оба способа представления знаний в единую базу. Другим полезным свойством данного метода является возможность последующей интерпретации правил экспертами. Однако метод деревьев решений предполагает качественную природу признаков. Мы предлагаем метод классификации на основе формирования нечетких правил, базирующихся на фаззификации исходных количественных признаков нечеткими множествами.

В следующем разделе описывается алгоритм формирования нечетких правил на основе обучающей выборки из базы прецедентов.

3.3. Алгоритм классификации прецедентов на основе генерирования системы нечетких правил

В диссертационной работе разработан метод классификации на основе формирования нечетких правил, базирующихся на фаззификации исходных количественных признаков x_1, \dots, x_n нечеткими множествами. Определим лингвистические переменные V_i , соответствующие признакам $x_i, i = \overline{1, n}$. Пусть каждая переменная V_i имеет J возможных значений $A_i^{(j)}$, являющиеся нечеткими множествами, определяемыми функциями принадлежности $\mu_{A_i^{(j)}}$ на универсальных множествах $X_i, i = \overline{1, n}$. С учетом этих обозначений база знаний, состоящая из набора правил, получаемых из исходной базы прецедентов, формируется в соответствии со следующим алгоритмом.

Алгоритм генерирования нечетких правил из базы прецедентов

Этап 1. Задается число J и тип функций принадлежности, определенных для каждой лингвистической переменной V_i , соответствующей признаку x_i . Универсальное множество X_i для переменной V_i определяется динамически как интервал между минимальным и максимальным значением данного признака на обучающем наборе данных.

Этап 2. Для каждого прецедента вида (1) из обучающей выборки определяется нечеткое правило *Rule* следующего вида:

$$IF V_1 = F(x_1) \& V_2 = F(x_2) \& \dots \& V_n = F(x_n) THEN R = d \quad (3.4)$$

где R - переменная, номинальными значениями которой являются имена классов, $F(x_i)$ является нечетким значением (нечетким множеством), которое принимает лингвистическая переменная V_i в результате фаззификации значений признаков x_i :

$$F(x_i) = A_i^{(j^*)}, \text{ где } j^* = \arg \max_{j=1, J} \mu_{A_i^{(j)}}(x_i) \quad (3.5)$$

Этап 3. Вычисляется степень истинности $TD(Rule)$ для каждого правила

$$TD(Rule) = \min_{i=1, n} \mu_{A_i^{(j^*)}}(x_i) \quad (3.6)$$

Этап 4. Проводится разрешение конфликтов между правилами. После этапа 2 мы имеем набор правил, однозначно соответствующих множеству прецедентов. Однако этот набор может содержать подмножества правил с одинаковыми предпосылками. Если для всех правил в такой подгруппе мы имеем одно и то же заключение, дубликаты правила просто удаляются из множества. Для правил с противоречивыми выводами, предлагаются две различные стратегии для разрешения конфликта. Пусть имеется подмножество m правил $Rule_1, Rule_2, \dots, Rule_m$ вида (2) с одинаковыми предпосылками, но различными заключениями. Разделим набор индексов $I = \{1, 2, \dots, m\}$ на K

подмножеств I_1, I_2, \dots, I_K , соответствующих классам решений d_1, d_2, \dots, d_K , в заключениях правил, $\dim I_k = m_k, \sum_{k=1}^K m_k = m$.

Первая стратегия разрешения конфликтов - получение оптимального правила с индексом.

$$l^* = \arg \max_{l \in I} TD(Rule_l) \quad (3.7)$$

и выбор решения d , соответствующего заключительной части оптимального правила.

Вторая стратегия разрешения конфликтов учитывает не только максимальную степень истинности правила, но и количество правил для каждого заключения. Оптимальное решение d_k^* , назначаемое конфликтующему набору правил, может быть получено следующим образом

$$k^* = \arg \max_k \left(\sum_{i \in I_k} TD(Rule_i) \right) \quad (3.8)$$

Исследования алгоритма будем проводить на известных эталонных наборах данных: «Ирис», «Уровень знаний студентов». Набор данных «Ирис» содержит 150 примеров, и характеризуется четырьмя атрибутами: длина чашелистиков в см, ширина чашелистиков в см, длина лепестков в см, ширина лепестков в см. В данный набор входят три класса, в каждом из которых содержится по 50 прецедентов. Классообразующим атрибутом прецедента является класс ириса: Setosa, Versicolour или Virginica. Набор данных «Уровень знаний студентов» содержит 119 примеров, и характеризуется четырьмя атрибутами: SCG (показатель повторений материала студентами), STR (уровень затрат времени на обучение); LPR (уровень понимания связи предмета и цели на экзамене), UNS (уровень знаний студента). В данный набор входят три класса, в каждом из которых содержится разное количество прецедентов. Классообразующим атрибутом прецедента является класс, к которому относится уровень знаний студентов: Low, Middle, High [1].

На первом этапе нужно задать количество и тип функции принадлежности. Для определения и представления нечёткого множества используются функции принадлежности нескольких классов: кусочно-линейные, z – образные, s – образные, п – образные, которые, в свою очередь, имеют различные формы и могут быть заданы следующими аналитическими выражениями.

Треугольная функция. Треугольная функция принадлежности в общем виде задаётся аналитическим выражением вида:

$$\mu_{\Delta}(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{b-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (3.9)$$

где a, b, c – некоторые числовые параметры, упорядоченные отношением $a \leq b \leq c$ и принимающие произвольные действительные значения. Функция принадлежности порождает нормальное выпуклое унимодальное нечёткое множество с носителем – (a, c) , границами $(a, c) / \{b\}$, ядром $\{b\}$ и модой b .

Трапецевидная функция. Трапецевидная функция принадлежности в общем виде может быть задана аналитически следующим выражением:

$$\mu_T(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (3.10)$$

где a, b, c, d – некоторые числовые параметры, принимающие произвольные действительные значения и упорядоченные отношением $a \leq b \leq c \leq d$. Функция принадлежности порождает нормальное выпуклое нечёткое множество с носителем – интервалом (a, d) , границами $(a, b) \cup (c, d)$, ядром $\{b, c\}$.

Z – образная функция. График функции представлен z – образной кривой, функция также называется сплайн – функцией и в общем случае аналитически задаётся следующим выражением:

$$\mu_{z_1}(x; a, b) = \begin{cases} 1, & x < a \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-a}{b-a} \pi\right), & a \leq x \leq b \\ 0, & x > b \end{cases} \quad (3.11)$$

где a, b – некоторые числовые параметры, принимающие произвольные действительные значения и упорядоченные отношением: $a < b$. Сплайн – функция также может быть задана аналитическим выражением вида:

$$\mu_{z_2}(x; a, b) = \begin{cases} 1, & x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2} \\ 2\left(\frac{b-x}{b-a}\right)^2, & \frac{a+b}{2} < x < b \\ 0, & b \leq x \end{cases} \quad (3.12)$$

где a, b – некоторые числовые параметры, принимающие произвольные действительные значения и упорядоченные отношением: $a < b$.

S – образная функция. S – образная функция принадлежности в общем виде может быть задана аналитически следующим выражением:

$$\mu_{s_1}(x; a, b) = \begin{cases} 0, & x < a \\ \frac{1}{2} + \frac{1}{2} \cos\left(\frac{x-b}{b-a} \pi\right), & a \leq x \leq b \\ 1, & x > b \end{cases} \quad (3.13)$$

где a, b – некоторые числовые параметры, принимающие произвольные действительные значения и упорядоченные отношением: $a < b$. Сплайн – функция может быть задана аналитическим выражением вида:

$$\mu_{S_2}(x; a, b) = \begin{cases} 0, & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2} \\ 1 - 2\left(\frac{b-x}{b-a}\right)^2, & \frac{a+b}{2} < x < b \\ 1, & b \leq x \end{cases} \quad (3.14)$$

где a, b – некоторые числовые параметры, принимающие произвольные действительные значения и упорядоченные отношением: $a < b$. Данные функции принадлежности порождают нормальные выпуклые нечёткие множества с ядром $[b, +\infty)$ и носителем $(a, +\infty)$

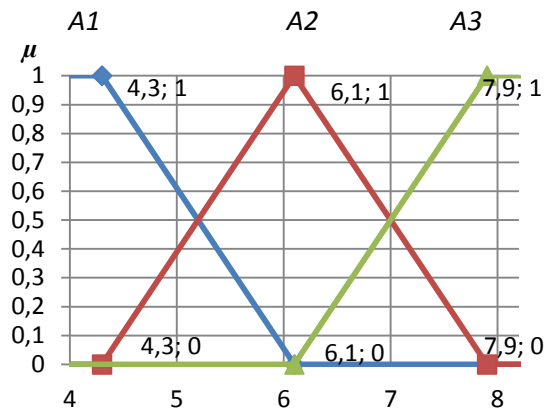
Π – образная функция. Π – образная функция принадлежности характеризуется формой кривой, представляющей сглаженную трапецию и в общем случае, задается аналитически следующим выражением:

$$\mu_{\Pi}(x; a, b, c, d) = f_S(x; a, b) * f_Z(x; c, d) \quad (3.15)$$

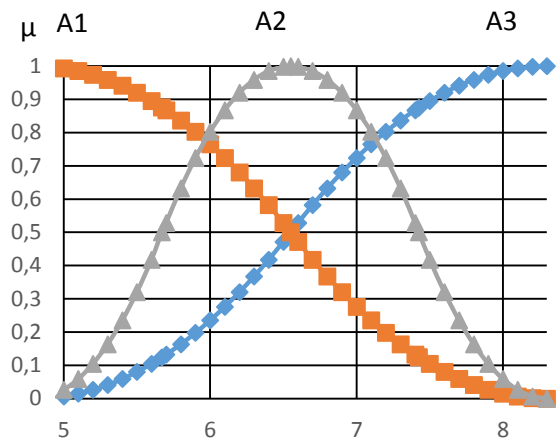
где a, b, c, d – некоторые числовые параметры, принимающие произвольные действительные значения и упорядоченные отношением $a \leq b \leq c \leq d$, а знак «*» обозначает арифметическое произведение значений соответствующих функций. Этот тип функций принадлежности порождает нормальные выпуклые нечёткие множества с носителем (a, d) , и ядром $[b, c]$.

В исследовании алгоритма будем использовать треугольную и S-образную функции принадлежности. Загруженные данные (набор прецедентов) группируем в классы, подсчитываем количество прецедентов в каждом классе и выводим их количество пользователю. Отбор прецедентов в обучающую выборку происходит случайным образом. Можно самостоятельно задать количество прецедентов для обучающей выборки. Из исходной выборки отбираем прецеденты в обучающую выборку, прецеденты, которые не попали в обучающую выборку, формируют набор прецедентов, из которых система выбирает заданное количество прецедентов для контрольной выборки.

Универсальное множество для каждой лингвистической переменной определяется динамически. В качестве универсального множества используем отрезок от минимального до максимального значения. Для задания терм-значений лингвистической переменной каждый полученный интервал (универсум) разделим на количество функций принадлежности равное $(2N+1)$, $N=1, 2, 3$, т.е. 3, 5 и 7. Для лингвистической переменной каждое терм-значение обозначим следующим образом: A_1 (Малый 1), ..., A_n (Большой n), B_1 (Малый 1), ..., B_n (Большой n), C_1 (Малый 1), ..., C_n (Большой n), D_1 (Малый 1), ..., D_n (Большой n) и для каждого из них определяем функцию принадлежности. На рис. 3.6а представлен пример такого разделения, где область определения атрибута разделена на 3 ФП ($N=1$) и функции принадлежности имеет тип Т, а на рис.3.6б представлен пример ФП для типа S.



а)

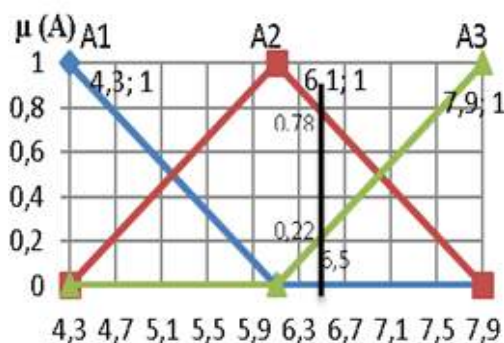


б)

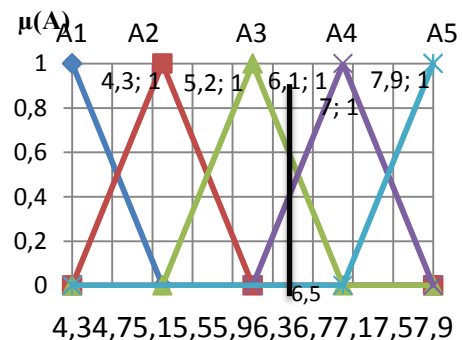
Рисунок 3.6. Функции принадлежности а) типа Т; б) типа S

Определим минимальное и максимальное значение для каждого атрибута (свойства) на обучающей выборке. Например, длина чашелистиков может варьироваться от 4,3 см до 7,9 см. Рассмотрим первый элемент данных, который попал в обучающую выборку: {6,5;2,8;4,6;1,5; Iris-versicolor}. Определим минимальное и максимальное значение для каждого атрибута в обучающей выборке, т.е. для атрибута А: минимальное значение 4,3, а максимальное – 7,9; для атрибута В: минимальное значение 2,0, а максимальное – 4,4; для атрибута С: минимальное значение 1,0, а максимальное – 6,9; для атрибута D: минимальное значение 0,1, а максимальное – 2,5.

На рис. 3.7 представлены функции принадлежности для двух атрибутов А и В. Пример такого разделения, где область определения атрибута А разделена на 3 ФП ($N=1$) представлены на рис. 3.7а, на рис. 3.7б – на 5 ФП, на рис. 3.7в – на 7 ФП, для атрибута В на рис.3.7г представлено разделение на 3 ФП, на рис. 3.7д – на 5 ФП, на рис. 3.7е – на 7 ФП.



а)



б)

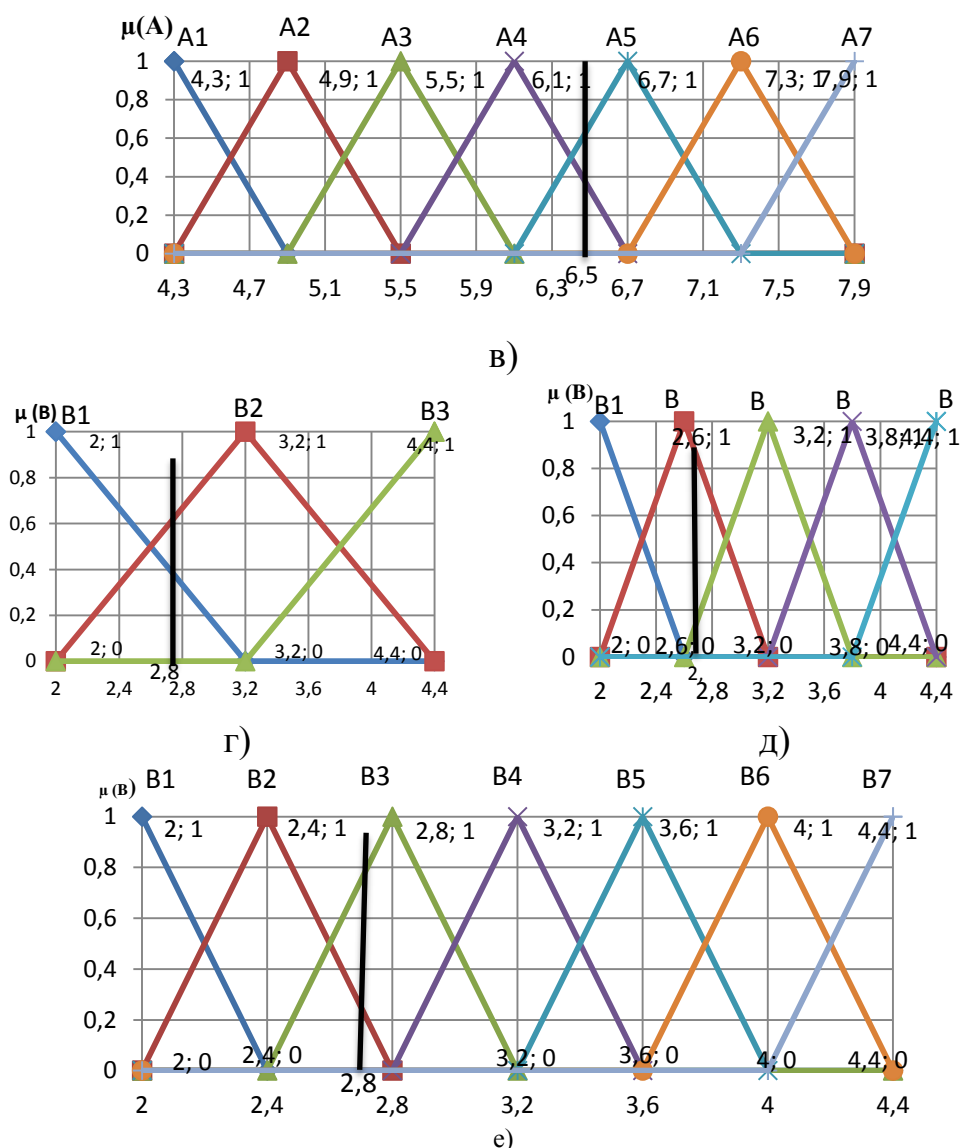


Рисунок 3.7. Функции принадлежности для атрибута А а) 3 ФП; б) 5 ФП; в) 7 ФП; для атрибута В г) 3 ФП, д) 5 ФП, е) 7 ФП и типа Т

На втором этапе необходимо сформировать нечеткие правила из множества прецедентов на основе обучающей выборки. Метод формирования правил из множества прецедентов на основе обучающей выборки использует преобразование выборки прецедентов в систему нечетких правил. Идея метода заключается в следующем. Множества определения признаков $X_1, X_2, X_3, \dots, X_n$ рассматриваются как универсальные множества, на которых определяются нечеткие терм-множества. Рассмотрим формирование правила для каждой лингвистической переменной V_i , соответствующей признаку X_i для этого зададим по 3 терма A_i, B_i, C_i («низкий», «средний» и

«высокий»), определяемые треугольными функциями принадлежности $\mu_{A_i}, \mu_{B_i}, \mu_{C_i}$. Тогда правило, соответствующее прецеденту, можно сформировать в соответствии с формулой 3.4.

Определим степень принадлежности обучающих данных для каждой ФП. Для примера рассмотрим атрибут А и функцию принадлежности **типа Т**. Выберем первое значение из элемента данных {6,5;2,8;4,6;1,5; Iris-versicolor}, т.е. $x_1=6,5$ для признака А и **трех ФП** (см. рис.3.7а). Значению $x_1=6,5$ соответствует значение функции принадлежности A_1 , равное 0, значение функции принадлежности A_2 , равное 0,78, значение функции принадлежности A_3 , равное 0,22. Максимальное значение равно 0,78 ($\max \{0;0,78;0,22\} = 0,78$) и принадлежит функции принадлежности A_2 , поэтому включается в предусловие нечеткого правила, как $x_1 [\max = 0,78 \text{ в } A_2]$, т.е. $x_1 = A_2$. Выберем второе значение из элемента данных {6,5;2,8;4,6;1,5; Iris-versicolor}, т.е. $x_2=2,8$ для атрибута В и **трех ФП** (см. рис. 3.7г). Значению $x_2=2,8$ соответствует значение функции принадлежности B_1 , равное 0,39, значение функции принадлежности B_2 , равное 0,61, значение функции принадлежности B_3 , равное 0. Максимальное значение равно 0,61 ($\max \{0,39;0,61;0\} = 0,61$) и принадлежит функции принадлежности B_2 , поэтому включается в предусловие нечеткого правила, как $x_2 [\max = 0,61 \text{ в } B_2]$, т.е. $x_2 = B_2$.

Определим степень принадлежности обучающих данных для каждой ФП. Для примера рассмотрим атрибут К и функцию принадлежности **типа S**. Выберем первое значение из элемента данных {6,5;2,8;4,6;1,5; Iris-versicolor}, т.е. $x_1=6,5$ для признака К и **трех ФП** (см. рис 3.8). Значению $x_1=6,5$ соответствует значение функции принадлежности K_1 , равное 0,68, значение функции принадлежности K_2 , равное 0,32, значение функции принадлежности K_3 , равное 0,92. Максимальное значение равно 0,92 ($\max \{0,68;0,32;0,92\} = 0,92$) и принадлежит функции принадлежности K_3 , поэтому включается в предусловие нечеткого правила, как $x_1 [\max = 0,92 \text{ в } K_3]$, т.е. $x_1 = K_3$.

Формируя предусловия для x_2 , x_3 и x_4 аналогичным образом, получаем следующие правила. Окончательно для типа Т можно записать правило:

$\{x_1, x_2, x_3, x_4; Z\} \Rightarrow$
 $\{x_1[\max = 0,78 \text{ в } A_2], x_2[\max = 0,61 \text{ в } B_2], x_3[\max = 0,45 \text{ в } C_2], x_4[\max = 0,85 \text{ в } D_1];$
Iris – Setosa

Правило для типа Т: $IF(x_1 = A_2, x_2 = B_2, x_3 = C_2, x_4 = D_1) Then Iris – Setosa$

Аналогично для типа S можно записать правило:

$\{x_1, x_2, x_3, x_4; Z\} \Rightarrow$
 $\{x_1[\max 0,92 \text{ в } K_3], x_2[\max 0,75 \text{ в } K_1], x_3[\max 0,46 \text{ в } K_2], x_4[\max 0,83 \text{ в } K_3];$
Iris – Setosa

Правило для типа S: $IF(x_1 = K_3, x_2 = K_1, x_3 = K_2, x_4 = K_3) Then Iris – Setosa$

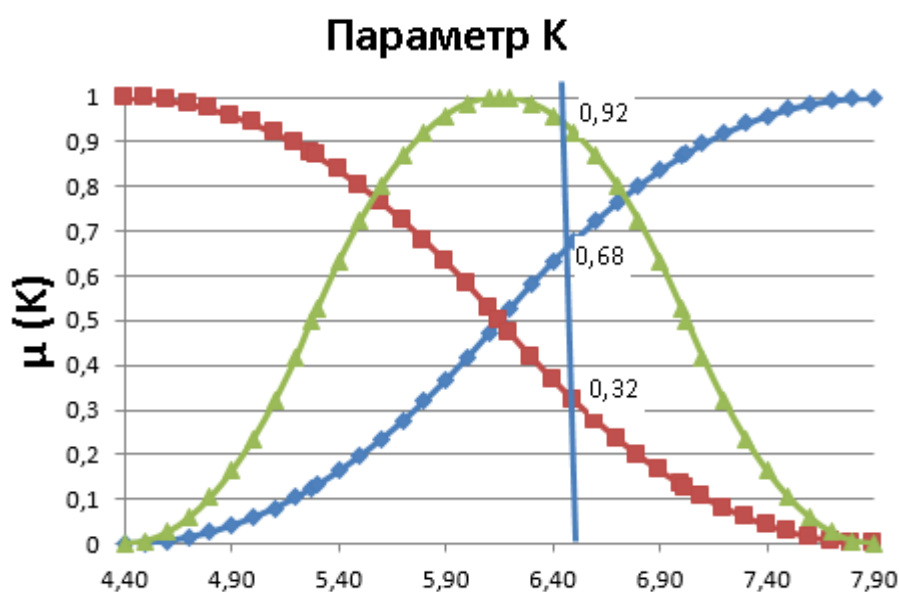


Рисунок 3.8. Функции принадлежности для атрибута К 3 ФП и типа S

Рассмотрим атрибут А для **пяти ФП**. Выберем первое значение из элемента данных {6,5;2,8;4,6;1,5; Iris-versicolor}, т.е. $x_1=6,5$ для признака А и пяти ФП (см. Рис 3.7б). Значению $x_1=6,5$ соответствует значение функции принадлежности A_1 , равное 0, значение функции принадлежности A_2 , равное 0, значение функции принадлежности A_3 , равное 0,58, значение функции принадлежности A_4 , равное 0,4, значение функции принадлежности A_5 , равное 0. Максимальное значение равно 0,58 ($\max \{0;0;0,58;0,4;0\}=0,58$) и принадлежит функции принадлежности A_3 , поэтому включается в предусловие нечеткого правила, как $x_1[\max = 0,58 \text{ в } A_3]$, т.е. $x_1 = A_3$. Выберем второе значение из элемента данных {6,5;2,8;4,6;1,5; Iris-versicolor}, т.е. $x_2=2,8$ для

признака В и **пяти ФП** (см. рис 3.7д). Значению $x_2=2,8$ соответствует значение функции принадлежности B_1 , равное 0, значение функции принадлежности B_2 , равное 0,78, значение функции принадлежности B_3 , равное 0,22, значение функции принадлежности B_4 , равное 0, значение функции принадлежности B_5 , равное 0. Максимальное значение равно 0,78 ($\max \{0;0,78;0,22;0;0\} = 0,78$) и принадлежит функции принадлежности B_2 , поэтому включается в предусловие нечеткого правила, как $x_2 [\max = 0,78 \text{ в } B_2]$, т.е. $x_2 = B_2$. Формируя предусловия для x_3 и x_4 аналогичным образом, получаем следующие правила:

$$\{x_1, x_2, x_3, x_4; Z\} \Rightarrow$$

$$\{x_1 [\max = 0,58 \text{ в } A_3], x_2 [\max = 0,78 \text{ в } B_2], x_3 [\max = 0,75 \text{ в } C_5], x_4 [\max = 0,9 \text{ в } D_3];$$

Iris - virginica

Правило: IF ($x_1 = A_3, x_2 = B_2, x_3 = C_5, x_4 = D_3$) Then Iris - virginica

Рассмотрим атрибут А для **семи ФП**. Выберем первое значение из элемента данных $\{6,5;2,8;4,6;1,5; \text{Iris-versicolor}\}$, т.е. $x_1=6,5$ для атрибута А и семи ФП (см. рис 3.7в). Значению $x_1=6,5$ соответствует значение функции принадлежности A_1 , равное 0, значение функции принадлежности A_2 , равное 0, значение функции принадлежности A_3 , равное 0, значение функции принадлежности A_4 , равное 0,38, значение функции принадлежности A_5 , равное 0,62, значение функции принадлежности A_6 , равное 0, значение функции принадлежности A_7 , равное 0. Максимальное значение равно 0,62 ($\max \{0;0;0;0,38;0,62;0;0\} = 0,62$) и принадлежит функции принадлежности A_5 , поэтому включается в предусловие нечеткого правила, как $x_1 [\max = 0,62 \text{ в } A_5]$, т.е. $x_1 = A_5$. Выберем второе значение из элемента данных $\{6,5;2,8;4,6;1,5; \text{Iris-versicolor}\}$, т.е. $x_2=2,8$ для атрибута В и **семи ФП** (см. рис 3.7е). Значению $x_2=2,8$ соответствует значение функции принадлежности B_1 , равное 0, значение функции принадлежности B_2 , равное 0,22, значение функции принадлежности B_3 , равное 0,8, значение функции принадлежности B_4 , равное 0, значение функции принадлежности B_5 , равное 0, значение функции принадлежности B_6 , равное 0, значение функции принадлежности B_7 , равное 0. Максимальное значение равно 0,8 ($\max \{0;0,22;0,8;0;0;0;0\} = 0,8$) и принадлежит функции

принадлежности B_3 , поэтому включается в предусловие нечеткого правила, как $x_2 [\max = 0,8 \text{ в } B_3]$, т.е. $x_2 = B_3$. Формируя предусловия для x_3 и x_4 аналогичным образом, получаем следующие правила:

$$\{x_1, x_2, x_3, x_4; Z\} \Rightarrow$$

$$\{x_1 [\max = 0,62 \text{ в } A_5], x_2 [\max = 0,8 \text{ в } B_3], x_3 [\max = 0,5 \text{ в } C_7], x_4 [\max = 0,76 \text{ в } D_5];$$

Iris - versicolor

$$\text{Правило: } IF(x_1 = A_5, x_2 = B_3, x_3 = C_7, x_4 = D_5) \text{ Then } Iris - versicolor$$

На третьем этапе необходимо сопоставление каждому правилу R степень истинности $SP(R)$. При переходе от прецедентов к набору нечетких правил, может возникнуть ситуация, когда множеству правил с одинаковыми предпосылками соответствуют различные заключения, важно сопоставить степень истинности правила, т.о. чтобы у большего количества прецедентов класс был определен верно.

На четвертом этапе необходимо разрешить возникшие конфликты между правилами. Исследуем 2 стратегии по разрешению конфликтов.

Первая стратегия. Для каждого правила введем степень истинности, которая рассчитывается как максимальное значение функций принадлежности для каждого атрибута. Для правила из примера, см. рис.3.7а для трех ФП $SP(R) = \max(0,78, 0,61, 0,45, 0,85) = 0,85$. Для формирования правил используется несколько прецедентов. В процессе формирования нечетких правил, мы можем получить целый набор правил, антецеденты которых совпадают, а консеквенты разные. Такой набор правил называется конфликтным. Таким образом, в модели мы приняли допущение, что если антецеденты правил одинаковые, а консеквенты разные, то разрешить противоречие поможет степень истинности. Из множества конфликтующих правил с одинаковыми условиями выбирается правило с максимальной степенью истинности. Рассмотрим пример:

$$IF(x_1 = A_3, x_2 = B_2, x_3 = C_4, x_4 = D_3) \text{ Then } Z = Iris - versicolor, SP_1(R) = 0,64$$

$$IF(x_1 = A_3, x_2 = B_2, x_3 = C_4, x_4 = D_3) \text{ Then } Z = Iris - virginica, SP_2(R) = 0,5$$

$IF(x_1 = A_3, x_2 = B_2, x_3 = C_4, x_4 = D_3) \text{ Then } Z = \text{Iris - virginica}, SP_3(R) = 0,55$

Разрешение конфликта для этих трех правил приводит к единственному правилу:

$IF(x_1 = A_3, x_2 = B_2, x_3 = C_4, x_4 = D_3) \text{ Then } Z = \text{Iris - versicolor},$

т.к. $SP(R) = \max(0,64, 0,5, 0,55) = 0,64$. Мы выбираем класс Iris-versicolor, соответственно 2 случая у нас будут некорректно сопоставлены. В этом случае, мы не учитываем количество правил с одинаковыми антецедентами.

Вторая стратегия разрешения конфликтов учитывает не только максимальную степень истинности правила, но и количество правил для каждого заключения. Находим общее количество правил с одинаковыми антецедентами, для нашего примера это будет 3. Далее рассчитываем новую степень истинности для выбора класса. В табл. 3.1 представлен механизм расчета степени истинности с учетом количества одинаковых правил для выбора класса.

Таблица 3.1. Выбор класса с учетом количества одинаковых правил

№	Класс	Значение SP(R)	Расчет SP(R) для выбора класса
1	Iris-versicolor	0,213	$SP(R) = 1/3 * SP_1 = 1/3 * 0,64 = 0,213$
2	Iris-virginica	0,35	$SP(R) = 1/3 * SP_2 + 1/3 * SP_3 = 1/3 * 0,5 + 1/3 * 0,55 = 0,35$

В общем случае получаем следующую формулу для определения класса:

$$SP^*(R) = \max_{k \in I_{class}} SP_k(R), class = \arg\left\{ \max_{j=1, n} \sum_{i \in I_j} \frac{1}{m} SP_i(R) \right\}, \quad (3.16)$$

где m – количество конфликтующих правил ($i=1, \dots, m$), $SP_i(R)$ – степень истинности i-го правила, n – количество классов, множество I разбивается на j-подмножеств, в соответствии с разбиением конкурирующих правил по классам.

Определив по формуле класс, правилу ставим в соответствие степень истинности $SP^*(R)$. Для примера значение $SP^* = 0,35$ и это будет класс Iris-virginica. В таблицу правил для правила:

$IF(x_1 = A_3, x_2 = B_2, x_3 = C_4, x_4 = D_3)$ установим класс *Iris - virginica* со

степенью истинности 0,55. Таким образом, двум наборам данных будет правильно поставлен класс.

Результатом работы алгоритма будет созданная база нечетких правил.

IF($x_1 = A_2, x_2 = B_2, x_3 = C_2, x_4 = D_1$)*Then Iris - versicolor*

IF($x_1 = A_1, x_2 = B_3, x_3 = C_1, x_4 = D_3$)*Then Iris - virginica*

IF($x_1 = A_2, x_2 = B_1, x_3 = C_3, x_4 = D_1$)*Then Iris - virginica* и т.д.

3.4. Экспериментальные исследования алгоритма классификации прецедентов

В данном разделе будут показаны результаты экспериментальных исследований рассмотренного метода обучения системы нечетких правил на известном эталонном наборе данных (НД): «Ирис» и НД «Уровень знаний студентов» [137].

Разные типы функции принадлежности

Для проведения эксперимента с различными типами функций принадлежности был выбран набор данных «Ирис», из которого случайным образом были сформированы обучающая и контрольная выборки. Нечеткие правила, полученные из обучающей выборки прецедентов были применены к прецедентам из контрольной выборки для их классификации. Компьютерный эксперимент для этого набора данных был проведен 25 раз. Табл. 3.2 показывает точность классификации на контрольной выборке НД «Ирис».

С помощью рассматриваемой модели была достигнута хорошая классификационная точность для двух типов функции принадлежности (см. табл. 3.2 и рис.3.9). Под достаточной классификационной точностью будем понимать уровень точности более 0,800. Среднее значение точности для функций принадлежности типа *S* оказалось выше, чем для типа *T*. Это можно объяснить тем, что функции принадлежности типа *S* более гладкие (не содержат разрывов производных), по сравнению с функциями принадлежности типа *T*.

Таблица 3.2. Тест на точность НД «Ирис» для разных типов ФП

Номер испытания	Исходная выборка	Обучающая выборка	Точность классификации	
			ФП типа T	ФП типа S
1	150	120	0,900	0,970
2	150	120	0,870	0,970
3	150	120	0,930	0,930
4	150	120	0,870	0,970
5	150	120	0,870	0,830
6	150	120	0,970	0,930
7	150	120	0,930	0,890
8	150	120	0,970	0,910
9	150	120	0,970	0,890
10	150	120	0,890	0,970
11	150	120	0,840	0,910
12	150	120	0,820	0,870
13	150	120	0,840	0,910
14	150	120	0,890	0,910
15	150	120	0,800	0,910
16	150	120	0,890	0,890
17	150	120	0,870	0,930
18	150	120	0,870	0,970
19	150	120	0,930	0,910
20	150	120	0,890	0,910
21	150	120	0,870	0,930
22	150	120	0,960	0,960
23	150	120	0,870	0,960
24	150	120	0,910	0,980
25	150	120	0,840	0,910
Среднее	150	120	0,893	0,928

Точность классификации

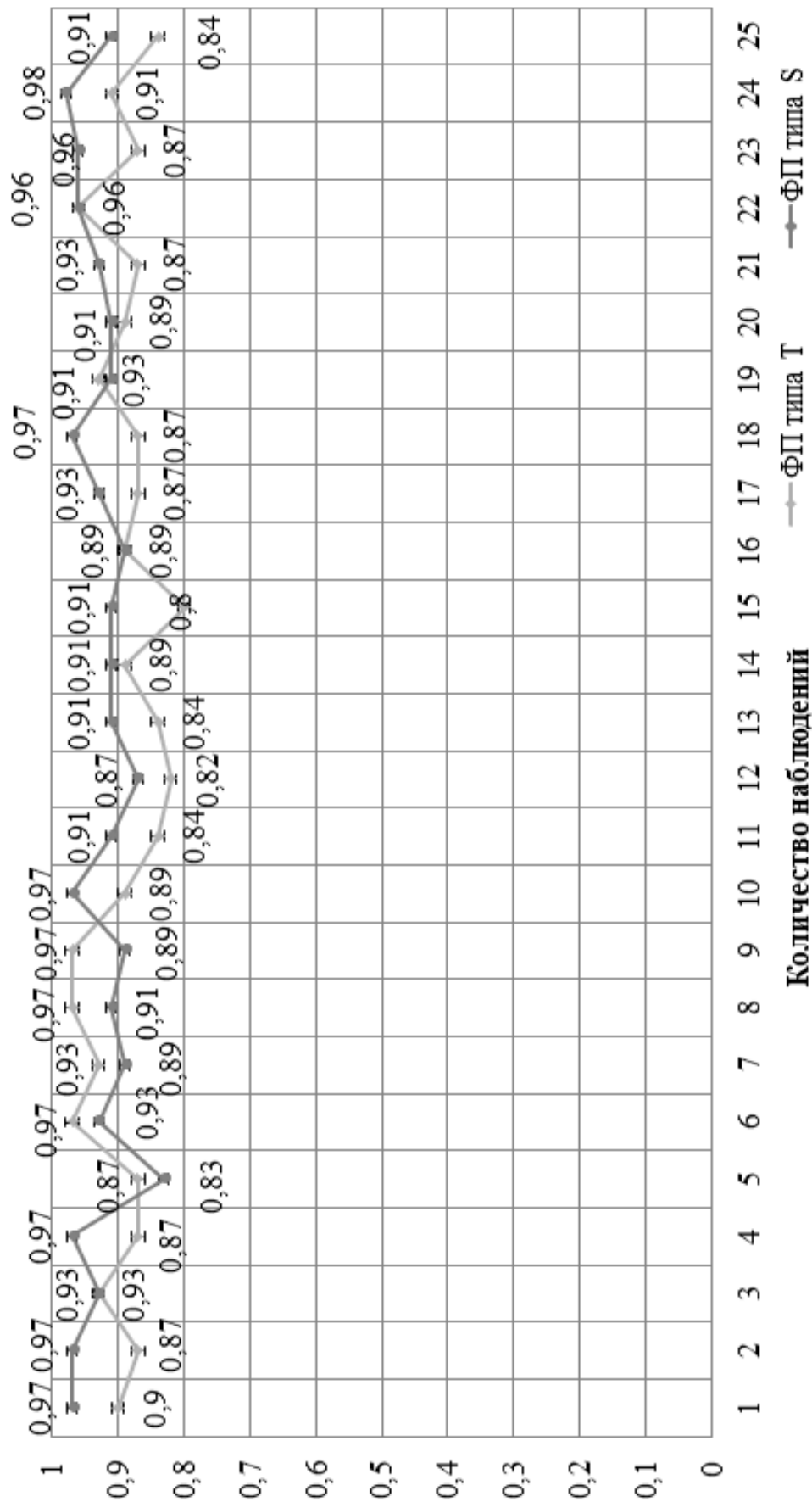


Рисунок 3.9. Тест на точность НД «Ирис»

Различные наборы данных и различное количество функций принадлежности лингвистической переменной

В экспериментах, проводимых над НД «Ирис», случайным образом была сформирована обучающая выборка, а остальные данные попали в контрольную выборку. Нечеткие правила были применены к контрольной выборке. Число экспериментов было 10 для каждого вида испытаний. Для каждого атрибута было определено минимальное и максимальное значение на обучающей выборке. Далее каждый полученный интервал был разделен на 3,5 или 7 ФП. Т.е. количество ФП, например, для атрибута А (длина чашелистика) в правилах изменялось от трех (A_1, A_2, A_3) до семи ($A_1, A_2, A_3, A_4, A_5, A_6, A_7$). Для обучения было выбрано 120 прецедентов. Табл. 3.3 представленная ниже, показывает точность классификации на контрольной выборке для НД «Ирис». Используем первую стратегию разрешения конфликтов, т.е. если антецеденты правил одинаковые, а консеквенты разные, то решить противоречие поможет максимальная степень истинности.

Таблица 3.3. Тест на точность НД «Ирис» для разного количества ФП

Номер испытания	Точность классификации		
	3 ФП	5 ФП	7 ФП
1	0,90	0,83	0,83
2	0,87	0,80	0,63
3	0,97	0,77	0,73
4	0,80	0,63	0,63
5	0,83	0,73	0,57
6	0,87	0,83	0,60
7	0,80	0,70	0,63
8	0,90	0,77	0,70
9	0,93	0,83	0,73
10	0,87	0,67	0,80
Среднее	0,874	0,756	0,685

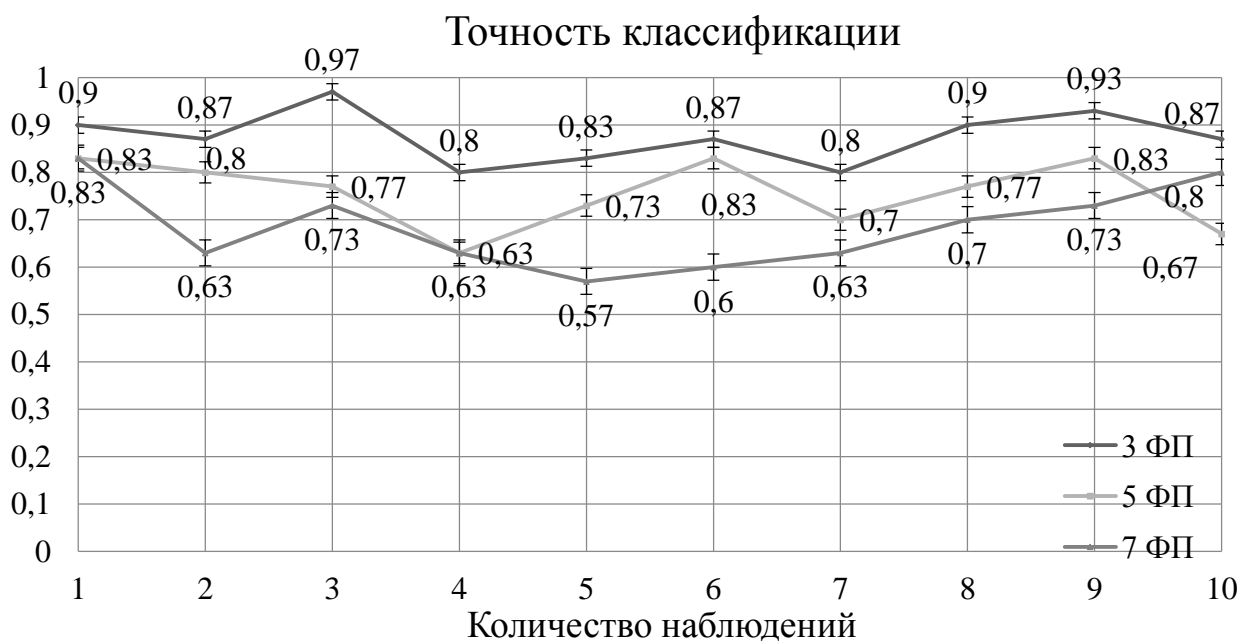


Рисунок 3.10. Тест на точность НД «Ирис»

С помощью предложенной модели была достигнута хорошая классификационная точность для 3 ФП, при увеличении числа ФП классификационная точность снижается, т.к. количество сочетаний становится больше и не все данные попадают в обучающую выборку. Из этого можно сделать вывод, что наибольшую классификационную точность, можно получить, используя только 3 ФП на данном НД.

Аналогичные эксперименты были проведены для набора данных «Уровень знаний студентов». Для обучения было выбрано 90 прецедентов. Табл. 3.4 представленная ниже, показывает точность классификации на контрольной выборке для этого НД.

Таблица 3.4. Тест на точность НД «Уровень знаний студентов»

Номер испытания	Точность классификации		
	3 ФП	5 ФП	7 ФП
1	0,76	0,65	0,59
2	0,72	0,62	0,59
3	0,72	0,62	0,55
4	0,76	0,59	0,52
5	0,72	0,65	0,55
6	0,68	0,65	0,55
7	0,76	0,59	0,52
8	0,70	0,68	0,59
9	0,68	0,62	0,55
10	0,76	0,59	0,52
Среднее	0,73	0,63	0,55

Точность классификации

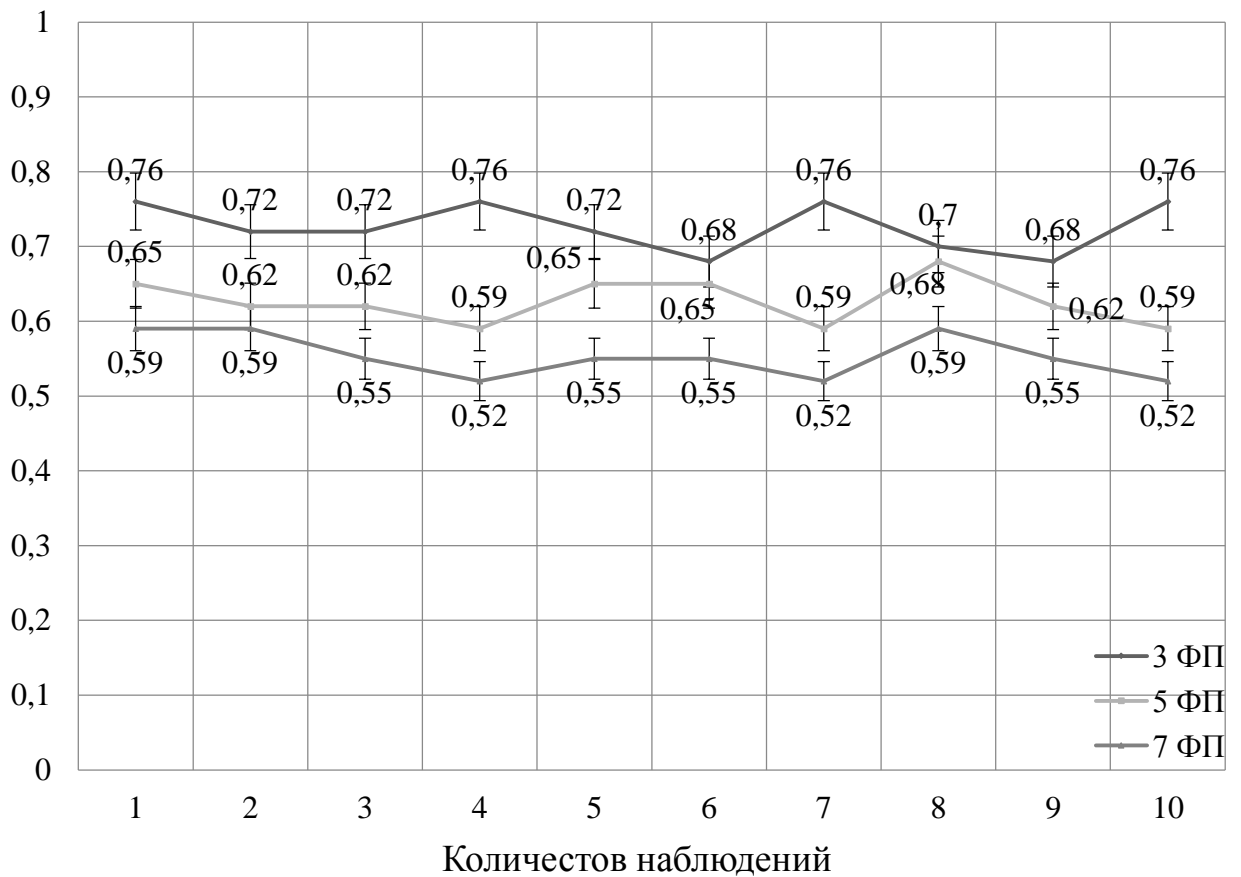


Рисунок 3.11. Тест на точность НД «Уровень знаний студентов»

С помощью предложенной модели была достигнута нормальная классификационная точность для 3 ФП, при увеличении числа ФП классификационная точность снижается, т.к. количество сочетаний становится больше и не все данные попадают в обучающую выборку при случайном отборе. Из этого можно сделать вывод, что наибольшую классификационную точность, можно получить, используя только 3 ФП на данном НД.

Разное количество прецедентов в обучающей выборке

Интересное явление наблюдалось в экспериментах, в которых после обучения нечетких правил, оригинальная база прецедентов могла быть уменьшена до малых размеров, и при этом предоставляла высокую точность классификации для прецедентной системы. Подтверждение результатов можно увидеть в приложении Б таблицах Б.5- Б.10. Исходный размер базы прецедентов для всех испытаний 150.

Точность классификации

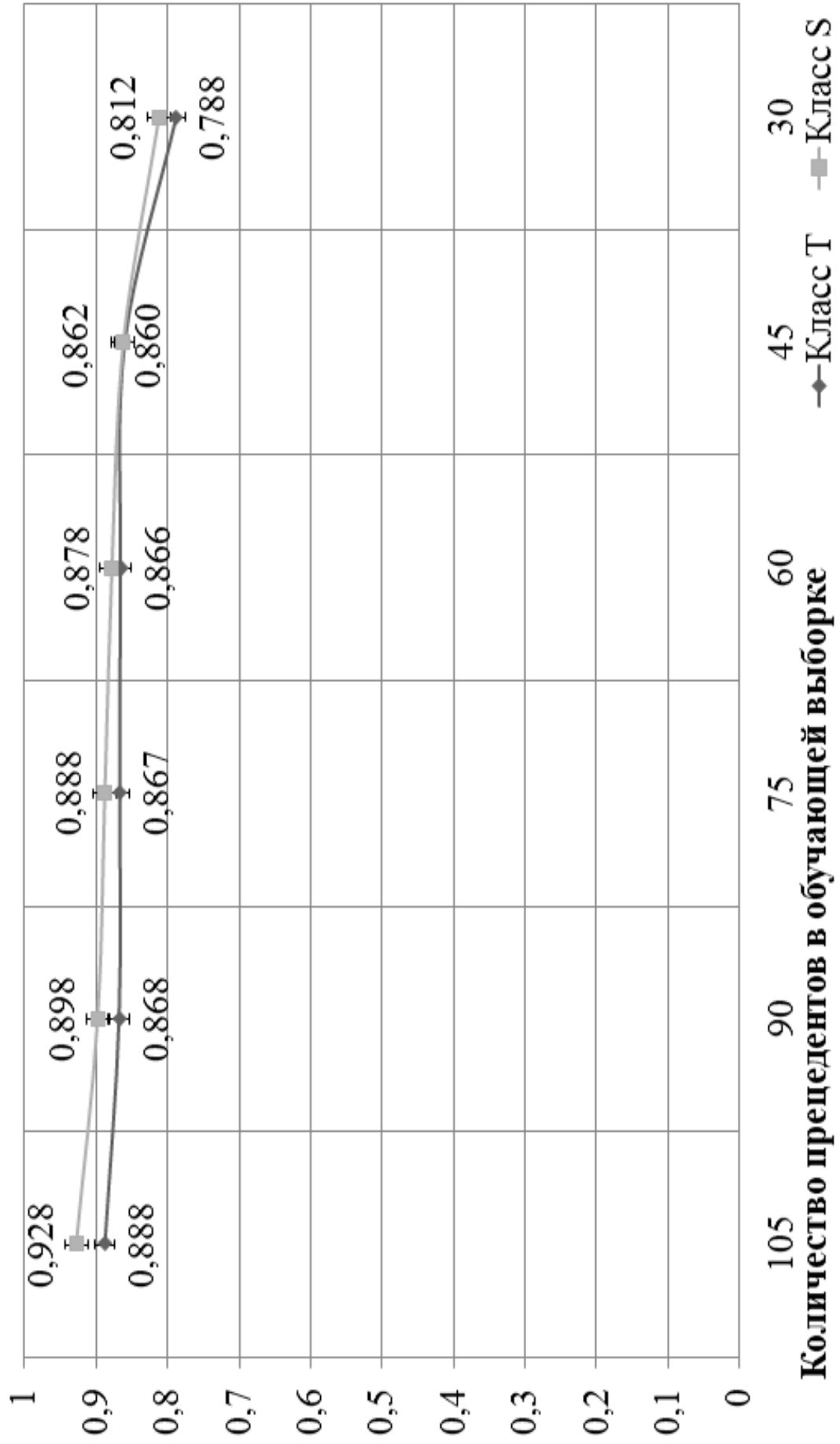


Рисунок 3.12. Точность на НД «Ирис»

Из таблиц Б.5-Б.10 и рис. 3.12 видно, что при значительном сокращении обучающей выборки (до 30 прецедентов из 150) точность поиска остается достаточно высокой, это подтверждает гипотезу о том, что при небольшом количестве исходных данных с помощью предложенного алгоритма можно делать прогнозы с достаточным уровнем точности.

В табл. 3.5 представленной ниже, показана точность классификации на НД «Ирис» с использованием разных стратегий разрешения конфликтов. Вторая стратегия показывает лучшие результаты. Для обучения было выбрано 90 прецедентов [114,156]

Таблица 3.5. Тест на точность с изменением модели для НД «Ирис»

Номер испытания	Точность классификации 3 ФП		Точность классификации 5 ФП	
	Первая стратегия разрешения конфликтов	Вторая стратегия разрешения конфликтов	Первая стратегия разрешения конфликтов	Вторая стратегия разрешения конфликтов
1	0,80	0,90	0,83	0,86
2	0,83	0,90	0,70	0,79
3	0,90	0,87	0,73	0,83
4	0,80	0,87	0,93	0,93
5	0,87	0,93	0,83	0,93
6	0,83	0,93	0,86	0,79
7	0,93	0,90	0,80	0,87
8	0,87	0,93	0,76	0,80
9	0,87	0,83	0,76	0,83
10	0,90	0,87	0,83	0,86
Среднее	0,868	0,893	0,803	0,849

Эффективность модели с использованием второй стратегии разрешения конфликтов и при сокращении прецедентов в ОВ

Проведем эксперименты, в которых попробуем сократить количество прецедентов в ОВ. На основании предыдущих испытания мы выяснили, что оптимальное число функций принадлежности равно 3. Значит, будем использовать 3 ФП, и возьмем модель с использованием второй стратегии разрешения конфликтов, т.к. она показывает лучшие результаты. Результаты экспериментов представлены на рис. 3.13.

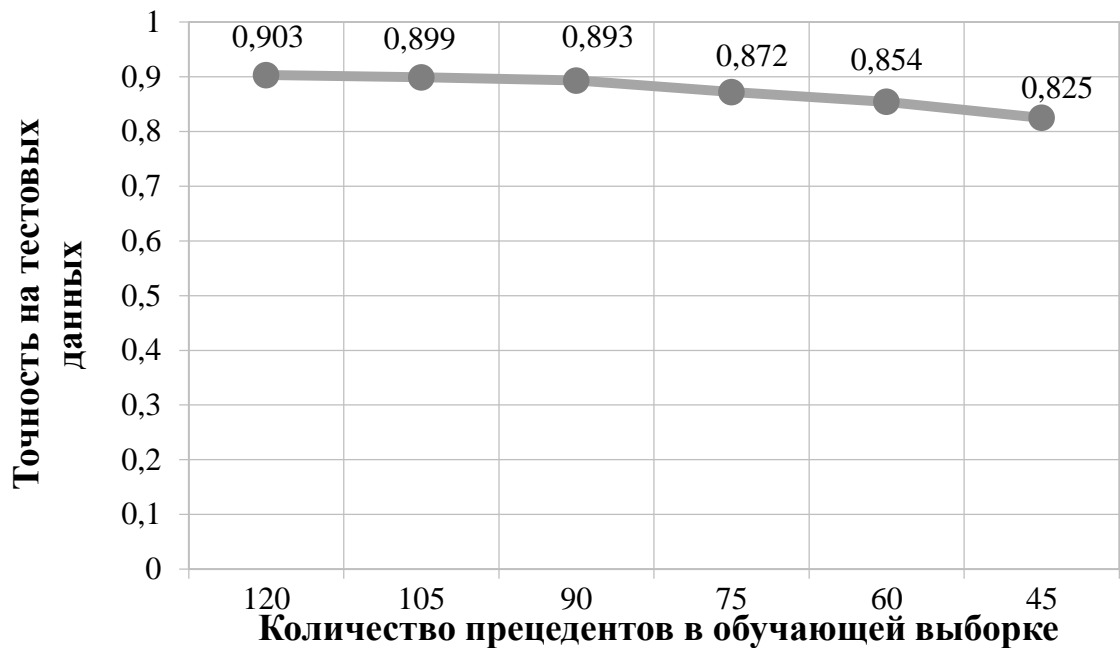


Рисунок 3.13. Точность на НД «Ирис» при сокращении ОВ

На рис. 3.13 видно, что при значительном сокращении тестовых данных (до 45 прецедентов из 150) точность остается достаточно высокой, это подтверждает гипотезу о том, что при небольшом количестве исходных данных с помощью предложенного алгоритма можно делать прогнозы с достаточным уровнем точности. Экспериментальные исследования, которые не вошли в главу 3, можно посмотреть в приложении Б (Табл.Б.1-Б.4).

3.5. Выводы

Основными результатами главы 3 являются:

1. Предложен подход к управлению знаниями в ИТ-подразделениях на основе гибридной модели представления знаний, интегрирующей прецедентный подход с моделью представления знаний в виде нечетких правил. Предлагаемый подход поддерживает динамическую трансформацию знаний из неявной формы (в виде прецедентов) в явную форму (в виде правил принятия решений) и обратно. Разработан механизм интеграции нечеткого логического вывода с механизмом извлечения прецедентов из базы знаний.

2. Разработан и исследован алгоритм классификации прецедентов на основе генерирования нечетких правил. Предложен механизм использования данного метода для извлечения прецедентов, семантически близких к решаемой проблеме пользователя.

3. Исследования эффективности предложенного алгоритма с различными параметрами и на различных наборах данных показали высокую точность классификации на контрольной выборке, которая составляла 25% от общего объема. Исследования, проводимые при значительном сокращении обучающей выборки, показали высокую точность классификации, это подтверждает гипотезу о том, что даже при небольшом количестве исходных данных с помощью предложенного алгоритма можно делать прогнозы с достаточным уровнем точности.

4. Накоплена база нечетких правил, которая представляет знания в явной компактной форме и является хорошо интерпретируемой консультантами и пользователями. Это создает возможность динамического участия человека во взаимодействии с системой, основанной на прецедентах, для адаптации поведения системы к конкретным требованиям.

ГЛАВА 4. РЕАЛИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИИ В ОБЛАСТИ ИТ-КОНСУЛЬТИРОВАНИЯ

В главе описана новая информационная технология поддержки принятия решений, помогающая накапливать, систематизировать, интегрировать и эффективно использовать опыт консультантов. Реализована и внедрена интеллектуальная система поддержки принятия решений на основе использования этой технологии в области ИТ-консультирования. Полученные в этой главе результаты отражены в публикациях [49,55,87,88,115].

4.1. Описание деятельности ИТ-подразделения

Любая организация или подразделение является сложной социально-экономической системой. Рассмотрим ИТ-подразделение, которое занимается внедрением, разработкой и сопровождением информационных систем. Для осуществления своей деятельности имеется главное подразделение, в которое входит три отдела: внедрения, сопровождения и разработки, т.к. они занимаются разными функциями. Организационную структуру ИТ-подразделения представим следующим образом (см. рис. 4.1):

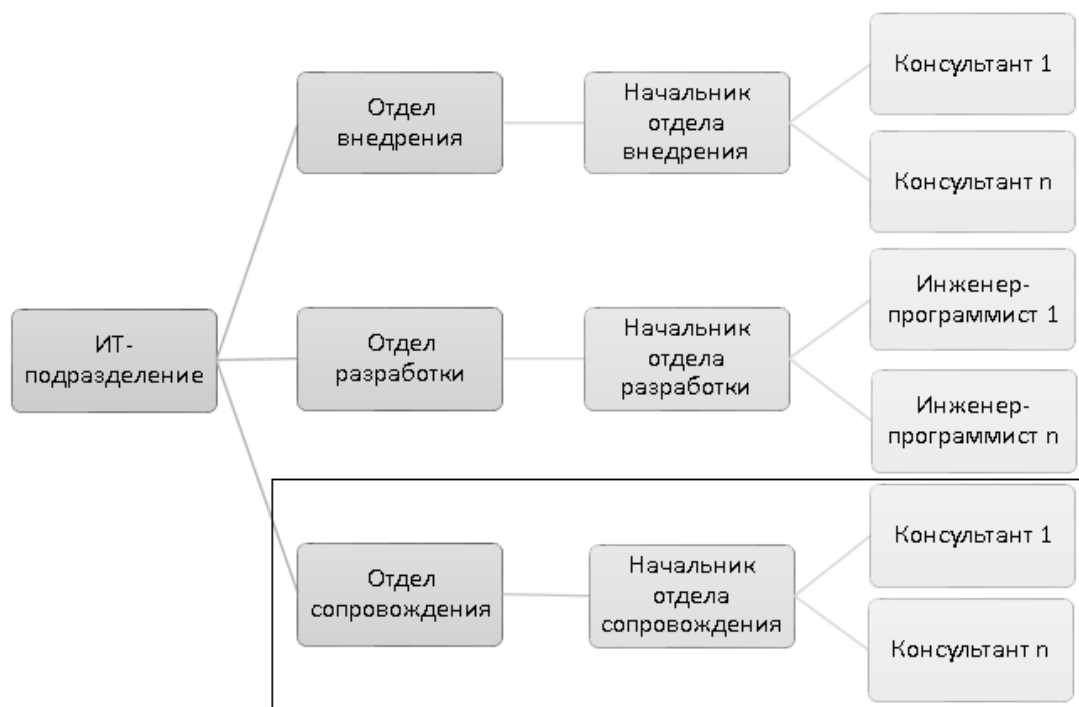


Рисунок 4.1. Организационная структура ИТ-подразделения

Отдел внедрения занимается первичной настройкой нового программного обеспечения (ПО), первичным обучением пользователей работе с новым ПО, анализом данных, формированием файлов для загрузки данных в новое ПО.

Отдел разработки занимается разработкой нового ПО на всех этапах внедрения, доработкой системы в рамках требований пользователей на этапе сопровождения системы, поддержанием системы в актуальном состоянии, оптимизацией процессов работы системы.

Отдел сопровождения занимается поддержкой пользователей, консультированием по функциональным возможностям программы, написанием технических заданий и выполнением критических операций по работе в системе.

Рассмотрим подробнее отдел сопровождения, так как он играет ключевую роль в организации ИТ-поддержки пользователей.

Основными задачами этого отдела являются:

- 1) Обеспечение функционирования и эксплуатации ПО.
- 2) Организация, координация и сопровождение ПО.
- 3) Организация оперативной эксплуатации и сопровождение ПО, находящегося в промышленной эксплуатации.
- 4) Оказание консультаций подразделениям по вопросам, входящим в компетенцию отдела сопровождения информационных систем.
- 5) Организация работы с персоналом в соответствии с требованиями действующих нормативных документов, подготовка и повышение квалификации кадров.

Основными функциями отдела сопровождения являются:

- 1) Консультирование пользователей по работе с системой.
- 2) Разработка пользовательской документации.
- 3) Подготовка необходимой эксплуатационной документации к разработанному программному обеспечению.
- 4) Курирование переданного на предприятия ПО комплексов и задач.
- 5) Разработка инструктивных и методических материалов.

- б) Проведение анализа причин нарушений, в работе внедренных программных систем, разработка предложений по их устранению и предупреждению, повышение качества надежности работы системы.

Отдел сопровождения состоит из консультантов, которые отвечают за свою область учета (бухгалтерский учет, кадровый учет, расчет заработной платы, договорной блок, ведение платежных операций, ведение налогового учета, ведение операций с покупателями и т.п.). В обязанности консультанта-аналитика входит анализ заявок пользователей, написание технических заданий и инструкций пользователям. Консультант является связующим звеном между пользователями информационных систем и программистами, иными словами, переводит требования пользователей на язык программистов.

Консультанты в ИТ-подразделениях пользуются общей технологией, представленной на рис.4.2. Обработка заявок пользователей консультантами чаще всего происходит на основании личного опыта, индивидуального разбора проблемы консультантом, а также поиска необходимой информации в интернете без помощи программных средств.

В рамках создания интеллектуальной системы поддержки принятия решений была построена онтология, с использованием редактора онтологий *Protégé OWL* – программного продукта, разработанного Стэнфордским университетом. Онтология состоит из 71 концепта класса *Payroll*, 82 концептов класса *Accounting* и 11 концептов класса *ContractUnit*.

Онтология была преобразована в файл xml-формата, который при помощи специальной обработки был импортирован в СППР. СППР ИТ консультирования – это новое прикладное решение, разработанное на платформе 1С:Предприятие. Платформа представляет собой программную оболочку над базой данных и содержит в конфигураторе пакет *EnterpriseDevelopmentTools*, базирующийся на основе Eclipse, который позволил реализовать интеллектуальную СППР на основе построенной в *Protégé OWL* онтологии предметной области. В следующем подразделе представим описание новой информационной технологии принятия решений.

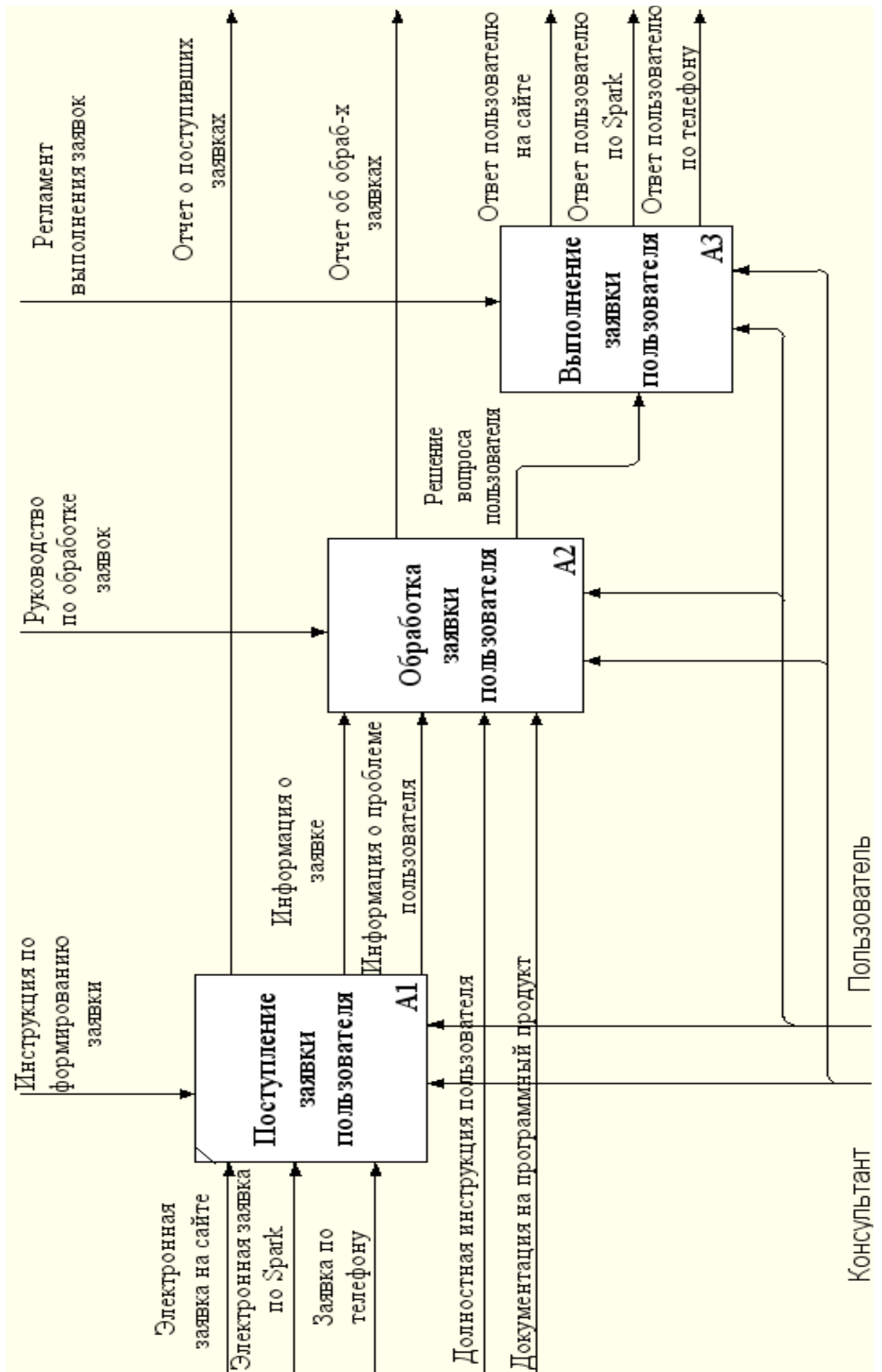


Рисунок 4.2. Типовая технология работы с заявками пользователей

4.2. Новая информационная технология поддержки принятия решений

Новая технология поддержки принятия решений предполагает, что блок «Обработка заявки пользователя» теперь выполняется не на основе личного опыта аналитика, а с использованием последовательности действий, представленных на рис. 4.3.

Для решения проблемы пользователя аналитик использует СППР с базой прецедентов, интегрированных с онтологией предметной области. Все прецеденты разбиты на классы семантически близких прецедентов, причем в системе предусмотрено 3 степени детализации – от укрупненной классификации, содержащий достаточно большое число прецедентов в одном классе, до более детальной, когда класс содержит небольшое число действительно очень близких проблем.

В соответствии с предлагаемым подходом аналитик, получив информацию о проблеме от пользователя, начинает заполнять специальную форму, создавая экземпляр класса Precedent в соответствии с общей структурой (рис.2.22). Затем, в соответствии с полученным описанием проблемы, аналитик связывает данный инцидент с необходимым числом концептов онтологии, после чего вычисляется сила семантических связей с ключевыми словами. Далее с использованием построенных нечетких классифицирующих правил определяется класс прецедентов, семантически наиболее близких к проблеме пользователя. Затем количество полученных прецедентов может быть сокращено с помощью введения дополнительных ограничений, полученных от пользователя. Наконец, консультант анализирует решения из набора семантически близких прецедентов, использует уже готовое решение наиболее сходного прецедента, или объединяет решения из разных прецедентов. Сформировав решение, аналитик передает его пользователю. При необходимости создается новый прецедент в базе знаний.

В следующем подразделе представим реализацию этой технологии в виде интеллектуальной СППР для консультантов ИТ-подразделения.

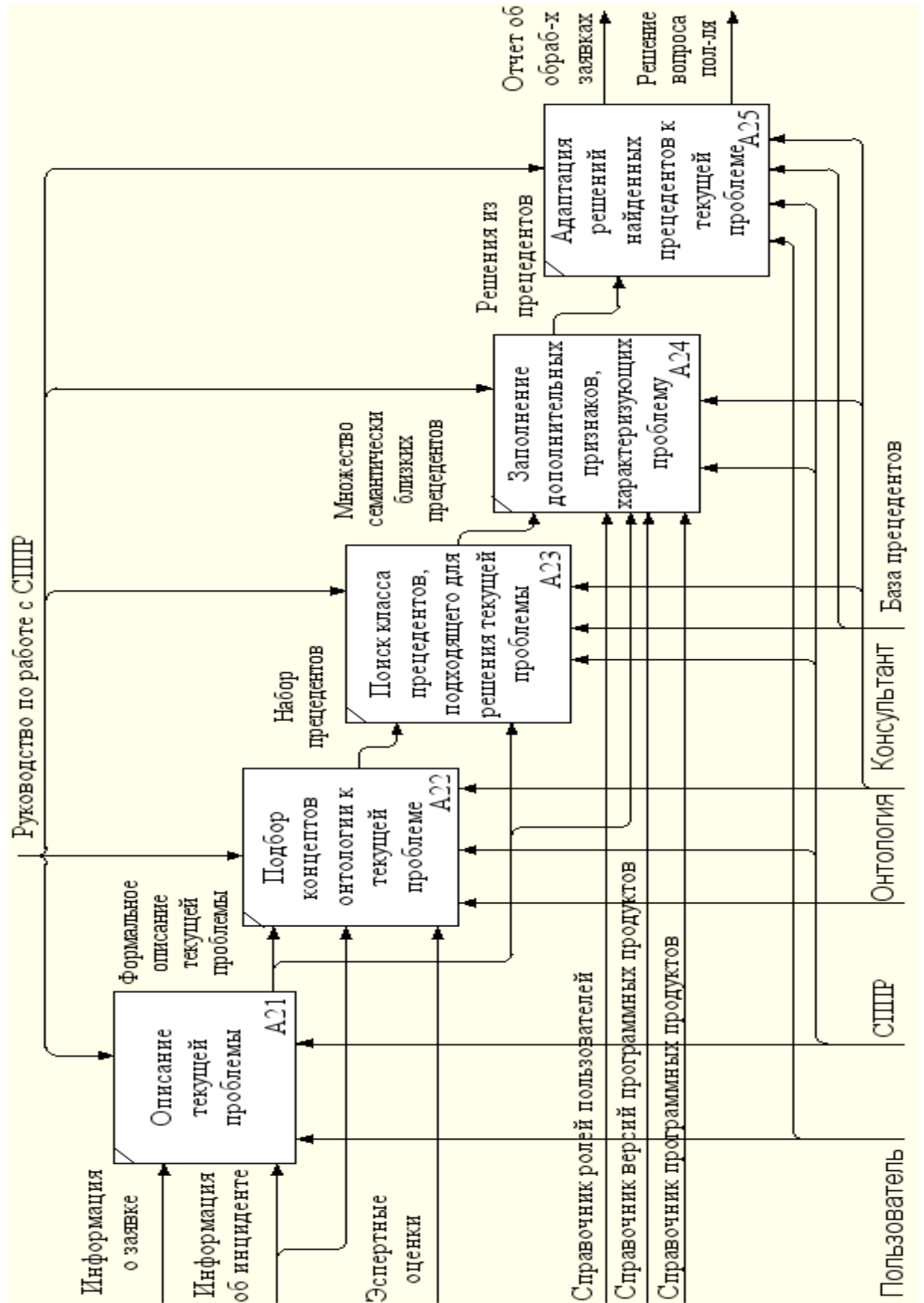


Рисунок 4.3. Новая технология обработки заявки пользователя

4.2. Структура и описание СППР для консультантов ИТ-отдела

С целью оптимизации работы ИТ-подразделения была разработана СППР для консультантов ИТ-отдела [7,26]. Информационная система работает на базе платформы 1С.

Платформа «1С:Предприятие 8» – это предметно-ориентированная среда разработки (также и среда работы с базой) для решения задач, связанных с автоматизацией предприятий. На платформе разрабатываются и запускаются конфигурации. Конфигурация 1С – прикладное решение, созданное с помощью 1С платформы, которое содержит описание структуры таблиц базы данных и алгоритмы работы с этими данными. На основе конфигурации платформа формирует базу данных, т.е. непосредственно создает таблицы и связи между ними согласно описаниям. Пользователь, работающий с системой, создает записи в этих таблицах. В пользовательском режиме платформа выполняет алгоритмы работы с данными [77].

На базе платформы 1С: Предприятие 8 была разработана новая конфигурация интеллектуальной СППР в области ИТ-консультирования. При входе в систему, необходимо выбрать пользователя и пароль, т.к. в настоящее время с системой работает несколько консультантов.

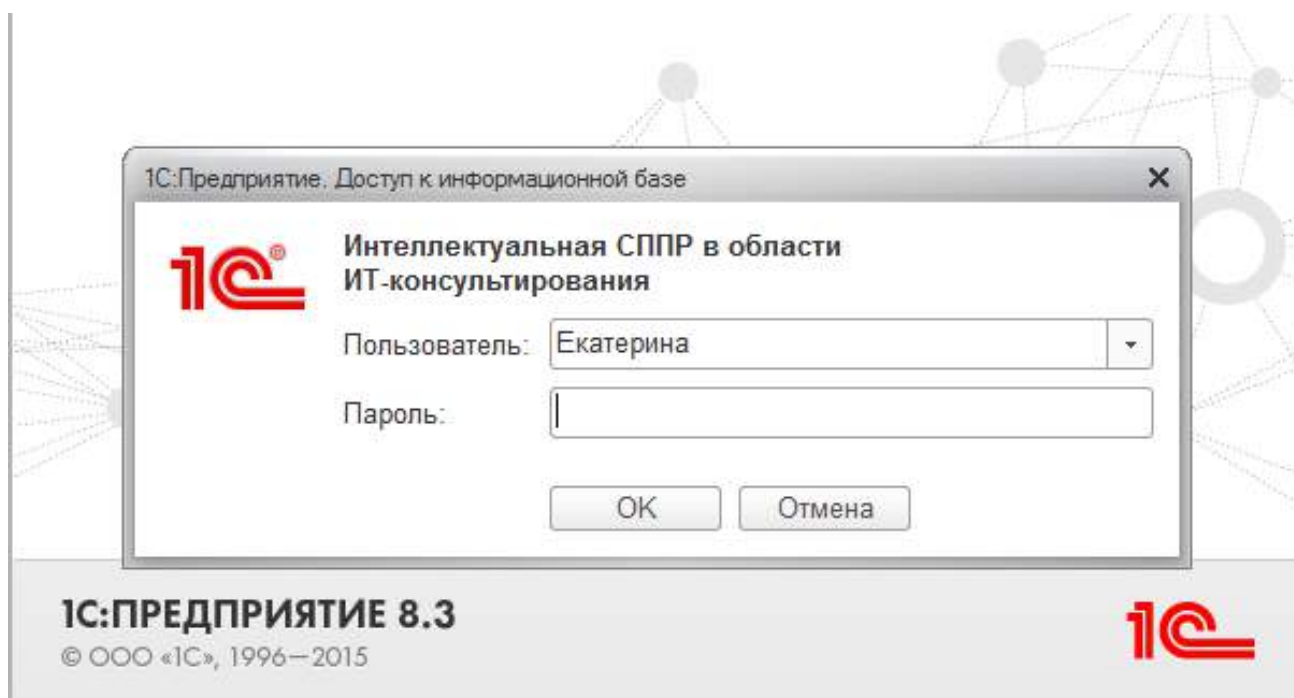


Рисунок 4.4. Вход в СППР

СППР состоит из набора справочников:

1. Версии программных продуктов – в этом справочнике содержится информация о том, какие версии программных продуктов поддерживаются в базе знаний. Версию необходимо знать, т.к. в более поздних версиях программных продуктов могут уже быть внесены исправления, и консультация по вопросу пользователя будет обозначать, что нужно установить более новую версию программы. На рис. 4.5 представлена форма справочника «Версии программных продуктов».

← →

☆

Версии программных продуктов

Создать

Наименование ↓	Код	Владелец
1С:Бухгалтерия 3.0	ЕИС	1С
1С:Бухгалтерия 7.7	0000000004	1С
1С:Бухгалтерия 8.2	0000000003	1С
1С:Зарплата и Кадры 7.7	0000000001	1С
1С:ЗУП 2.5	0000000002	1С
1С:ЗУП 3.0	0000000014	1С
Ахapta 1.0	0000000005	Аксапта
Ахapta 1.5	0000000006	Аксапта
Ахapta 2.0	0000000007	Аксапта
Ахapta 2.5	0000000008	Аксапта
Ахapta 3.0	0000000009	Аксапта
Dynamics AX 2009	0000000011	Аксапта
Dynamics AX 2012	0000000012	Аксапта
Dynamics AX 4.0	0000000010	Аксапта

Рисунок 4.5. Версии программных продуктов

Справочник состоит из полей «Наименование» – наименование программного продукта, «Код» – для быстрого поиска, и «Владелец» – компания, которая производит это программный продукт.

2. Справочник «Изменения» – в этом справочнике хранится информация о вносимых изменениях в прецеденты. Этот справочник необходим, чтобы отследить, кем и когда были сделаны изменения в прецеденте. На рис. 4.6 представлена форма справочника «Изменения».

← →

☆ **Изменения**

Создать

Поиск (Ctrl+F)

















Период	↓	Объект	Пользователь
 09.09.2016 9:35:36		При создании нового сотрудника возникает ошибка	Екатерина
 09.09.2016 9:35:51		При создании сотрудника возникает ошибка, физическо...	Екатерина
 09.09.2016 9:38:04		Как ввести отпуск по уходу за ребенком до 1,5 лет?	Екатерина
 09.09.2016 9:38:10		Как занести сальдо по количеству дней отпуска?	Екатерина
 09.09.2016 9:38:15		Проблема отражения отпуска по уходу за ребенком в та...	Екатерина
 09.09.2016 9:41:37		Как внести больничный лист по беременности и родам?	Екатерина
 09.09.2016 9:42:09		Как внести больничный лист?	Екатерина
 09.09.2016 9:42:32		Как проверить правильность занесения больничного ли...	Екатерина
 09.09.2016 9:42:47		Сотрудник заболел во время отпуска	Екатерина
 09.09.2016 9:43:06		Сотрудник заболел во время отпуска	Екатерина
 09.09.2016 9:43:17		Сотрудник заболел до отпуска, как ввести больничный ...	Екатерина
 09.09.2016 11:01:28		Как занести сальдо по количеству дней отпуска?	Екатерина
 09.09.2016 11:01:40		Проблема отражения отпуска по уходу за ребенком в та...	Екатерина
 09.09.2016 11:02:03		Как ввести отпуск по уходу за ребенком до 1,5 лет?	Екатерина
 09.09.2016 11:02:57		Как занести сальдо по количеству дней отпуска?	Екатерина
 09.09.2016 11:03:04		Проблема отражения отпуска по уходу за ребенком в та...	Екатерина

Рисунок 4.6. Изменения

Справочник состоит из полей: «Период» – дата и время, когда были сделаны изменения, «Объект» – наименование объекта, над которым было произведено изменения, «Пользователь» – имя пользователя, который внес изменения.

3. Справочник «Ошибки» – в этом справочнике хранится информация об ошибках, которые возникают в процессе работы с программой. На рис. 4.7 представлен фрагмент иерархии справочника «Ошибки».

← →
☆ Ошибки

Создать
Создать группу

Наименование	↓	Код	Описание ошибки	Технология решения	Возможность исправления пользователем самостоятельно
⊖ Ошибки					
⊖ 1. Расчеты с персоналом		000000040			
⊖ 1. Кадровый учет		000000001			
⊖ 1. Кадровые документы		000000002			
⊖ 1. Прием, перевод, увольнения		000000003			
⊕ 1. Приказы о приеме		000000004			
⊕ 2. Приказы о переводе		000000006			
⊕ 3. Приказы об увольнении		000000008			
⊕ 4. Приказы о присвоении классного чина		000000009			
⊕ 2. Отклонения		000000010			
⊖ 3. Планирование отпуска		000000026			
○ 1. График отпусков		000000027			
○ 2. Отклонения от графика отпусков		000000028			
⊖ 4. Ведение штатного расписания		000000029			
⊕ 1. Справочник подразделений		000000030			
⊕ 2. Справочник должностей		000000031			
⊕ 3. Штатное расписание		000000032			
⊕ 5. Воинский учет		000000033			
⊕ 2. Табельный учет		000000037			
⊕ 3. Расчет заработной платы		000000038			
⊕ 4. Налоги и взносы		000000039			
⊕ 5. Отчетность		000000267			
⊕ 6. Бухгалтерский учет зарплат		000000271			
⊕ 2. Бухгалтерский учет		000000041			

Рисунок 4.7. Фрагмент иерархии справочника «Ошибки»

Справочник «Ошибки» состоит из полей: «Наименование» – класс ошибок, «Код» – для быстрого поиска, «Описание ошибки», «Технология решения», «Возможность исправления пользователем самостоятельно». Рассмотрим форму ввода «Ошибки». «Наименование» – кратко сформулированное описание ошибки, которая возникла у пользователя, «Родитель» – класс, к которому относится ошибка. «Описание ошибки» – описание ошибки пользователем. «Технология решения» – полное описание последовательности действий для решения возникшей ошибки. «Возможность исправления пользователем самостоятельно» – логическое поле, выбор из вариантов «Да» или «Нет», т.е. может ли пользователь самостоятельно исправить ошибку, или нужно обязательное исправление ошибки консультантом. На рис. 4.8 представлено описание ошибки «Сообщение об ошибке «Нет вакантных мест»».

Сообщение об ошибке «Нет вакантных мест» (Ошибки) (1С:Предприятие)

Главное | Изменения | Файлы

Записать и закрыть | Записать | Еще

Наименование: Сообщение об ошибке «Нет вакантных мест»

Родитель: 1. Приказы о приеме

Описание ошибки: Сообщение об ошибке «Нет вакантных мест»

Технология решения:

- 1) Проверить состав сотрудников по коду должности и строке ШР
- 2) Проверить ставки занятости
Беременность/увольнение - Зануляется автоматически
Отпуск с заменой/длительный больничный – Занулить в руку.
- 3) Если принимаем за кого-то, то принимаем временно
- 4) Проверить в ХБ ссылки
При приеме «нет 0» – «0»
При переводе 0-0
При увольнении 0-«Нет 0»
- 5) Проверить пересечение периодов за всю историю. М.Б. когда-то раньше не стыковки
- 6) RPayHistory выгрузить в excel Проверить HRMStaff, LastStaff

Возможность исправления пользователем самостоятельно: ☐

Рисунок 4.8. Сообщение об ошибке «Нет вакантных мест»

4. Справочник «Программные продукты» – в этом справочнике хранится информация о программных продуктах, по которым можно найти прецеденты в базе знаний. На рис. 4.9 представлена форма справочника, которая состоит из наименования и кода.

← → ☆ Программные продукты

Создать

Наименование ↓	Код
1С	1С
Аксapta	Аксapta

Рисунок 4.9. Справочник «Программные продукты»

5. Справочник «Роли пользователей» – в этом справочнике хранится информация о ролях пользователей. В программе 1С версии 3.0 появилась новинка – конвейерные документы, это значит, что кадровик создает, например, документ «Отпуск», а расчетчик этот же документ «рассчитывает» и когда возникает ошибка важно знать на стороне какого пользователя она возникла.

На рис. 4.10 представлена форма справочника «Роли пользователей» которая состоит из наименования и кода.

← → ☆ Роли пользователей

Создать

Наименование ↓	Код
– Бухгалтер	000000002
– Главный бухгалтер	000000007
– Договорник	000000006
– Зам. главного бухгалтера	000000008
– Кадровик	000000001
– ОТиЗ	000000005
– Расчетчик	000000004
– Табельщик	000000003

Рисунок 4.10. Справочник «Роли пользователей»

6. Регистр сведений «Хранилище файлов» – в этом регистре хранится информация о файлах, которые могут помочь в решении вопроса пользователя. Регистр содержит следующие поля: «Объект» – ошибка или прецедент, к которому принадлежит этот файл, «Имя файла» – это место откуда был загружен файл, «Описание файла» – краткое описание файла пользователем базы знаний. На рис. 4.11 представлена структура регистра сведений.

← → ☆ Хранилище файлов

Создать

Объект	Имя файла	Описание файла
✉ Как проверить правильность занесения больничного листа?	C:\Users\oysv\Desktop\Картинки дл БЗ\ошибки.png	
✉ Как занести сальдо по количеству дней отпуска?	C:\IC_Базы\База знаний\картинки\XB.jpg	Копирование данных - Кнопка

Рисунок 4.11. Регистр сведений «Хранилище файлов»

В следующем подразделе подробно представлена структура прецедента из базы знаний.

4.3. Описание прецедента в СППР

Была разработана СППР на платформе 1С. СППР содержит множество прецедентов, которые разделены на классы в соответствии с онтологией. Рассмотрим структуру прецедента. Прецедент представляет собой форму, которая состоит из 3 вкладок: «Главное», «Изменения» и «Файлы» [55].

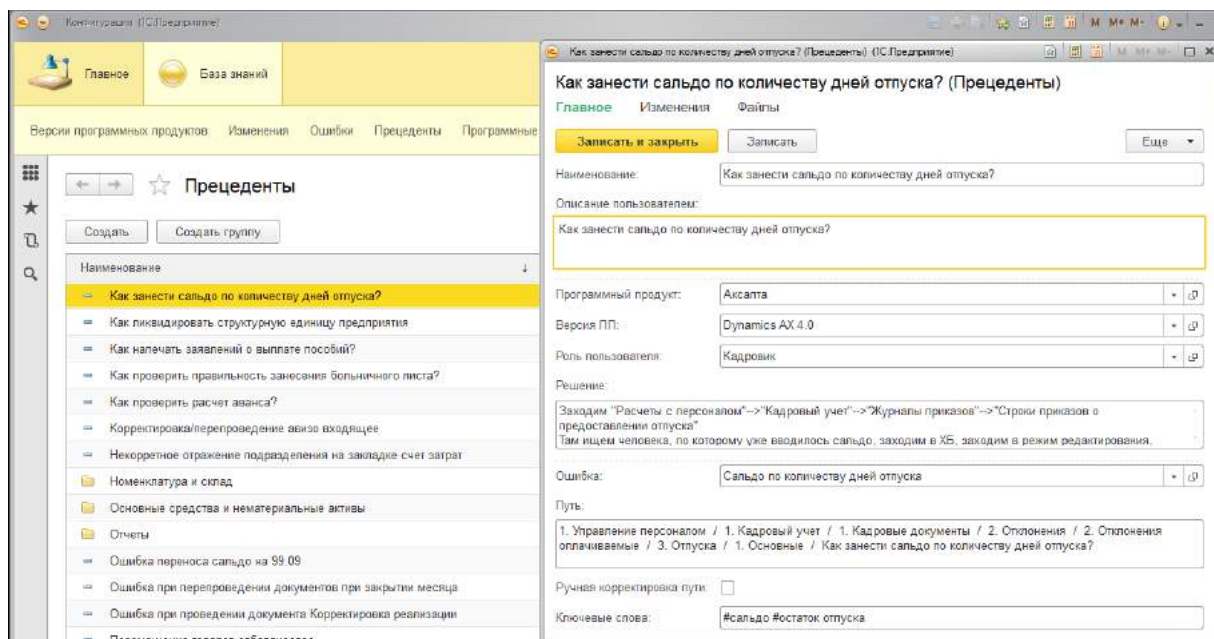


Рисунок 4.12. Прецедент, вкладка «Главное»

На вкладке «Главное» расположены основные поля прецедента:

Наименование – краткое описание проблемы пользователя, которое формулируется консультантом по ПП. Описание пользователя – информация, которую пользователь сообщает консультанту при формулировании запроса, например, в виде заявки или по телефону. Программный продукт - выбор из списка: 1С, АКСАПТА – наиболее популярные программные продукты, которые используются на сегодня в этих предметных областях. Версия ПП – релиз, т.к. программные продукты постоянно дорабатываются, в них исправляются ошибки, они обновляются, то при ответе пользователю на вопрос обязательно понимание с каким релизом он работает, может быть решением его проблемы будет просто обновление базы. Роль пользователя – кадровик, бухгалтер, табельщик; от роли зависит функционал, который можно использовать при решении проблемы пользователя. Технология решения вопроса – полное описание последовательности действий пользователя для решения его проблемы. Ошибка (возможно заполнено\возможно нет) – техническая ошибка, которая может быть решена только силами программистов. Путь – место, где возникает проблема. Возможность ручной корректировки пути – логическое поле, дает возможность изменить место

нахождения проблемы. Ключевые слова – основные слова, которые характеризуют прецедент.

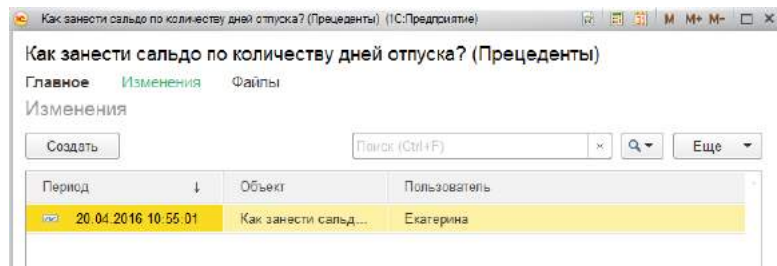


Рисунок 4.13. Прецедент, вкладка «Изменения»

На вкладке «Изменения» отражаются данные о пользователе, который вносил изменения в прецедент. Вести историю изменения прецедентов необходимо, когда с базой работает несколько консультантов. Всегда можно понять кто и когда вносил какие изменения в прецедент.

На вкладке «Файл» к прецеденту можно прикрепить файл, например, с ошибкой, возникающей в этом прецеденте, или с обработкой, которая поможет решить проблему, или с инструкцией по устранению проблемы и т.п.

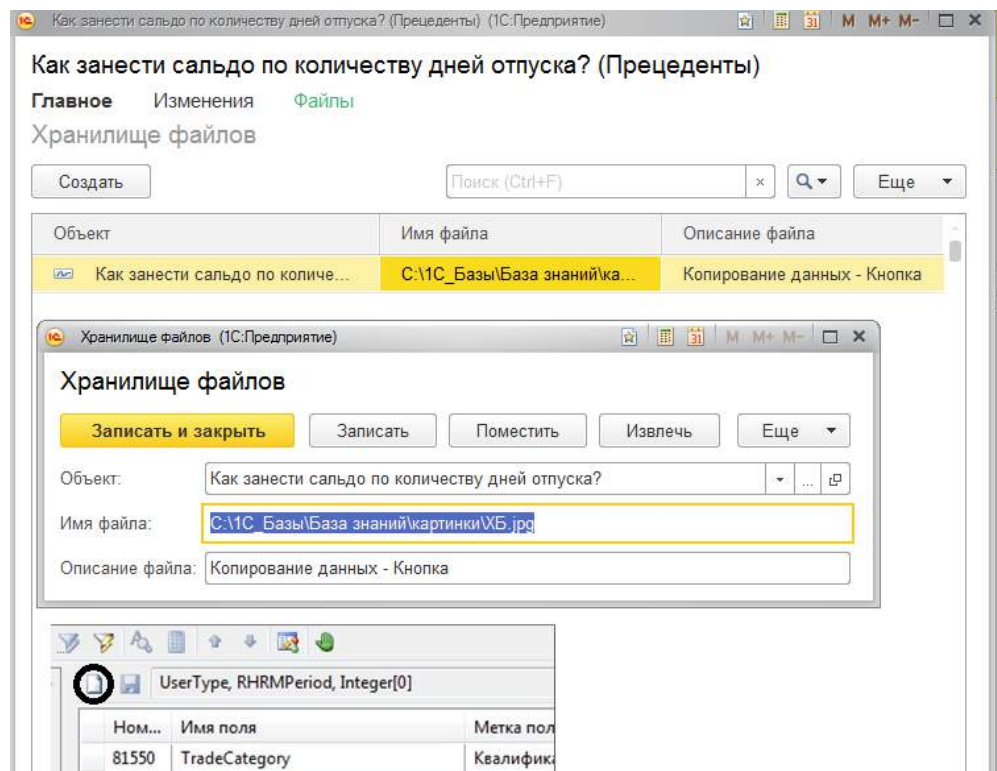


Рисунок 4.14. Прецедент, вкладка «Файл»

4.4. Экспериментальные исследования гибридной модели на наборе прецедентов в области ИТ-консультирования

Разные типы функции принадлежности

Для проведения эксперимента с различным типами функций принадлежности был выбран набор прецедентов, из которого случайным образом были сформированы обучающая и контрольная выборки. Нечеткие правила, полученные из обучающей выборки прецедентов (см. Приложение В, табл.В.2), были применены к прецедентам из контрольной выборки для их классификации. Компьютерный эксперимент для этого набора данных был проведен 20 раз. Таблица 4.1 показывает точность классификации для фрагмента базы прецедентов.

Таблица 4.1. Тест на точность для разных типов ФП

Номер испытания	Исходная выборка	Обучающая выборка	Точность классификации	
			ФП типа T	ФП типа S
1	120	90	0,730	0,730
2	120	90	0,730	0,830
3	120	90	0,870	0,770
4	120	90	0,800	0,770
5	120	90	0,870	0,900
6	120	90	0,830	0,870
7	120	90	0,770	0,770
8	120	90	0,730	0,870
9	120	90	0,770	0,770
10	120	90	0,800	0,900
11	120	90	0,870	0,870
12	120	90	0,730	0,770
13	120	90	0,800	0,900
14	120	90	0,700	0,870
15	120	90	0,730	0,770
16	120	90	0,870	0,870
17	120	90	0,830	0,930
18	120	90	0,870	0,870
19	120	90	0,870	0,970
20	120	90	0,830	0,800
Среднее	120	90	0,800	0,840

Точность классификации для разных типов ФП

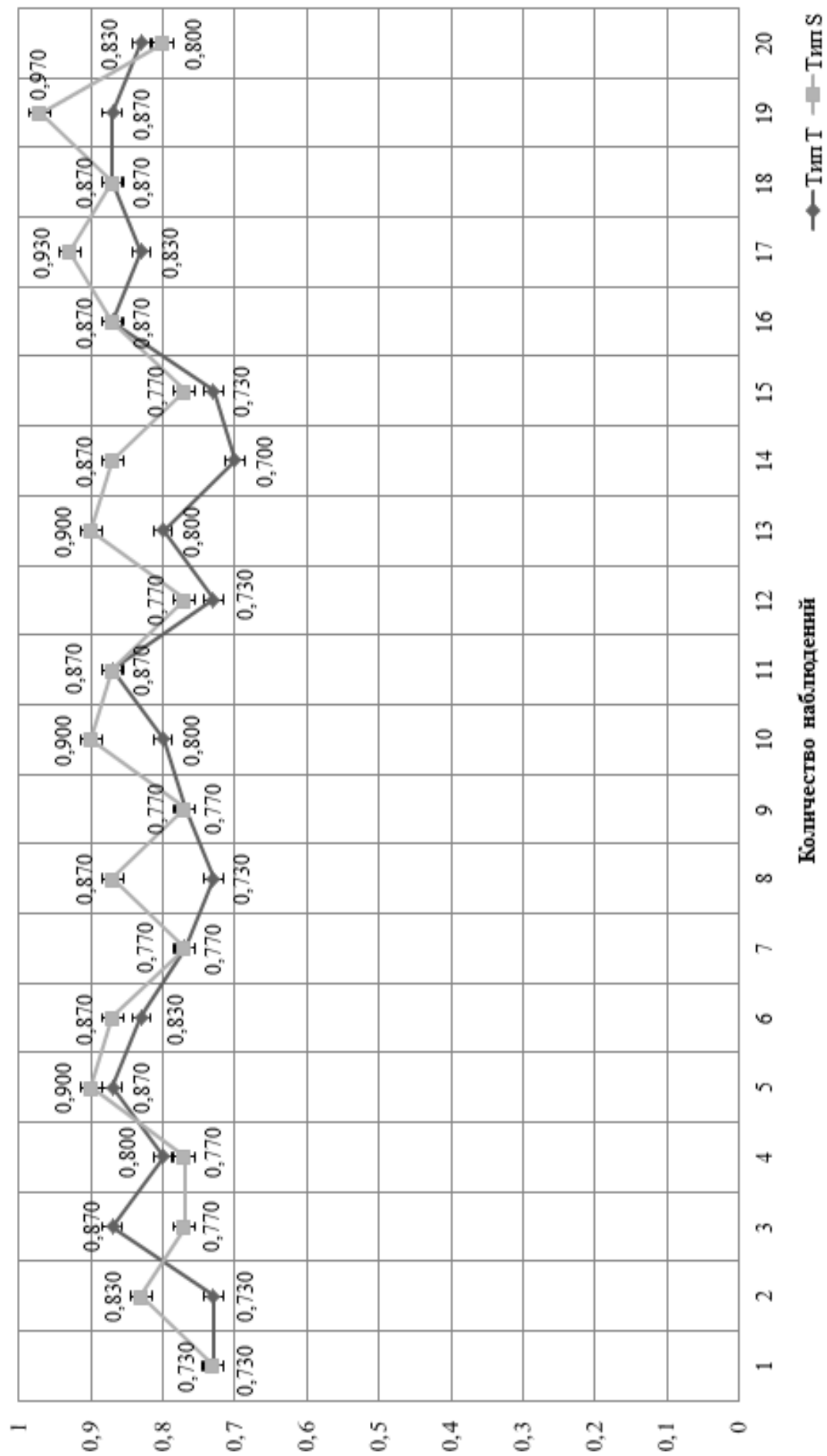


Рисунок 4.15. Точность классификации для разных типов ФП

Исследования точности для разных типов ФП для классификации прецедентов на реальных данных, полученных при индексировании прецедентов концептами онтологии, подтверждают результаты, полученные на контрольных выборках (рис. 4.15). Среднее значение точности для гладких функций принадлежности типа S оказалось выше, чем для треугольных функций принадлежности типа T .

Разное количество функций принадлежности

Для проведения эксперимента с разным количеством функций принадлежности был выбран набор прецедентов, из которого случайным образом были сформированы обучающая и контрольная выборки. Нечеткие правила, полученные из обучающей выборки прецедентов, были применены к прецедентам из контрольной выборки для их классификации. Компьютерный эксперимент для этого набора данных был проведен 20 раз. Для каждого атрибута было определено минимальное и максимальное значение на обучающей выборке. Далее каждый полученный интервал был разделен на 3, 5 или 7 ФП. Таблица 4.2 представленная ниже, показывает точность классификации на контрольной выборке на базе прецедентов.

Таблица 4.2. Тест на точность для разного количества ФП

Номер испытания	Точность классификации		
	3 ФП	5 ФП	7 ФП
1	0,730	0,830	0,800
2	0,830	0,730	0,700
3	0,770	0,830	0,570
4	0,770	0,830	0,670
5	0,900	0,730	0,570
6	0,870	0,830	0,900
7	0,770	0,800	0,800
8	0,870	0,600	0,670
9	0,770	0,730	0,730
10	0,900	0,730	0,700
11	0,870	0,630	0,730
12	0,770	0,700	0,670
13	0,900	0,700	0,700
14	0,870	0,730	0,630
15	0,770	0,730	0,770
16	0,870	0,600	0,870
17	0,930	0,800	0,800
18	0,870	0,730	0,770
19	0,970	0,730	0,700
20	0,800	0,800	0,700
Среднее	0,840	0,740	0,720

Точность классификации для разного количества ФП

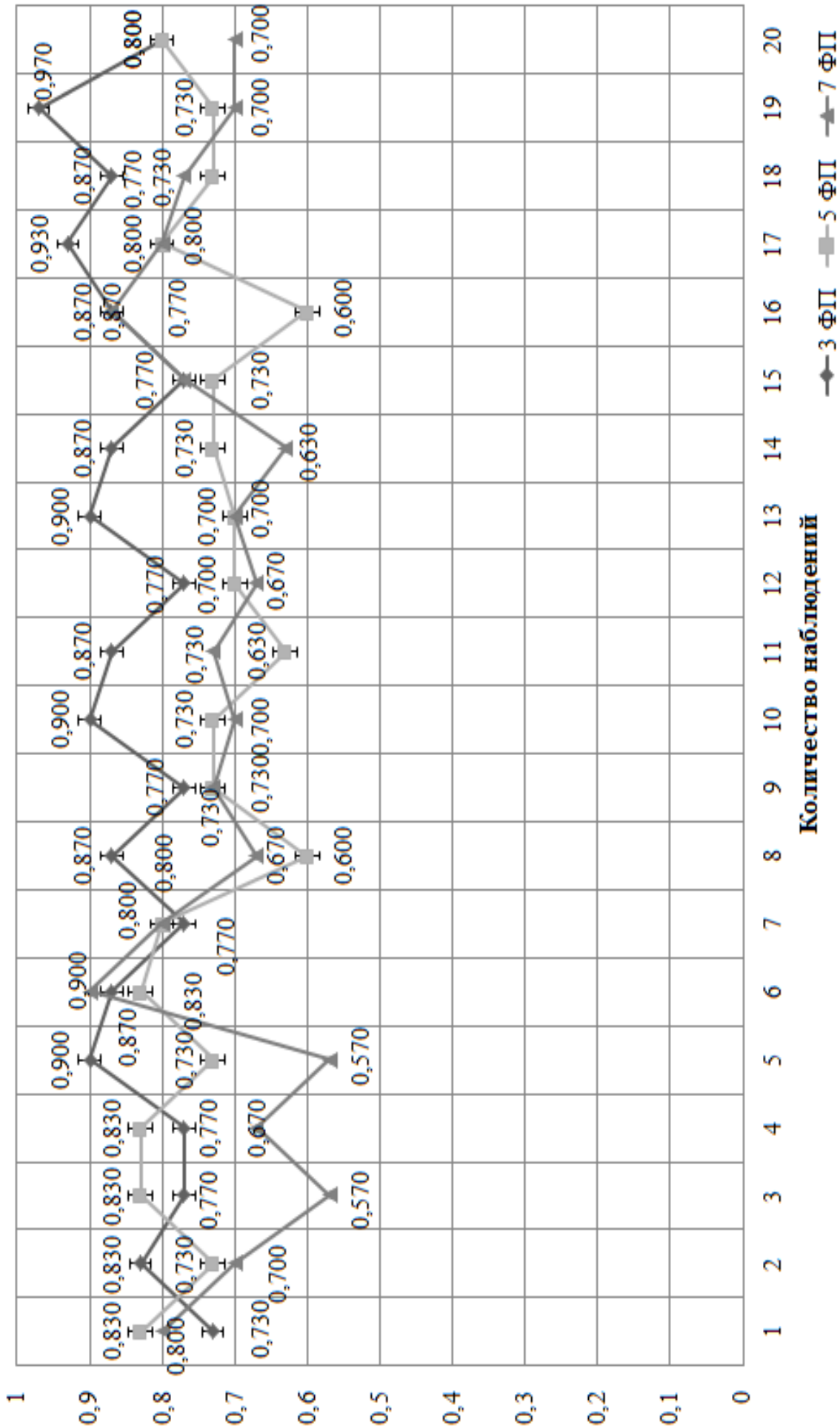


Рисунок 4.16. Точность классификации для разного количества ФП

Исследования точности для разного количества ФП для классификации прецедентов на реальных данных, полученных при индексировании прецедентов концептами онтологии, подтверждают результаты, полученные на контрольных выборках (рис. 4.16). Среднее значение точности для 3 ФП выше по сравнению с другими. При увеличении числа ФП классификационная точность снижается, т.к. количество сочетаний становится больше и не все данные попадают в обучающую выборку. Из этого можно сделать вывод, что наибольшую классификационную точность, можно получить, используя только 3 ФП на данном наборе данных.

Точность классификации при уменьшении прецедентов в ОВ

Интересное явление наблюдалось в экспериментах, в которых после обучения нечетких правил, оригинальная база прецедентов для обучения могла быть уменьшена до малых размеров, и при этом предоставляла высокую точность классификации для прецедентной системы. Подтверждение результатов можно увидеть в таблицах 4.3-4.7. Исходный размер базы прецедентов для всех испытаний 120.

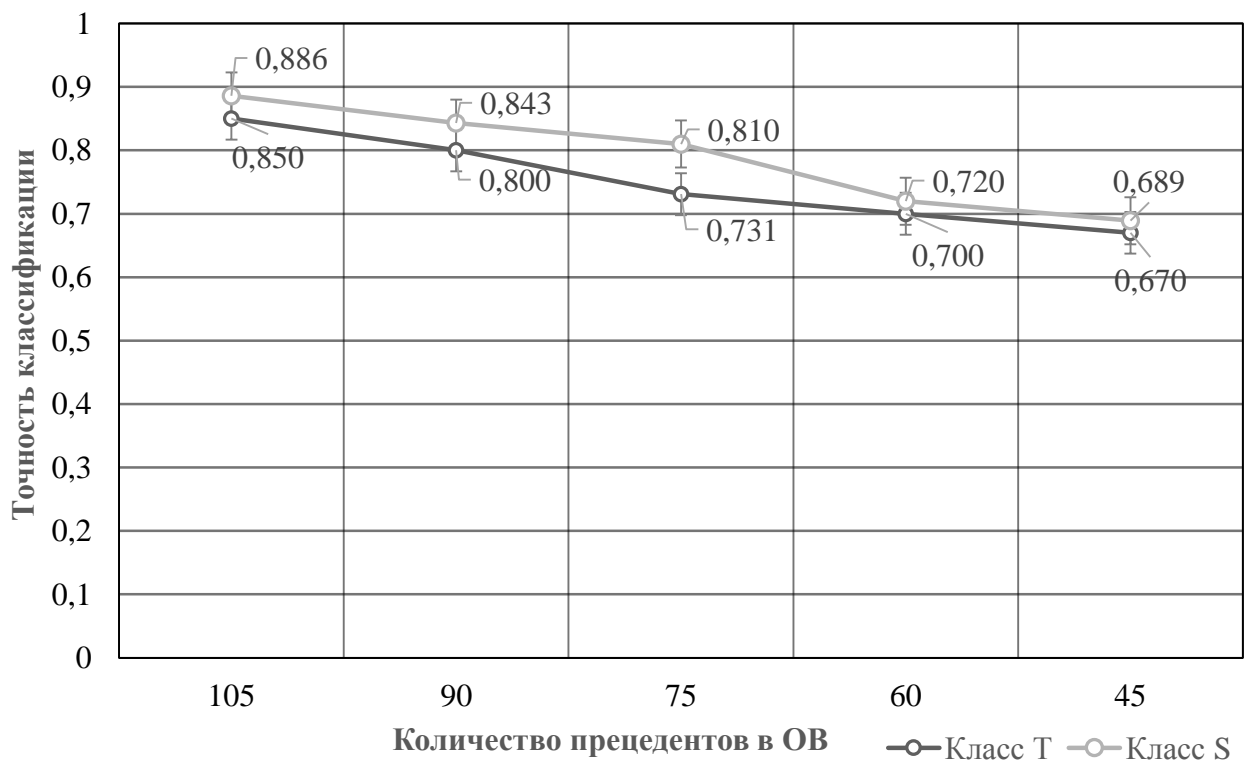


Рисунок 4.17. Точность классификации

Таблица 4.3 Точность классификации при ОВ 105 прецедентов

№ испытаний	ОВ	Точность на тестовых данных	
		ФП типа Т	ФП типа S
1	105	0,770	0,900
2	105	0,870	0,830
3	105	0,870	0,870
4	105	0,900	0,930
5	105	0,870	0,900
6	105	0,830	0,870
7	105	0,870	0,830
8	105	0,900	0,870
9	105	0,770	0,870
10	105	0,870	0,930
11	105	0,870	0,870
12	105	0,770	0,900
13	105	0,800	0,900
14	105	0,900	0,870
15	105	0,730	0,930
16	105	0,870	0,870
17	105	0,830	0,930
18	105	0,870	0,870
19	105	0,870	0,970
20	105	0,830	0,800
21	105	0,870	0,870
22	105	0,830	0,900
23	105	0,900	0,870
24	105	0,930	0,870
25	105	0,870	0,930
Среднее	105	0,850	0,886

Таблица 4.4. Точность классификации при

ОВ 90 прецедентов

№ испыт аний	ОВ	Точность на тестовых данных	
		ФП типа Т	ФП типа S
1	90	0,730	0,730
2	90	0,730	0,830
3	90	0,870	0,770
4	90	0,800	0,770
5	90	0,870	0,90
6	90	0,830	0,870
7	90	0,770	0,770
8	90	0,730	0,870
9	90	0,770	0,770
10	90	0,800	0,900
11	90	0,870	0,870
12	90	0,730	0,770
13	90	0,800	0,900
14	90	0,700	0,870
15	90	0,730	0,770
16	90	0,870	0,870
17	90	0,830	0,930
18	90	0,870	0,870
19	90	0,870	0,970
20	90	0,830	0,800
21	90	0,870	0,870
22	90	0,730	0,770
23	90	0,870	0,870
24	90	0,730	0,870
25	90	0,800	0,900
Среднее	90	0,800	0,843

Таблица 4.5. Точность классификации при

ОВ 75 прецедентов

№ испыта ний	ОВ	Точность на тестовых данных	
		ФП типа Т	ФП типа S
1	75	0,700	0,830
2	75	0,800	0,870
3	75	0,700	0,770
4	75	0,700	0,830
5	75	0,730	0,830
6	75	0,670	0,870
7	75	0,700	0,900
8	75	0,700	0,830
9	75	0,600	0,830
10	75	0,670	0,870
11	75	0,630	0,830
12	75	0,700	0,830
13	75	0,770	0,770
14	75	0,730	0,800
15	75	0,730	0,830
16	75	0,770	0,730
17	75	0,770	0,730
18	75	0,800	0,830
19	75	0,800	0,830
20	75	0,670	0,770
21	75	0,730	0,730
22	75	0,830	0,800
23	75	0,770	0,770
24	75	0,770	0,800
25	75	0,830	0,770
Среднее	75	0,731	0,810

Таблица 4.6. Точность классификации
при ОВ 60 прецедентов

№ испыт аний	ОВ	Точность на тестовых данных	
		ФП типа Т	ФП типа S
1	60	0,770	0,570
2	60	0,700	0,670
3	60	0,770	0,700
4	60	0,770	0,670
5	60	0,700	0,870
6	60	0,630	0,730
7	60	0,670	0,530
8	60	0,630	0,630
9	60	0,770	0,630
10	60	0,530	0,770
11	60	0,670	0,730
12	60	0,670	0,800
13	60	0,670	0,770
14	60	0,670	0,800
15	60	0,630	0,670
16	60	0,630	0,670
17	60	0,670	0,630
18	60	0,770	0,770
19	60	0,670	0,800
20	60	0,830	0,830
21	60	0,630	0,800
22	60	0,770	0,770
23	60	0,630	0,770
24	60	0,830	0,730
25	60	0,830	0,700
Среднее	60	0,700	0,720

Таблица 4.7. Точность классификации при
ОВ 45 прецедентов

№ испыта ний	ОВ	Точность на тестовых данных	
		ФП типа Т	ФП типа S
1	45	0,670	0,700
2	45	0,630	0,630
3	45	0,630	0,630
4	45	0,670	0,630
5	45	0,630	0,630
6	45	0,670	0,700
7	45	0,700	0,670
8	45	0,670	0,770
9	45	0,700	0,670
10	45	0,670	0,630
11	45	0,600	0,570
12	45	0,670	0,630
13	45	0,670	0,770
14	45	0,630	0,630
15	45	0,670	0,630
16	45	0,630	0,730
17	45	0,630	0,730
18	45	0,670	0,670
19	45	0,700	0,730
20	45	0,630	0,770
21	45	0,700	0,800
22	45	0,770	0,770
23	45	0,700	0,770
24	45	0,730	0,700
25	45	0,700	0,670
Среднее	45	0,670	0,689

Из рис. 4.17 и таблиц 4.3-4.7 видно, что при значительном сокращении обучающей выборки (до 45 прецедентов из 120) точность поиска остается достаточно высокой, это подтверждает гипотезу о том, что при небольшом количестве исходных данных с помощью предложенного алгоритма можно делать прогнозы с достаточным уровнем точности.

4.5. Выводы

В главе 4 была описана реализация интеллектуальной системы поддержки принятия решений, которая основана на эффективном управлении знаниями, помогающая использовать опыт консультантов в ИТ-подразделении. Для наполнения базы знаний прецедентами было сформулировано по области учета расчетов с персоналом около 250 прецедентов, по области бухгалтерского учета около 250 прецедентов. В настоящее время база прецедентов находится на стадии дальнейшего заполнения.

Для демонстрации работы системы рассмотрено ИТ-подразделение, и деятельность основных работников этого подразделения – консультантов. Предложены критерии и методика эффективного поиска прецедентов для решения вопросов пользователей, основанные на опыте консультантов ИТ отдела, позволяющие значительно сократить время, потраченное на решение вопросов пользователей.

Интеграция прецедентной модели с нечеткой логикой на реальных данных показала хорошие результаты, точность классификации составила выше 0,8, что подтверждено результатами экспериментальных исследований.

В главе представлено описание реализации СППР и базы прецедентов на языке 1С. Разработанная СППР имеет простой и удобный интерфейс, позволяет пользователю достаточно быстро вносить прецеденты в базу, и достаточно быстро и эффективно получать решения из базы прецедентов и осуществлять свою основную консультационную деятельность. Эффективность работы системы подтверждается актами внедрения, представленными в приложении 4.

Применение прецедентного подхода в сфере консультирования пользователей, где часто приходится оказывать помощь пользователям на

основе опыта, полученного при решении аналогичных задач в прошлом, оказалось весьма эффективным средством, значительно сокращающим время, необходимое для принятия решений.

Кроме того, разработанная система прошла государственную регистрацию как программа для ЭВМ, о чём Федеральной службой по интеллектуальной собственности, патентам и товарным знакам выдано свидетельство № 2017613946 от 04 апреля 2017 г., и № 2017615280 от 11 мая 2017 г.

ЗАКЛЮЧЕНИЕ

Проведенные в диссертационной работе исследования образуют теоретическую и практическую основу для создания новой информационной технологии поддержки принятия решений в области ИТ-консультирования. Основными результатами исследования являются:

1. Предложен новый подход к организации интеллектуальной поддержки принятия решений в области ИТ-консультирования на основе метода прецедентов. Разработана структура прецедентов, хранимых в базе знаний и используемых для решения новых задач. Произведено наполнение базы прецедентов ретроспективной информацией.

2. Разработана гибридная модель представления и извлечения знаний, интегрирующая прецеденты принятия решений и онтологию предметной области, а также метод индексирования прецедентов онтологией предметной области, отличительным преимуществом которого является возможность связывания прецедента с несколькими концептами онтологии с опосредованным вычислением весов терминальных концептов, наиболее точно определяющих семантику данного фрагмента ретроспективных знаний.. Данный инструментарий позволяет повысить качество взаимодействия консультантов-аналитиков с пользователями программного обеспечения, являющимися специалистами в предметной области, способствует лучшей идентификации проблемы пользователя, что приводит к повышению релевантности извлекаемых прецедентов.

3. Построена онтология, описывающая наиболее востребованные у ИТ-пользователей области: бухгалтерский и кадровый учет, расчеты с персоналом, оформление договорных отношений с клиентами. Иерархия концептов дополнена свойствами – весовыми коэффициентами, задающими силу иерархических связей, а также синонимами концептов. Последняя возможность оказывается полезной, когда специалисты различных предметных областей используют разные термины для одних и тех же понятий.

4. Разработан метод формирования нечетких правил для классификации прецедентов. Предложенный алгоритм применим для классификации данных, которые могут быть описаны как числовыми, так и категориальными признаками, что позволяет дополнять полученные правила необходимым числом конкретизирующих предпосылок для динамической настройки механизма выдачи релевантных прецедентов. Проведенные исследования показывают достаточную классификационную точность метода как на тестовых, так и на реальных данных при соответствующем выборе типа и числа функций принадлежности.

5. Разработана интеллектуальная система поддержки принятия решений, которая обеспечивает сбор и хранение прецедентов решаемых проблем, а также предоставляет консультанту возможность автоматического выбора семантически близких прецедентов для решения вопроса пользователя.

6. Разработанные онтологическая модель знаний, модельное, алгоритмическое и программное обеспечение информационной системы поддержки принятия решений могут быть использованы в качестве базовых для создания интеллектуальных систем консультирования в ИТ-подразделениях организаций любого профиля.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Авдеенко Т.В., Макарова Е.С. Метод определения релевантности прецедентов на основе нечетких лингвистических правил // Научный вестник НГТУ, №1(62). – Новосибирск: Изд-во НГТУ, 2016. – с. 17-34.
2. Авдеенко Т.В., Бакаев М.А. Гибридная модель представления знаний для реализации вывода во фреймовой онтологии // Научный вестник НГТУ, № 3(52). – Новосибирск: Изд-во НГТУ, 2013. – с. 84-90.
3. Авдеенко Т.В., Макарова Е.С. Система поддержки принятия решений В IT-подразделениях на основе интеграции прецедентного подхода и онтологии// Вестник АГТУ Серия: Управление, вычислительная техника и информатика, №3. – Астрахань: Изд-во АГТУ, 2017. – с. 85-99.
4. Астахова Н.И., Москвитина Г.И. Теория управления. М.: Изд-во Юрайт, 2013. – 375 с.
5. Бениаминов Е.М., Болдина Д.М. Система представления знаний Ontolingua – принципы и перспективы. / Бениаминов Е. М., Болдина Д. М. // URL: <http://beniaminov.rsuh.ru/Stanford.pdf> (дата обращения: 01.02.2015).
6. Бертоланфи Л. Общая теория систем: критический обзор / Л. Бертоланфи //Исследования по общей теории систем. - М.: Наука, 1973. – с. 23-82.
7. Борисов А.Н. Построение интеллектуальных систем, основанных на знаниях, с повторным использованием компонентов. /Борисов А.Н.// URL: http://www.conf.ostis.net/images/7/70/08._Борисов_А.Н._-_ПострИСОнЗсПИК.pdf(дата обращения: 15.04.2016).
8. Бурков В.Н., Коргин Н.А., Новиков Д.А. Введение в теорию управления организационными системами / Под ред. чл.-корр. РАН Д.А. Новикова. – М.: Либроком, 2009. – 264 с.
9. Быкова Н.Н. Педагогические аспекты управления знаниями по освоению дистанционных образовательных технологий в учебном процессе // Царскосельские чтения. 2016. №XX. URL: [http://cyberleninka.ru/article/n/pedagogicheskie-aspekty-upravleniya-znaniyami-po-](http://cyberleninka.ru/article/n/pedagogicheskie-aspekty-upravleniya-znaniyami-po)

osvoeniyu-distantcionnyh-obrazovatelnyh-tehnologiy-v-uchebnom-protsesse (дата обращения: 01.02.2017).

10.Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений, №2, 2009. – с. 45-57.

11.Варшавский П.Р., Еремеев А.П. Методы правдоподобных рассуждений на основе аналогий и прецедентов для интеллектуальных систем поддержки принятия решений //Новости искусственного интеллекта, №3, 2006. – с. 39-62.

12.Васильев В. И., Белков Н. В. Система поддержки принятия решений по обеспечению безопасности персональных данных//Вестник УГАТУ, № 5 (45). – Уфа: Изд-во УГАТУ, 2011. – с. 54–65.

13.Воробьев А.Д. В центре внимания модель управления знаниями // Вопросы управления. 2016. №3 (21). URL: <http://cyberleninka.ru/article/n/v-tsentre-vnimaniya-model-upravleniya-znaniyami> (дата обращения: 01.02.2016).

14.Втюрин В.А. Автоматизированные системы управления технологическими процессами. Основы АСУТП СПб.: Санкт-Петербургская Государственная Лесотехническая Академия имени С.М. Кирова, 2006. – 152 с.

15.Гаврилова Т.А. Инженерия знаний // Инновационное развитие: экономика, интеллектуальные ресурсы, управление знаниями / под ред. Б.З. Мильнера. М.: ИНФРА-М, 2009. – с. 480–500.

16.Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. - СПб: Питер, 2003. – 345 с.

17.Гладун А.Я., Рогушина Ю.В. Онтологии в корпоративных системах, Часть II // Корпоративные системы №1 / 2006 //URL: <http://www.management.com.ua/ims/ims116.html>

18.Добров Б.В., Лукашевич Н.В. Лингвистическая онтология по естественным наукам и технологиям: основные принципы разработки и текущее состояние // Десятая национальная конференция по искусственному интеллекту с международным участием (Обнинск, 25-28 сентября 2006 г.) – М.: Физматлит, 2006.

19.Еремеев А.П., Куриленко И.Е. Расширение возможностей моделирования временных зависимостей в интеллектуальных системах на основе применения темпоральных прецедентов / Интеллектуальные системы. Коллективная монография. Выпуск 6. / Под. Ред. В.М. Курейчика. – М.: Физматлит, 2013. – с. 89-118.

20.Еремеев А.П., Куриленко И.Е. Реализация механизма временных рассуждений в современных интеллектуальных системах // Известия РАН. Теория и системы управления, №2, 2007. – с. 120–136.

21.Ефименко И.В., Хорошевский В.Ф. Онтологическое моделирование экономики предприятий и отраслей современной России: Часть 1. Онтологическое моделирование: подходы, модели, методы, средства, решения: препринт WP7/2011/08 (ч. 1) [Текст] / И. В. Ефименко, В. Ф. Хорошевский; Нац. исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2011. – 76 с.

22. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. – Новосибирск: ИМ СО РАН, 1999. – 270 с.

23.Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л. Заде. – М.: Мир, 1976. – 168 с.

24.Захарова И.В. Математическая модель семантического поиска с использованием онтологического подхода: диссертация ... кандидата физико-математических наук: 05.13.18 / Захарова Ирина Викторовна - Челябинск, 2010. - 120 с.

25.Звягинцев И.В., Рузакова О.В. Формирование системы управления знаниями в организации с использованием неявных знаний // Фундаментальные исследования. 2015. №2-20. URL: <http://cyberleninka.ru/article/n/formirovanie-sistemy-upravleniya-znaniyami-v-organizatsii-s-ispolzovaniem-neyavnyh-znaniy> (дата обращения: 10.02.2016).

26.Информационные системы в экономике: Учебник для студ. высш. учеб. заведений / В.Б. Уткин, К.В. Балдин. – М.: Издательский центр «Академия», 2004. – 288 с.

27. Информационные технологии управления: Учебное пособие / Под ред. Ю.М. Черкасова. - М.: ИНФРА-М, 2001. – 216 с.

28. Казекин М.М. История языков представления онтологий // Компьютерные инструменты в образовании. - СПб.: Центр информатизации образования "КИО", №4, 2008. – с. 3-11.

29. Календжян С.О., Бёме Г. Система эффективного управления. Теория и практика применения делегирования полномочий и ответственности. М.: Издательский дом «Дело» РАНХиГС, 2012. – 112 с.

30. Калиниченко Л.А. Поддержка баз данных с онтологическими зависимостями на основе дескриптивных логик// Сборник трудов Второго симпозиума «Онтологическое моделирование», Москва, ИПИ РАН, 2011 г. – с.1-12.

31. Карпов Л.Е., Юдин В.Н. Адаптивное управление по прецедентам, основанное на классификации состояний управляемых объектов // Труды Института Системного Программирования РАН, т.13, ч.2. М., 2007. – с. 37-57.

32. Карпов Л.Е., Юдин В.Н. Методы добычи данных при построении локальной метрики в системах вывода по прецедентам // Институт Системного Программирования РАН, Препринт, 2006.

33. Качала В.В. Концепция управления знаниями в консалтинговой компании // Статистика и экономика. 2005. №3. URL: <http://cyberleninka.ru/article/n/kontseptsiya-upravleniya-znaniyami-v-konsaltingovoy-kompanii> (дата обращения: 28.07.2017).

34. Кобринский Б.А. Извлечение экспертных знаний: групповой вариант // Новости искусственного интеллекта, №3, 2004. – с.58-66.

35. Когаловский М.Р. Семантическое структурирование контента научных электронных библиотек на основе онтологий / М.Р. Когаловский Паринов. С. И // В сб. «Современные технологии интеграции информационных ресурсов: сборник научных трудов», 2011. – Вып. 2.

36. Константинова Н.С. Онтологии как системы хранения знаний// Н.С. Константинова, О.А. Митрофанова, Санкт-Петербургский государственный университет, 2012. – с.54.

37.Ле Хоай Исследование и разработка электронных библиотек на основе явного описания семантики ресурсов с использованием технологий Semantic Web: диссертация ... кандидата технических наук: 05.13.11 / Ле Хоай – Новосибирск, 2014. – 181 с.

38.Ляпунов А.М. «Общая задача об устойчивости движения»/М.:Изд-во Технико-теоретической литературы. 1892.

39.Макарова Е.С. Анализ существующих моделей представления знаний и их интеграция для решения задач медицинской диагностики / Е. С. Макарова // Информатика: проблемы, методология, технологии : материалы 13 междунар. науч.-метод. конф., Воронеж, 7–8 февр. 2013 г. В 4 т. – Воронеж: ИПЦ Воронеж. гос. ун-та, 2013. – Т.2. – с. 298-302.

40.Макарова Е.С. Интеллектуальные технологии для решения задачи постановки диагноза / Е.С. Макарова // Innovative information technologies : international scientific - practical conference, Prague, 22-26 apr. 2013 = Инновационные информационные технологии: материалы междунар. науч.-практ. конф., Прага, 22-26 апр. 2013. – Москва: МИЭМ, 2013. – Т. 4. – с. 160-164.

41.Макарова Е.С. Исследование влияния параметров нечеткой модели на точность классификации прецедентов//Вестник АГТУ. Серия: Управление, вычислительная техника и информатика, №4. – Астрахань: Изд-во АГТУ, 2017. – с. 7-18.

42.Макарова Е.С. Модель байесовской сети для решения задачи постановки диагноза / Е.С. Макарова // Современные материалы, техника и технология: материалы 3-й Международной научно-практической конференции, 27 декабря 2013 г. – Курск, 2013. – Т2. – с.308-311.

43.Макарова Е.С. Интеллектуальная поддержка управления IT-подразделением//Материалы Всероссийской научно-практической

конференции студентов, аспирантов и молодых ученых(19-20 ноября 2015 года), Юргинский технологический институт. – Томск: Изд-во ТПУ, 2015. – с.163-165.

44.Макарова Е.С. Архитектура модели байесовской сети и прецедентной модели // Наука. Технологии. Инновации //Материалы всероссийской научной конференции молодых учёных, НГТУ. – Новосибирск: Изд-во НГТУ, 2014. – с.50-52.

45.Макарова Е.С. Гибридная модель для интеллектуальной поддержки пользователей//Материалы 4-й Международной молодежной научной конференции (19-20 ноября 2015 года), Юго-Зап. гос. ун-т., Курск: Изд-во ЮЗГУ, 2015. – Т3. – с. 52-55.

46.Макарова Е.С. Гибридная модель для интеллектуальной поддержки управления отделом консультантов ИТ-подразделения // Материалы Всероссийского молодежного научного форума (12-14 ноября 2015 года), НГТУ. – Новосибирск: Изд-во НГТУ, 2015. – с. 38-43.

47.Макарова Е.С. Диалоговый прецедентный подход//Материалы 4-й Международной научно-практической конференции современные инновации в науке и технике, Юго-Зап. гос. ун-т., Курск: Изд-во ЮЗГУ, 2014. – Т3. – с.30-34.

48.Макарова Е.С. Нечеткая логика в прецедентном подходе/Авдеенко Т.В, Макарова Е.С.// Роль технических наук в развитии общества: сборник материалов Международной научно-практической конференции (26-27 ноября 2015 года), КузГТУ. – Кемерово: Изд-во КузГТУ, 2015. – с.57-60.

49.Макарова Е.С. Интеллектуальная поддержка работы ИТ-подразделения на основе метода прецедентов// Теоретический и практический взгляд на современное состояние науки: сборник материалов Международной научно-практической конференции (29-30 сентября 2015 года) КузГТУ, Кемерово: Изд-во КузГТУ, 2015. – с.72-74.

50.Макарова Е.С. Интеллектуальная система для решения задачи консультирования // Современные инструментальные системы,

информационные технологии и инновации: сборник научных трудов XII-ой Международной научно-практической конференции (19-20 марта 2015 года)/ Юго-Зап. гос. ун-т., Курск: Изд-во ЮЗГУ, 2015. – Т3. – с. 16-20.

51.Макарова Е.С. Нечеткие правила для оценки подобия в прецедентном подходе//Материалы 5-й Международной научно-практической конференции (29-30 декабря 2015 года), Юго-Зап. гос. ун-т., Курск: Изд-во ЮЗГУ, 2015. – с. 87-90.

52.Макарова Е.С. Нечеткие правила для оценки сходства прецедентов// Материалы VI Международной молодежной научной конференции (25-26 февраля 2016 года), Юго-Зап. гос. ун-т., Курск: Изд-во ЮЗГУ, 2016. – Т3. – с. 73-77.

53.Макарова Е.С. Прецедентный подход для решения задачи постановки диагноза//Материалы 2-й Международной научно-практической конференции будущее науки, Юго-Зап. гос. ун-т., Курск: Изд-во ЮЗГУ, 2014. – Т2. – с.15-18.

54.Макарова Е.С. Разработка информационно-советующей системы на основе прецедентной модели // Наука. Технологии. Инновации//Материалы всероссийской научной конференции молодых учёных НГТУ, Новосибирск: Изд-во НГТУ, 2012. – с. 27-28.

55.Макарова Е.С. Система поддержки принятия решений на основе прецедентного подхода//Молодежь и XXI век - 2015: материалы V Международной молодежной научной конференции (26-27 февраля 2015 года), Юго-Зап. гос. ун-т., Курск: Изд-во ЮЗГУ, 2015. – Т2. – с.121-125.

56.Макарова Е.С., Авдеенко Т.В. Архитектура гибридной системы для решения задачи медицинской диагностики//Материалы XIV Международной конференции Информатика: проблемы, методология, технологии, ВГУ, Воронеж: Изд-во ВГУ, 2014. – Т3. – с.129-133.

57.Максвелл Д.К., Вышнеградский И.А., Стодола А. Теория автоматического регулирования (линеаризованные задачи) / Под ред. А.А. Андропова и И.Н. Вознесенского. М.: Изд-во АН СССР, 1949.

58. Маслов А.В. Управление знаниями в организации // Международный журнал прикладных и фундаментальных исследований. 2015. №6-3. URL: <http://cyberleninka.ru/article/n/upravlenie-znaniyami-v-organizatsii-1> (дата обращения: 05.10.2016).

59. Массель Л.В. Применение онтологического, когнитивного и событийного моделирования для анализа развития и последствий чрезвычайных ситуаций в энергетике / Проблемы безопасности и чрезвычайных ситуаций. – №2. – 2010. – с. 34-43.

60. Мескон М., Альберт М., Хедоури Ф. Основы Менеджмента (Management) / пер. Л.И. Евенко. – М.: Дело, 1997. – 704 с.

61. Методы представления знаний: Метод. указ. / Сост. И.Л. Коробова. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2003. – 24 с.

62. Мехонцева Д.М. Самоуправление и управление: вопросы общей теории систем / Д.М. Мехонцева. - Красноярск: КГУ, 2000. – 248 с.

63. Модуль техподдержки «Битрикс» // URL: <https://www.acrit-studio.ru/1sbitrix/modul-tehpodderzhki-bitriks/> (дата обращения: 25.08.2017).

64. Нечеткие множества и теория возможностей: последние достижения / под ред. Р.Р. Ягера; пер. с англ. В.Б. Кузьмина; под ред. С.И. Травкина. – М.: Радио и связь, 1986. – 408 с.

65. Нечипоренко О.А. Использование технологии Case-Based Reasoning в проектировании программных систем // Перспективные информационные технологии и информационные среды. – 2002. – № 3. – с. 27–32.

66. Никоненко А.А. Обзор баз знаний онтологического типа / А.А. Никоненко // Искусственный интеллект. – 2002. – № 4. – с. 157–163.

67. Новак В. Математические принципы нечёткой логики / В. Новак, И. Перфильева, И. Мочкрож; под ред. Аверкина А.Н. – М.: ФИЗМАТЛИТ, 2006. – 352 с.

68. Новиков А.М., Новиков Д.А. Методология. – М.: Синтег, 2007. – 668 с.

69. Новиков Д.А. Модели и методы управления развитием региональных образовательных систем. М.: ИУО РАО, 2001. – 83 с.

- 70.Новиков Д.А. Теория управления образовательными системами. – М.: Народное образование, 2009. – 416 с.
- 71.Новиков Д.А. Введение в теорию управления образовательными системами. М.: Эгвес, 2009. – 156 с.
- 72.Новиков Д.А. Принципы управления образовательными системами. М.: НП АПО, 2010. – 48 с.
- 73.Новиков Д.А., Глотова Н.П. Модели и механизмы управления образовательными сетями и комплексами. М.: Институт управления образованием РАО, 2004. – 142 с.
- 74.Овдей, О.М. Обзор инструментов инженерии онтологий//О.М. Овдей, Г.Ю. Проскудина // Институт программных систем НАН Украины.–2004. – Т.7. – URL: <http://rcdl.ru/doc/2004/paper26.pdf> (дата обращения: 17.04.2016).
- 75.Овчинникова Е.В. Значение системы управления знаниями в образовании // Царскосельские чтения. 2016. №XX. URL: <http://cyberleninka.ru/article/n/znachenie-sistemy-upravleniya-znaniyami-v-obrazovanii> (дата обращения: 07.02.2016).
- 76.Онтологические методы и средства обработки предметных знаний: монография / А.В. Палагин, С.Л. Крытый, Н.Г. Петренко. – Луганск: изд-во ВНУ им. В. Даля, 2012. – 324 с.
- 77.Описание программного продукта 1С. М., 1996-2016 – URL: <http://www.1c.ru/> (Дата обращения: 20.04.2015).
- 78.Осипов Г.С. Искусственный Интеллект: Состояние исследований и взгляд в будущее // Осипов Г.С./ URL: <http://www.robo-psychologist.ru/node/20>(дата обращения: 14.04.2016).
- 79.Основы теории управления экономическими системами: краткий курс лекций для аспирантов 2 курса направления подготовки 38.06.01 Экономика профиль «Экономика и управление народным хозяйством (АПК и сельское хозяйство)» / Сост. И.П. Глебов // ФГБОУ ВПО «Саратовский ГАУ». – Саратов, 2014. – 133 с.

80.Платицын В.А. Практический справочник по построению Онтологий OWL в Protégé 4 // В.А. Платицын. – Манчестерский университет. – 2012.

81.Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер с англ. – М.: Издательский дом «Вильямс», 2006. – 1408 с.

82.Ревякин И.А. Использование OTRS систем/Ревякин И.А.// URL:<https://usingotrs.wordpress.com/category/теория/> (дата обращения: 24.08.2017).

83.Росеева О.И. Организация эффективного поиска на основе онтологий / О.И. Росеева, Ю.А. Загорулько // Труды международного семинара «Диалог'2001 по компьютерной лингвистике и ее приложениям». – Аксаково, 2001. – Т. 2. – с.333-342.

84.Румизен М.К. «Управление знаниями». – М.: АСТ. – 2004. – 318 с.

85.Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы / пер. с пол. И.Д. Рудинского. – 2-е изд., стер. – М.: Горячая линия–Телеком, 2013. – 384 с.

86.Рыжов А.П. Элементы теории нечётких множеств и её приложений: учебное пособие / А.П. Рыжов. – М.: Диалог-МГУ, 2003. – 81 с.

87.Свидетельство о государственной регистрации программы для ЭВМ №2017613946 Интеллектуальная система поддержки принятия решения на основе методов прецедентов в области IT-консалтинга/Авдеенко Т.В., Макарова Е.С.; заявитель и патентообладатель ФГБОУ Во «Новосибирский государственный технический университет». - №2017611083; заявл. 10.02.2017; зарегистрировано в реестре программ для ЭВМ 04.04.2017 г.

88.Свидетельство о государственной регистрации программы для ЭВМ №2017615280 Интеллектуальная система управления знаниями в IT-подразделении /Авдеенко Т.В., Макарова Е.С.; заявитель и патентообладатель ФГБОУ ВО «Новосибирский государственный технический университет». - №2017612409; заявл. 24.03.2017; зарегистрировано в реестре программ для ЭВМ 11.05.2017 г.

89.Сердюков К.Е. Исследование возможностей генетического алгоритма для извлечения релевантных прецедентов в системах поддержки принятия решений / К.Е. Сердюков, Т.В. Авдеенко, Е.С. Макарова // Информационные технологии и нанотехнологии (ИТНТ-2017): сб. тр. 3 междунар. конф. и молодежной шк., к 75-летию Самарского университета, Самара, 25–27 апр. 2017 г. – Самара: Изд-во Новая техника, 2017. – с. 1872–1878.

90.Система Service Desk Intraservice// URL: <http://www.itsmonline.ru/software/intraservice/> (дата обращения: 25.08.2017).

91.Система Service Desk Итилиум// URL: <https://itilium.ru/itilium/detail.php?ID=349> (дата обращения: 25.08.2017).

92.Системы Helpdesk (Service Desk)// URL: <http://www.itsmonline.ru/helpdesk/> - (дата обращения: 25.08.2017).

93.Скородумова О.Б. Новые потребности информационного общества и задачи управления // Инновационная наука. 2016. №2-5 (14). URL: <http://cyberleninka.ru/article/n/novye-potrebnosti-informatsionnogo-obschestva-i-zadachi-upravleniya> (дата обращения: 01.04.2016).

94.Теслинова Е.А. Разработка онтологии системы управления знаниями организации с использованием методологии концептуального проектирования // Успехи современного естествознания. – 2006. – № 9 – с. 96-98.

95.Томских М.С. Управление знаниями как основа обеспечения качества жизни // Учёные записки ЗабГУ. Серия: Философия, социология, культурология, социальная работа. 2016. №3. URL: <http://cyberleninka.ru/article/n/upravlenie-znaniyami-kak-osnova-obespecheniya-kachestva-zhizni> (дата обращения: 15.02.2016).

96.Топчий В.В. Управление знаниями – одна из основных концепций управления в сфере высшего образования // Царскосельские чтения. 2016. №XX. URL: <http://cyberleninka.ru/article/n/upravlenie-znaniyami-odna-iz-osnovnyh-kontseptsiy-upravleniya-v-sfere-vysshego-obrazovaniya> (дата обращения: 01.02.2017).

97. Тузовский А.Ф. Интеллектуальное пространство в системах управления знаниями / А.Ф. Тузовский, В.З. Ямпольский // Известия высших учебных заведений. Физика. – 2004. – № 7. – с. 23–29.

98. Тузовский А.Ф. Построение базы знаний организации с использованием системы онтологий / А.Ф. Тузовский, С.В. Козлов // Интеллектуальные системы (INTELS-2006): Труды Седьмого Международного симпозиума – Краснодар, 26–30 июня 2006. – М.: Русаки, 2006. – с. 290–294.

99. Тузовский, А.Ф. Построение модели знаний организации с использованием системы онтологий / А.Ф. Тузовский, С.В. Козлов // Компьютерная лингвистика и интеллектуальные технологии: Труды Международной конференции «Диалог 2006» – Бекасово, 31 мая – 4 июня 2006. – М.: Изд-во РГГУ, 2006. – с. 508–512.

100. Холл Р.Х. Организации: структура, процессы, результаты: учеб. пособие / Р.Х. Холл; пер. с англ. - СПб.: Питер, 2001. – 532 с.

101. Шкарупета Е.В. Организационно-экономический механизм управления знаниями в социально-экономических системах // Автореф. диссертации на соиск уч. степ канд. эк. наук, Воронеж, 2008.

102. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику. – URL: <http://matlab.exponenta.ru/fuzzylogic/book1/> (дата обращения: 21.02.2017).

103. Экономика и управление: мир необъятного Монография для студентов технических и экономических специальностей, / Сост.: Долгов Д. И.– 2014. – 521 с.

104. Юдин В.Н., Карпов Л.Е., Ватазин А.В. Методы интеллектуального анализа данных и вывода по прецедентам в программной системе поддержки врачебных решений, М., Альманах клинической медицины, 2008, т.17, часть 1. – с. 266-269.

105. Aamodt A., Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications 7(1), 1994, pp. 39-59.

106. Aamodt A. Towards robust expert systems that learn from experience—an architectural framework. In: J Boose, B Gaines, J-G Ganascia (eds.), EKAU-89; Third European Knowledge Acquisition for Knowledge-Based Systems Workshop, pp. 311-326.
107. Aamodt A. A knowledge intensive approach to problem solving and sustained learning. Ph.D dissertation, University of Trondheim, Norwegian Institute of Technology, May.
108. Aguirre-Pemran J, Belren Dıaz-Agudo B, Jimenez-Dıaz G. Searching Museum Routes Using CBR, Proceedings of the 24th International Conference on Case-Based Reasoning (ICCBR 2016) 2016, pp.1-15.
109. Ahmed M. U., Begum S., Funk P., Xiong N., Scheele B. V. “Case-based reasoning for diagnosis of stress using enhanced cosine and fuzzy similarity,” Transactions on Case-Based Reasoning for Multimedia Data, vol. 1, no. 1, 2008, pp. 3–19.
110. Alexander J.H., Freiling M.J., Shulman S.J., Staley J.L., Rehfuss S., Messick S.L. Knowledge level engineering: ontological analysis. AAAI-86 21, 1986, pp. 963-968.
111. Althoff K.D. "Knowledge acquisition in the domain of CBC machine centres: the MOLTKE approach". In: EKAU-89, Third European Workshop on Knowledge-Based Systems, J Boos, B Gainé and JG Ganascia (eds.), pp. 180-195.
112. An A, Cercone N. ELEM2: A Learning System for More Accurate Classifications, Proc. 12th Biennial Conf. Canadian Soc. Computational Studies of Intelligence, AI '98, Vancouver, B.C., Canada, 1998.
113. Avdeenko T. Acquisition of knowledge in the form of fuzzy rules for cases classification / T.V. Avdeenko, E.S. Makarova // Lecture Notes in Computer Science - LNCS, 2017. - Vol. 10387. – pp. 536-544.
114. Avdeenko T.V. Integration of case-based and rule-based reasoning through fuzzy inference in decision support systems / T.V. Avdeenko, E.S. Makarova // Procedia Computer Science. - 2017. – Vol. 103. – pp. 447–453.

115. Avdeenko T.V. The case-based decision support system in the field of IT-consulting / T.V. Avdeenko, E.S. Makarova // Journal of Physics: Conference Series. - 2017. - Vol. 803. – Art. 012008 (6 p.).

116. Avdeenko T.V, Makarova E.S, Klavsuts I.L. Artificial Intelligence Support of Knowledge Transformation in Knowledge Management Systems // 13 International scientific technical conference on actual on actual problems of electronic instrument engineering (APEIE):proc., Novosibirsk, 3-6 October 2016. – Novosibirsk: NSTU, 2016. – Vol. 1. – pp. 195-200.

117. Ayzenshtadt V., Langenhan C., Roith J. Comparative Evaluation of Rule-Based and Case-Based Retrieval Coordination for Search of Architectural Building Designs, Proceedings of the 24th International Conference on Case-Based Reasoning (ICCBR 2016) 2016, pp.16-31.

118. Belecheanu R., Pawar K.S., Barson R.J., Bredehorst B., Weber F. (2003). The application of case based reasoning to decision support in new product development. Integrated Manufacturing Systems, 14(1), pp. 36–45.

119. Bench-Capon T.M., Coenen F. The maintenance of legal knowledge based systems. AI Review, 1992, pp. 129-143.

120. Bonzano A., Cunningham P., Smyth B. Using introspective learning to improve retrieval in CBR: A case study in air traffic control, ICCBR'97, Rhode Island, USA, July 1997.

121. Branting K. Exploiting the complementarity of rules and precedents with reciprocity and fairness. In: Proceedings from the Case-Based Reasoning Workshop 1991, Washington DC, May 1991. Sponsored by DARPA. Morgan Kaufmann, 1991, pp. 39-50.

122. Brooke, Sand Jackson C. "Advances in elicitation by exception". In: Proc. istSGESInt. Workshop on Knowledge Based Systems Methodologies, pp. 70-78, British Computer Society SGES.

123. Chandrasekaran B. Design problem solving: a task analysis. AI Magazine Winter, 1990, pp. 59-73.

124. Chandrasekaran B. Generic tasks in knowledge-based reasoning: high level building blocks for expert system design". *IEEE Expert* 1, 1986, pp. 23-30.
125. Chang C.L. Using case-based reasoning to diagnostic screening of children with developmental delay. *Expert Systems with Applications*, 28, pp. 237–247.
126. Changchien S.W., Lin, M.C. Design and implementation of a case-based reasoning system for marketing plans. *Expert Systems with Applications*, 28, pp. 43–53.
127. Cheng Y.S.J., Cheng P.S. Case-based reasoning system for predicting Yarn Tenacity. *Textile Research Journal*, 74(8), pp. 718–722.
128. Choobineh, J., Lo A.W. Should rule-based reasoning be enhanced by case-based reasoning for conceptual database design? A theory and an experiment. *The Journal of Computer Information Systems*, 46(2), pp. 69–77.
129. Choy K.L., Fan, K.H., Lo V. Development of an intelligent customer–supplier relationship management system: The application of case-based reasoning. *Industrial Management and Data Systems*, 103(3/4), pp. 263–274.
130. Choy K.L., Lee W.B. A generic tool for the selection and management of supplier relationships in an outsourced manufacturing environment: The application of case based reasoning. *Logistics Information Management*, 15(4), pp. 235–253.
131. Cirovic G., Cekic, Z. Case-based reasoning model applied as a decision support for construction projects. *Kybernetool*, 31(5/6), pp. 896–908.
132. Coyle L., Cunningham P.: Improving recommendation ranking by learning personal feature weights. In: Funk, P., González Calero, P.A. (eds.) *ECCBR 2004*. LNCS (LNAI), vol. 3155, pp. 560–572.
133. Diaper D. *Knowledge Elicitation: Principles, Techniques and Applications*. Ellis Horwood, 1989.
134. Ding Y., Wang Y.M., Liu M.X. The public service oriented analysis of China's urban e-government development. *E-government*, 2014, 01 pp.111-117.
135. Dong R, Smyth B. Personalized Opinion-Based Recommendation, *Proceedings of the 24th International Conference on Case-Based Reasoning (ICCBR 2016)* 2016, pp. 93-107.

136. Dubois D., Prade H. Fuzzy sets and systems theory and applications. – London: Academic Press, 1980. – 411 p.
137. Fischer I, Poland J. Amplifying the Block Matrix Structure for Spectral Clustering. Telecommunications Lab. 2005.
138. Gentner D. Structure mapping - a theoretical framework for analogy. Cognitive Science, 1983, - Vol.7. pp. 155- 170.
139. GLPI система Helpdesk (Service Desk)// URL:<http://glpi-project.org/spip.php?article44> (дата обращения: 24.08.2017).
140. Golobardes E., Llorca X., Salamo, M., Marti, J. Computer aided diagnosis with case-based reasoning and genetic algorithms. Knowledge-Based Systems, 15, pp. 45–52.
141. Gruber T.R. A translation approach to portable ontologies / T.R. Gruber // Knowledge Acquisition. 1993. – № 5(2). – pp. 199-220. – URL: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html> (дата обращения: 21.02.2017).
142. Gruber T.R. Ontolingua: A Mechanism to Support Portable Ontologies, Technical Report KSL pp. 91-66, 1992. Knowledge Systems Laboratory, Stanford University.
143. Gruber T.R. Ontology to appear in the Encyclopedia of Database Systems / Ling Liu and M. Tamer Özsu (Eds.). – Springer-Verlag, 2008. – URL: http://tomgruber.org/writing/ontology_definition-2007.htm (дата обращения: 26.01.2017).
144. Guarino N. Some Ontological Principles for Designing Upper Level Lexical Resources // Proceedings of First International Conference on Language Resources and Evaluation. Granada, Spain, 1998.
145. Guiu J. M., Ribe E.G., Mansilla E.B., Fabrega X.L. Automatic diagnosis with genetic algorithms and case-based reasoning. Artificial Intelligence in Engineering, 13, pp. 367–372.
146. Gwen G.M. Applying case-based reasoning to KM. Information Today, 20(4), 60 p.

147. Hammond K.J. CHEF: A model of case-based planning. In: Proc. American Association for Artificial Intelligence, AAAI-86, Philadelphia, PA, 1986.
148. Hinrichs T.R. Problem Solving in Open Worlds. Lawrence Erlbaum, 1992.
149. Juan Y.K., Shin S.G., Perng Y.H. Decision support for housing customization: A hybrid approach using case-based reasoning and genetic algorithm. *Expert Systems with Application*, 31, pp. 83–93.
150. Kablan A. Adaptive neuro fuzzy inference systems for high frequency financial trading and forecasting // Third International Conference on Advanced Engineering Computing and Applications in Sciences. – Los Alamitos, California: IEEE Computer Society, 2009. – pp. 105–110.
151. Keane M. "Where's the beef? The absence of pragmatic factors in theories of analogy". In: ECAI-88 pp. 327-332.
152. Kitano H. "Challenges for massive parallelism". In: Proc. 13th. Conference on Artificial Intelligence, UCAI-93, pp. 813-834.
153. Koton P. Using experience in learning and problem solving. Ph. D thesis, Massachusetts Institute Technology, Laboratory of Computer Science, 1989.
154. Lim G., Ahn H., Lee H. Formulating strategies for stakeholder management: A case-based reasoning approach. *Expert Systems with Applications*, 28, pp. 831–840.
155. Liu X.W, Hu K.J. DSS for E-government project approval based on ontology and CBR, *J. of Computer Applications*, 2009, 29 (3): pp. 896-899.
156. Makarova E.S. A case-based reasoning approach with fuzzy linguistic rules: accuracy validation and application in interface design-support intelligent system / E.S. Makarova, T.V. Avdeenko, M.A. Bakaev // 11 International forum on strategic technology (IFOST 2016): proc., Novosibirsk, 1–3 June 2016. – Novosibirsk: NSTU, 2016. – Vol. 1. – pp. 505-509.
157. Marling C., Sqalli M., Rissland E., Hector M.A., Aha D. Case-based reasoning integrations. *AI Magazine*, 23(1), pp. 69–86.
158. Marques V., Farinha J.T., Brito A. Case-based reasoning and fuzzy logic in fault diagnosis // *WSEAS Transactions on Computers*. – 2009. – Vol. 8. – pp. 1408–1417.

159. Massie S., Craw S., Wiratunga N. "When similar cases don't have similar solutions," In: Proc. 7th Int. Conf. Case-Based Reasoning, Belfast, Northern Ireland, 2007. – pp.92-106.
160. Maxwell J.C. On Gavornos. Proceedings of the Royal Society of London. Vol.16. – 1868 p.
161. Moore C.J., Lehane M.S., Proce C.J. "Case-based reasoning for decision support in engineering design". In: Proc. IEE Colloquium on Case-Based Reasoning: Prospects for Applications. Digest No: 1994/057, pp. 4/1-4/4.
162. Motta E., Rajan T., Eisenstadt M. A methodology and tool for knowledge acquisition in KEATS-2. In: G Guida and C Tasso (eds.), Topics in Expert System Design: Methodologies and Tools, pp. 297-322.
163. Nilsson M., Sollenborn M. "Advancements and Trends in medical case-based reasoning: An overview of systems and system development," in Proc. 17th Int. FLAIRS Conf., 2004. – pp. 178–183.
164. Nirenburg S., Raskin V. Ontological Semantics. Cambridge, MA, 2004.
165. Nonaka I., Takeuchi H. The knowledge creating company: how Japanese companies create the dynamics of innovation, New York: Oxford University Press, 1995. – 284 p.
166. Noy N., McGuinness D.L. Ontology Development 101: A Guide to Creating Your First Ontology. // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.
//URL:http://protege.stanford.edu/publications/ontology_development/ontology101.html
167. Oberle D. Semantic Management of Middleware, Springer, Berlin, 2006.
168. OWL, язык веб-онтологий. Краткий обзор. 1994-2016. URL: <http://www.thalion.kiev.ua/idx.php/7/009/article/#s1.2> (Дата обращения: 15.03.2016).
169. Oxman R.E., 1993a. "PRECEDENTS: Memory structure in design case libraries". In: CAAD Futures 93.

170. Oxman R.E. 1993b. "Case-based design support: Supporting architectural composition through precedent libraries". *Journal of Architectural Planning Research*.
171. Quinlan J. R. *Induction of Decision Trees*. *Machine Learning* 1: 81-106, Kluwer Academic Publishers, 1986.
172. Richter A.M., Weiss S. "Similarity, uncertainty and case-based reasoning in PATDEX". In: RS Boyer (ed.), *Automated Reasoning, Essays in Honour of Woody Bledsoe*. Kluwer, pp. 249-265.
173. Rissland E. Examples in legal reasoning: Legal hypotheticals. In: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence, IJCAI, Karlsruhe*.
174. Schank R.C. *Dynamic Memory: A theory of reminding and learning in computers and people*. Cambridge University Press, 1982.
175. Schank R.C., Abelson R.P. *Scripts, Plans, Goals and Understanding*. Erlbau, 1977.
176. Shanavas N., Asokan S. *Ontology-Based Document Mining System for IT Support Service // Procedia Computer Science. International Conference on Information and Communication Technologies (ICICT 2014) 2015*. – pp. 329–336.
177. Shin K.S., Han I. Case-based reasoning supported by genetic algorithms for corporate bond rating. *Expert Systems with Application*, 16/ – pp. 85–95.
178. Simpson R.L. *A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation*. Technical Report GIT-ICS-85/18, Georgia Institute of Technology, School of Information and Computer Science.
179. Sycara E.P. *Resolving adversarial conflicts: An approach to Integrating Case-Based and Analytic Method* Technical Report GIT-ICS-87/26, Georgia Institute of Technology, School of Information and Computer Science, 1987.
180. Ushold M., Gruninger M. *Ontologies: Principles, Methods and Applications*. In: *Knowledge Engineering Review*, Vol. 11. №2. June 1996

181. Venkatamaran S., Krishnan R., Rao K.K. "A rule-case based system for image analysis". In: Proc 1st. European Workshop on Case-Based Reasoning, Posters & Presentations 2. – pp. 410-415.
182. vsDesk Российская help desk / service desk система// URL: https://vsdesk.ru/main/features#main_settings (дата обращения: 24.08.2017).
183. Watson I.D., Basden A., Brandon P.S. The client centred approach: expert system development. Expert Systems 9(4), 1992. – pp. 181-188.
184. Watson I.D., Abdullah S. "Developing case-based reasoning systems: a case study in diagnosing building defects". In: Proc. IEE Colloquium on Case-Based Reasoning: Prospects for Applications. Digest No: 1994/057. – pp. 1/1-1/3.
185. Wielinga B.J., Schreiber A.Th., Breuker J.A. KADS: A modelling approach to knowledge engineering. Knowledge Acquisition 4(1), 1992.
186. Wittgenstein L. Philosophical investigations. Blackwell. – pp. 31-34.
187. Xiong C-c, Wang L-t, Tao X., Deng Y. E-government Decision Support System Based on Case-based Reasoning and Ontology, 12th International Conference on Fuzzy Systems and Knowledge Discovery, 2015. – pp. 869-873.
188. Xiong N. Fuzzy rule-based similarity model enables learning from small case bases // Applied Soft Computing. – 2013. – Vol. 13. – pp. 2057–2064.
189. Xiong N. Learning fuzzy rules for similarity assessment in case-based reasoning // Expert Systems and Applications. – 2011. – Vol. 38. – pp. 10780–10786.
190. Xiong N., Funk P. Construction of fuzzy knowledge bases incorporating feature selection // Soft Computing. – 2006. – Vol. 10. – pp. 796–804.
191. Yang S., Robertson D. "A case-based reasoning system for regulatory information". In: Proc. IEE Colloquium on Case-Based Reasoning: Prospects for Applications. Digest No: 1994/057. – pp. 3/1-3/3.
192. Yu J. E-commerce recommendation system based on ontology and CBR. Yangtze University: Natural Science, 2013. – pp. 33-36.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

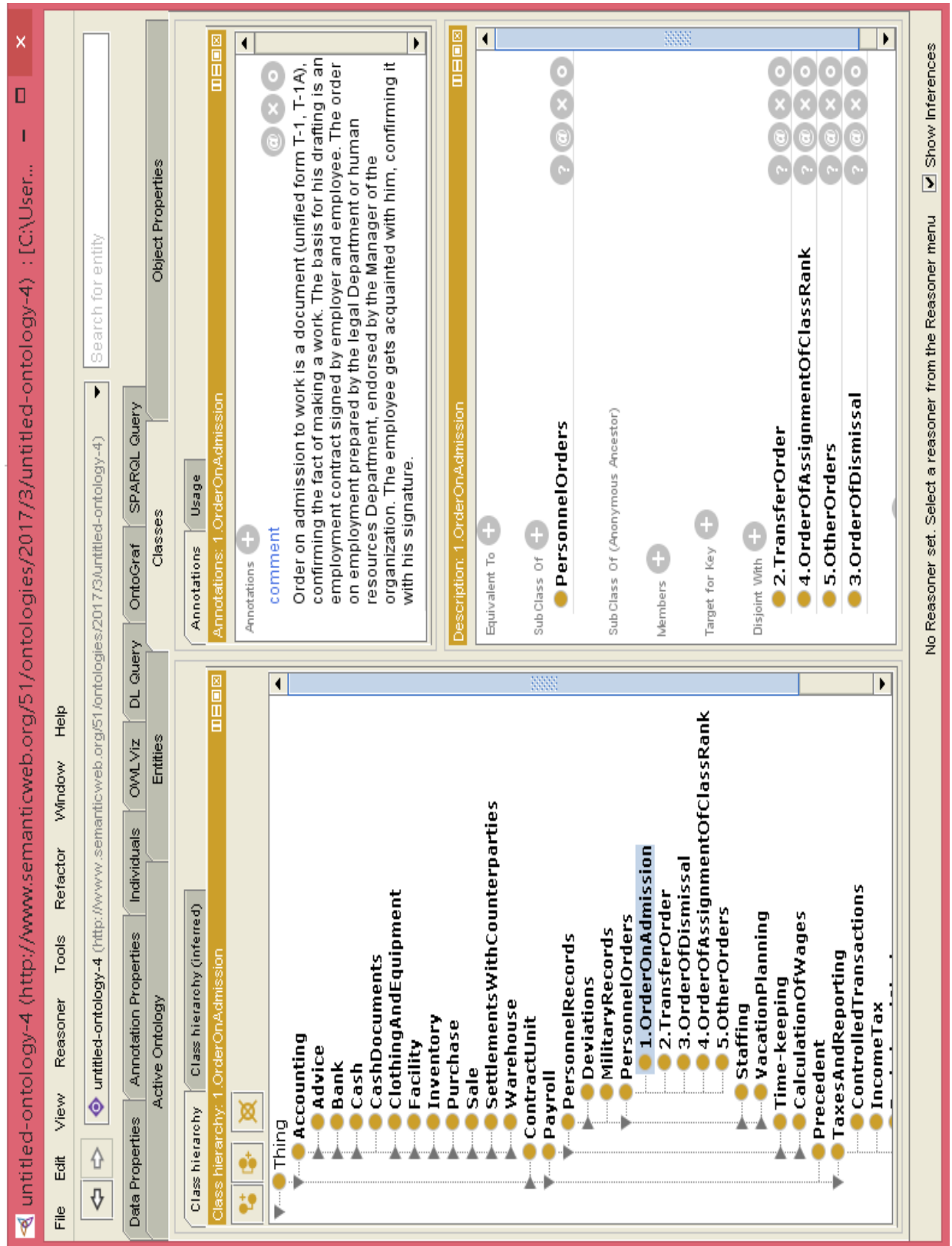


Рис. А.1 – Классы онтологии

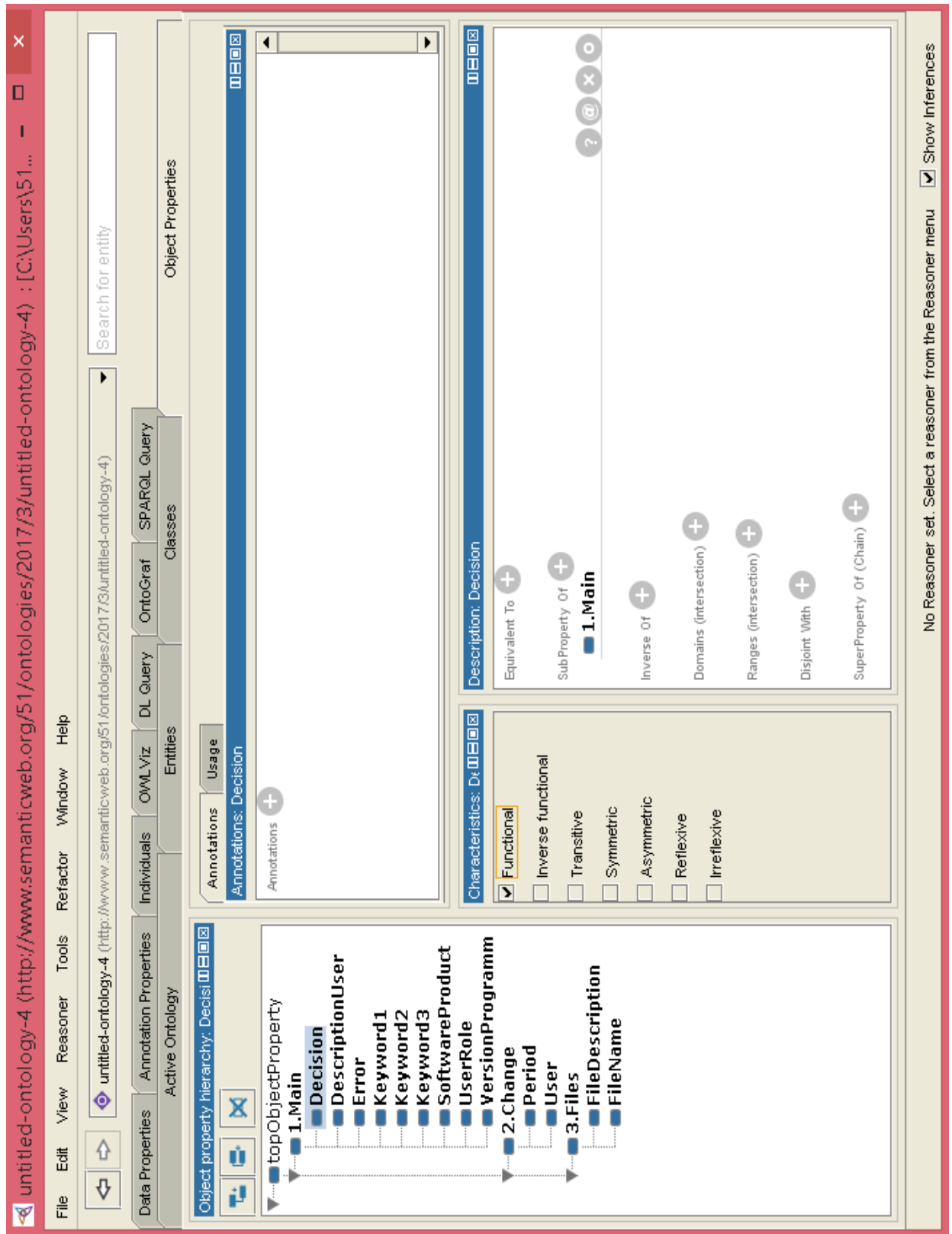


Рис. А.2 – Панель характеристик свойств

untitled-ontology-4 (http://www.semanticweb.org/51/ontologies/2017/3/untitled-ontology-4) : [C:\Users\51\Downloads\Онтология понятия...]

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-4 (http://www.semanticweb.org/51/ontologies/2017/3/untitled-ontology-4)

Active Ontology: Class hierarchy (Inferred)
Class hierarchy: 1.OrderOnAdmission

Class hierarchy (Inferred)

Class hierarchy: 1.OrderOnAdmission

Thing

- Accounting
- ContractUnit
- Payroll
- PersonnelRecords
- Deviations
- MilitaryRecords
- PersonnelOrders
- 1.OrderOnAdmission
- 2.TransferOrder
- 3.OrderOfDismissal
- 4.OrderOfAssignmentOfClassRank
- 5.OtherOrders
- Staffing
- VacationPlanning
- Time-keeping
- Timesheet
- WorkSchedules
- CalculationOfWages
- Precedent
- TaxesAndReporting
- ControlledTransactions
- IncomeTax
- Period-endClosing
- PropertyTaxes
- RecordKeeping
- RegulatedReports
- ValueAddedTax

Annotations

Annotations: A_return_to_his_former_place_of_work_during_

Usage

Usage: A_return_to_his_former_place_of_work_during_

Found 10 uses of A_return_to_his_former_place_of_work_during_

- A_return_to_his_former_place_of_work_during_
- A_return_to_his_former_place_of_work_during_
- A_return_to_his_former_place_of_work_during_
- A_return_to_his_former_place_of_work_during_
- A_return_to_his_former_place_of_work_during_
- Individual: A_return_to_his_former_place_of_work_during_
- A_return_to_his_former_place_of_work_during_
- A_return_to_his_former_place_of_work_during_
- A_return_to_his_former_place_of_work_during_
- A_return_to_his_former_place_of_work_during_

Property assertions: A_return_to_his_former_place_of_work_during_

Object property assertions

- Error Rate_staffing_busy
- DescriptionUser
- Returning_to_his_former_place_of_work_due_to_a_temporary_relocation

Data property assertions

- SoftwareVersion "Salary and personnel management 3.0"
- UserRole "Personnel officer"
- SoftwareProduct "1C"

Negative object property assertions

Negative data property assertions

Individuals: A_return_to_his_former_place_of_work_during_

Types

- Keyword1 some 2.TransferOrder
- Keyword2 some 3.StaffingSpecification
- Keyword3 some 5.OtherOrders

Same Individual As

Different Individuals

No Reasoner set. Select a reasoner from the Reasoner menu

Show Inferences

Рис. А.3 – Панель описания индивида 1

The screenshot displays the 'Individual Description' panel of an ontology editor. The interface includes a top navigation bar with tabs for 'Active Ontology', 'Entities', 'Classes', 'Object Properties', 'Data Properties', 'Annotation Properties', 'Individuals', 'OWL Viz', 'OntoGraf', 'Ontology Differences', and 'SPARQL Query'. The main panel is divided into several sections:

- Class hierarchy (inferred):** A tree view showing the hierarchy of classes. The root is 'Thing', which branches into 'Accounting', 'Advice', 'Bank', 'Cash', 'Inventory', 'Payroll', and 'Precedent'. Each of these further branches into more specific classes, such as '1. According To The Calculations', '2. In Business Inventories', '3. On Fixed Assets', '4. Other', '1. Payment Order', '2. Payment Requests', '3. Bank Statement', '4. Receipt To Account', '5. Writing Off From Account', '6. Recovery Of Payment', '7. Analysis Of Payments', 'Cash', 'Cash Documents', 'Clothing And Equipment', 'Facility', 'Inventory', 'Purchase', 'Sale', 'Settlements With Counterparties', '1. Acts Of Reconciliation Of Accounts', '2. Acts Of The Inventory Calculations', '3. Documents Of Payments To Contractors', '4. Adjustment Of Duty', 'Warehouse', 'Contract Unit', and 'Payroll'.
- Annotations:** A search bar and a list of annotations. The search bar contains the text 'Annotations: How_to_make_sick_leave_and_maternity_leave?'. The list below it shows several annotations, including '6. Error The_employee_fell_ill_before_the_holiday_how_to_enter_sick_leave?' and '2. DescriptionOfTheUser The_intersection_of_sick_and_vacation_how_to_correct'.
- Property assertions:** A search bar and a list of property assertions. The search bar contains the text 'Property assertions: How_to_make_sick_leave_and_maternity_leave?'. The list below it shows several property assertions, including '6. Error The_employee_fell_ill_before_the_holiday_how_to_enter_sick_leave?' and '2. DescriptionOfTheUser The_intersection_of_sick_and_vacation_how_to_correct'.
- Types:** A search bar and a list of types. The search bar contains the text 'Types: How_to_make_sick_leave_and_maternity_leave?'. The list below it shows several types, including 'Keywords some 3. Sick Leave', 'Keywords some 5. Vacation', and 'Keywords some CalculationOfPaidAbsences'.
- Negative assertions:** A search bar and a list of negative assertions. The search bar contains the text 'Negative assertions: How_to_make_sick_leave_and_maternity_leave?'. The list below it shows several negative assertions, including 'Negative object property assertions' and 'Negative data property assertions'.

Рис. А.4 – Панель описания индивида 2

ПРИЛОЖЕНИЕ Б

Обучающая выборка: 90; Тестовая выборка: 60

Таблица Б.1. Точность классификации прецедентов для разного количества функций принадлежности

Номер испытания	До модификации модели			После модификации модели		
	3 ФП	5 ФП	7 ФП	3 ФП	5 ФП	7 ФП
1	0,90	0,83	0,83	0,98	0,90	0,90
2	0,87	0,80	0,63	0,95	0,93	0,78
3	0,97	0,77	0,73	0,98	0,82	0,83
4	0,80	0,63	0,63	0,98	0,85	0,80
5	0,83	0,73	0,57	0,97	0,90	0,77
6	0,87	0,83	0,60	0,97	0,92	0,82
7	0,80	0,70	0,63	0,97	0,90	0,83
8	0,90	0,77	0,70	0,98	0,88	0,78
9	0,93	0,83	0,73	0,97	0,87	0,83
10	0,87	0,67	0,80	0,98	0,88	0,83
Среднее	0,88	0,76	0,69	0,973	0,885	0,817

Обучающая выборка: 90; Тестовая выборка: 60; Количество ФП: 3

Таблица Б.2. Точность классификации прецедентов для разного типа функций принадлежности

Номер испытания	До модификации модели		После модификации модели	
	Класс Т	Класс S	Класс Т	Класс S
1	0,90	0,97	0,98	1,00
2	0,87	0,97	0,98	1,00
3	0,93	0,93	1,00	0,97
4	0,87	0,97	1,00	0,98
5	0,87	0,83	1,00	0,97
6	0,85	0,92	0,98	1,00
7	0,87	0,88	0,98	0,95
8	0,82	0,90	0,98	0,93
9	0,87	0,87	1,00	1,00
10	0,85	0,90	0,98	0,97
Среднее	0,87	0,91	0,98	0,97

Таблица Б.3. Точность классификации при сокращении обучающей выборки на 15 прецедентов на каждом шаге разные типы ФП

Номер испытаний	Искомый размер ОВ	До модификации модели		После модификации модели	
		Класс Т	Класс S	Класс Т	Класс S
1	90	0,90	0,97	0,98	1,00
2	90	0,87	0,97	0,98	1,00
3	90	0,93	0,93	1,00	0,97
4	90	0,87	0,97	1,00	0,98
5	90	0,87	0,83	1,00	0,97
6	90	0,85	0,92	0,98	1,00
7	90	0,87	0,88	0,98	0,95
8	90	0,82	0,90	0,98	0,93
9	90	0,87	0,87	1,00	1,00
10	90	0,85	0,90	0,98	0,97
Среднее	90	0,87	0,91	0,98	0,97
11	60	0,84	0,92	0,88	0,93
12	60	0,83	0,90	0,92	0,94
13	60	0,88	0,91	0,98	0,90
14	60	0,88	0,87	0,98	0,92
15	60	0,86	0,90	0,97	0,91
16	60	0,86	0,92	0,99	0,92
17	60	0,86	0,83	0,90	0,89
18	60	0,88	0,89	0,94	0,97
19	60	0,86	0,90	0,94	0,94
20	60	0,87	0,89	0,91	0,92
Среднее	60	0,86	0,89	0,94	0,92
21	45	0,94	0,85	0,91	0,86
22	45	0,76	0,86	0,93	0,90
23	45	0,84	0,85	0,93	0,90
24	45	0,91	0,85	0,89	0,91
25	45	0,88	0,90	0,94	0,90
26	45	0,86	0,91	0,88	0,86
27	45	0,86	0,89	0,90	0,96
28	45	0,85	0,84	0,84	0,93
29	45	0,93	0,86	0,93	0,83
30	45	0,87	0,88	0,92	0,90
Среднее	45	0,87	0,87	0,91	0,90
31	30	0,83	0,88	0,84	0,94
32	30	0,69	0,67	0,89	0,81
33	30	0,83	0,81	0,90	0,87
34	30	0,68	0,83	0,89	0,87
35	30	0,79	0,82	0,94	0,91
36	30	0,79	0,84	0,89	0,86
37	30	0,72	0,83	0,91	0,86
38	30	0,87	0,80	0,84	0,88
39	30	0,82	0,86	0,91	0,78
40	30	0,85	0,86	0,90	0,83
Среднее	30	0,79	0,82	0,89	0,86

Таблица Б.4. Точность классификации при сокращении ОВ на 15 прецедентов на каждом шаге для разного количества ФП модиф. модель

Номер испытания	Искомый размер ОВ	Точность классификации					
		3 ФП		5 ФП		7 ФП	
		Класс Т	Класс S	Класс Т	Класс S	Класс Т	Класс S
1	90	0,98	0,95	0,98	1,00	0,97	0,98
2	90	0,98	0,97	0,98	1,00	1,00	0,98
3	90	1,00	0,97	0,97	1,00	1,00	0,98
4	90	1,00	0,97	0,98	1,00	0,98	0,98
5	90	1,00	1,00	0,95	1,00	0,98	1,00
6	90	0,98	0,98	1,00	1,00	0,97	1,00
7	90	0,98	0,97	0,98	0,98	1,00	0,98
8	90	1,00	0,98	1,00	1,00	0,97	0,98
9	90	1,00	0,98	0,98	1,00	0,97	0,98
10	90	1,00	0,98	0,98	1,00	0,98	1,00
Среднее	90	0,99	0,97	0,98	0,99	0,98	0,99
11	60	0,95	0,97	0,97	0,98	0,87	0,98
12	60	1,00	1,00	0,90	0,98	0,92	1,00
13	60	1,00	0,98	0,97	1,00	0,90	0,97
14	60	1,00	0,98	0,95	0,98	0,95	0,98
15	60	0,95	0,97	0,93	1,00	0,93	1,00
16	60	1,00	0,97	0,98	0,98	0,90	1,00
17	60	1,00	0,98	0,93	1,00	0,92	0,98
18	60	0,98	0,98	0,97	1,00	0,85	1,00
19	60	1,00	0,98	0,92	1,00	0,90	0,98
20	60	1,00	0,92	0,97	0,98	0,88	0,97
Среднее	60	0,99	0,97	0,95	0,99	0,90	0,98
21	45	0,98	0,98	0,92	0,97	0,78	0,98
22	45	0,98	0,95	0,95	0,87	0,92	1,00
23	45	0,92	0,93	0,85	0,98	0,90	1,00
24	45	0,98	0,95	0,90	1,00	0,82	1,00
25	45	1,00	0,93	0,93	0,97	0,83	0,92
26	45	0,98	1,00	0,97	0,98	0,90	1,00
27	45	0,98	0,98	0,95	0,98	0,78	0,98
28	45	0,97	0,90	0,93	0,98	0,97	1,00
29	45	0,97	0,97	0,93	0,95	0,92	0,98
30	45	1,00	0,97	0,88	1,00	0,90	0,98
Среднее	45	0,97	0,96	0,92	0,97	0,87	0,98
31	30	0,95	0,97	0,88	1,00	0,80	1,00
32	30	0,93	0,97	0,90	0,93	0,83	0,88
33	30	0,93	0,98	0,93	1,00	0,78	0,95
34	30	0,88	0,93	0,92	1,00	0,78	1,00
35	30	0,98	0,98	0,85	1,00	0,78	0,98
36	30	0,93	0,97	0,90	0,92	0,78	1,00
37	30	1,00	0,92	0,93	0,93	0,82	0,95
38	30	1,00	1,00	0,92	0,95	0,80	0,98
39	30	0,98	0,97	0,88	0,95	0,78	0,97
40	30	1,00	0,95	0,90	0,97	0,83	1,00
Среднее	30	0,96	0,96	0,90	0,97	0,80	0,97

Таблица Б.5. Точность на НД «Ирис» при
ОВ 105 прецедентов

№ испытаний	ОВ	Точность классификации на контрольной выборке	
		ФП типа T	ФП типа S
1	105	0,860	0,920
2	105	0,910	0,930
3	105	0,810	0,900
4	105	0,900	0,920
5	105	0,830	0,900
6	105	0,920	0,950
7	105	0,880	0,920
8	105	0,920	0,950
9	105	0,900	0,970
10	105	0,850	0,920
11	105	0,870	0,950
12	105	0,870	0,850
13	105	0,870	0,920
14	105	0,880	0,920
15	105	0,920	0,880
16	105	0,880	0,930
17	105	0,920	0,950
18	105	0,920	0,950
19	105	0,920	0,950
20	105	0,870	0,950
21	105	0,930	0,930
22	105	0,870	0,950
23	105	0,920	0,930
24	105	0,870	0,930
25	105	0,920	0,920
Среднее	105	0,888	0,928

Таблица Б.6. Точность на НД «Ирисы» при
ОВ 90 прецедентов

№ испытаний	ОВ	Точность классификации на контрольной выборке	
		ФП типа T	ФП типа S
1	90	0,860	0,920
2	90	0,910	0,930
3	90	0,810	0,890
4	90	0,820	0,880
5	90	0,830	0,800
6	90	0,900	0,950
7	90	0,880	0,820
8	90	0,920	0,950
9	90	0,800	0,970
10	90	0,850	0,920
11	90	0,830	0,900
12	90	0,850	0,850
13	90	0,870	0,920
14	90	0,880	0,920
15	90	0,920	0,880
16	90	0,880	0,830
17	90	0,820	0,900
18	90	0,900	0,880
19	90	0,900	0,900
20	90	0,870	0,950
21	90	0,930	0,930
22	90	0,870	0,920
23	90	0,920	0,930
24	90	0,870	0,830
25	90	0,830	0,880
Среднее	90	0,868	0,898

Таблица Б.7. Точность на НД «Ирис» при
ОВ 75 прецедентов

№ испытаний	ОВ	Точность классификации на контрольной выборке	
		ФП типа T	ФП типа S
1	75	0,860	0,910
2	75	0,880	0,880
3	75	0,930	0,890
4	75	0,860	0,890
5	75	0,890	0,960
6	75	0,790	0,910
7	75	0,860	0,860
8	75	0,880	0,880
9	75	0,860	0,870
10	75	0,830	0,880
11	75	0,890	0,870
12	75	0,920	0,910
13	75	0,830	0,910
14	75	0,820	0,910
15	75	0,810	0,940
16	75	0,920	0,860
17	75	0,880	0,820
18	75	0,830	0,870
19	75	0,840	0,890
20	75	0,880	0,830
21	75	0,890	0,910
22	75	0,880	0,890
23	75	0,880	0,910
24	75	0,900	0,870
25	75	0,860	0,890
Среднее	75	0,867	0,888

Таблица Б.8. Точность на НД «Ирисы»
при ОВ 60 прецедентов

№ испытаний	ОВ	Точность классификации на контрольной выборке	
		ФП типа T	ФП типа S
1	60	0,930	0,880
2	60	0,860	0,910
3	60	0,880	0,910
4	60	0,880	0,870
5	60	0,880	0,870
6	60	0,920	0,940
7	60	0,860	0,860
8	60	0,790	0,820
9	60	0,860	0,870
10	60	0,840	0,960
11	60	0,920	0,880
12	60	0,880	0,890
13	60	0,830	0,890
14	60	0,840	0,900
15	60	0,890	0,900
16	60	0,920	0,720
17	60	0,830	0,870
18	60	0,820	0,880
19	60	0,810	0,890
20	60	0,810	0,900
21	60	0,880	0,810
22	60	0,880	0,890
23	60	0,890	0,890
24	60	0,860	0,830
25	60	0,900	0,910
Среднее	60	0,866	0,878

Таблица Б.9. Точность на НД «Ирис» при
ОВ 45 прецедентов

№ испытаний	ОВ	Точность классификации на контрольной выборке	
		ФП типа T	ФП типа S
1	45	0,940	0,850
2	45	0,760	0,860
3	45	0,840	0,850
4	45	0,910	0,850
5	45	0,880	0,900
6	45	0,730	0,810
7	45	0,930	0,880
8	45	0,870	0,790
9	45	0,880	0,860
10	45	0,920	0,880
11	45	0,900	0,820
12	45	0,860	0,830
13	45	0,850	0,820
14	45	0,920	0,890
15	45	0,870	0,840
16	45	0,930	0,930
17	45	0,840	0,870
18	45	0,800	0,850
19	45	0,840	0,880
20	45	0,830	0,900
21	45	0,870	0,890
22	45	0,880	0,890
23	45	0,820	0,880
24	45	0,820	0,890
25	45	0,810	0,840
Среднее	45	0,860	0,862

Таблица Б.10. Точность на НД «Ирисы»
при ОВ 30 прецедентов

№ испытаний	ОВ	Точность классификации на контрольной выборке	
		ФП типа T	ФП типа S
1	30	0,880	0,900
2	30	0,830	0,880
3	30	0,690	0,670
4	30	0,830	0,810
5	30	0,680	0,830
6	30	0,790	0,820
7	30	0,750	0,830
8	30	0,810	0,780
9	30	0,760	0,780
10	30	0,900	0,790
11	30	0,780	0,710
12	30	0,800	0,770
13	30	0,800	0,860
14	30	0,800	0,870
15	30	0,830	0,770
16	30	0,810	0,820
17	30	0,840	0,800
18	30	0,750	0,680
19	30	0,660	0,870
20	30	0,750	0,870
21	30	0,910	0,850
22	30	0,770	0,820
23	30	0,820	0,820
24	30	0,670	0,830
25	30	0,810	0,870
Среднее	30	0,788	0,812

ПРИЛОЖЕНИЕ В

Таблица В.1. Количественные оценки семантических связей прецедентов с терминальными концептами онтологии (фрагмент)

[illegible]

Таблица В.2. Таблица правил (для ЗФП, и Т-типа ФП)

Receiving	Transfer	Dismissal	Vacation	Sick	Report card	Report	Prepayment	Calculation of basic	One-time accrual	Average earnings	Retention	Payment	Recalculation	2-NDFL	6-NDFL	Deductions employees	Insurance contributions	Other taxes	Entry	Class
A1	B1	C1	D2	E2	F2	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D1	E1	F2	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D2	E2	F1	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O2	P2	Q1	R1	S1	T1	Deviations
A1	B1	C1	D1	E2	F2	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D2	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D1	E1	F1	G2	H1	I1	J1	K2	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D2	E1	F2	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D2	E1	F2	G1	H1	I2	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K3	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D3	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D2	E1	F1	G1	H1	I1	J1	K1	L1	M3	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D3	E1	F2	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K2	L1	M2	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C2	D2	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Deviations
A1	B1	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O3	P1	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O2	P2	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L2	M1	N1	O2	P2	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P2	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O2	P2	Q1	R1	S1	T1	NDFL

Продолжение таблицы В.2

A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M2	N1	O2	P2	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G1	H2	I1	J1	K1	L1	M1	N1	O2	P2	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N2	O3	P1	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G1	H1	I1	J2	K1	L1	M1	N1	O2	P2	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O2	P1	Q1	R2	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P3	Q1	R1	S1	T1	NDFL
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L2	M3	N1	O1	P1	Q1	R1	S1	T1	Payroll
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K2	L1	M1	N3	O1	P1	Q1	R1	S1	T1	Payroll
A1	B1	C1	D1	E1	F1	G1	H1	I2	J2	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T3	Payroll
A1	B1	C1	D1	E1	F1	G1	H1	I2	J1	K1	L1	M3	N1	O1	P1	Q1	R1	S1	T1	Payroll
A1	B1	C1	D1	E1	F1	G1	H1	I2	J1	K1	L3	M1	N1	O1	P1	Q1	R1	S1	T1	Payroll
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L3	M1	N2	O1	P1	Q1	R1	S1	T1	Payroll
A1	B1	C1	D1	E1	F1	G1	H3	I1	J1	K1	L1	M2	N1	O1	P1	Q1	R1	S1	T1	Payroll
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L3	M2	N1	O1	P1	Q1	R1	S1	T1	Payroll
A1	B1	C1	D1	E1	F1	G2	H1	I1	J3	K1	L1	M1	N2	O1	P1	Q1	R1	S1	T1	Payroll
A1	B1	C1	D1	E1	F1	G1	H1	I1	J3	K2	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Payroll
A2	B1	C1	D1	E1	F1	G3	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	PersonnelOrders
A1	B1	C1	D1	E1	F1	G3	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	PersonnelOrders
A3	B1	C1	D1	E1	F2	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	PersonnelOrders
A3	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	PersonnelOrders
A1	B2	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	PersonnelOrders
A2	B1	C1	D1	E1	F1	G1	H1	I3	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	PersonnelOrders
A1	B1	C1	D1	E1	F1	G1	H3	I1	J1	K1	L1	M2	N1	O1	P1	Q1	R1	S1	T1	PersonnelOrders
A2	B1	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	PersonnelOrders
A1	B1	C1	D1	E2	F1	G2	H1	I1	J1	K2	L1	M1	N1	O1	P1	Q1	R1	S1	T1	SickLeave
A1	B1	C1	D1	E2	F1	G1	H1	I1	J1	K2	L1	M1	N1	O1	P1	Q1	R1	S1	T1	SickLeave
A1	B1	C1	D1	E3	F2	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	SickLeave
A1	B1	C1	D1	E2	F1	G1	H1	I1	J1	K1	L1	M1	N1	O2	P2	Q1	R1	S1	T1	SickLeave

Продолжение таблицы В.2

A1	B1	C1	D1	E3	F1	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	SickLeave
A1	B1	C1	D1	E2	F1	G1	H1	I1	J1	K1	L1	M3	N1	O1	P1	Q1	R1	S1	T1	SickLeave
A1	B1	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R2	S2	T1	TaxesAndContributions
A1	B1	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S3	T1	TaxesAndContributions
A1	B1	C1	D1	E1	F1	G1	H1	I2	J2	K1	L1	M1	N1	O1	P1	Q1	R2	S1	T1	TaxesAndContributions
A1	B1	C1	D1	E1	F1	G1	H1	I2	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T3	TaxesAndContributions
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N2	O1	P1	Q1	R3	S1	T1	TaxesAndContributions
A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R2	S2	T1	TaxesAndContributions
A1	B1	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T3	TaxesAndContributions
A1	B1	C1	D1	E1	F2	G1	H1	I3	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Time-keeping
A1	B1	C1	D1	E1	F2	G1	H3	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Time-keeping
A1	B1	C1	D1	E1	F3	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Time-keeping
A1	B1	C1	D1	E1	F2	G1	H1	I1	J1	K3	L1	M1	N1	O1	P1	Q1	R1	S1	T1	Time-keeping
A1	B1	C1	D1	E1	F2	G1	H1	I1	J1	K1	L1	M1	N3	O1	P1	Q1	R1	S1	T1	Time-keeping
A1	B3	C1	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	TransferOrder
A1	B3	C1	D1	E1	F2	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	TransferOrder
A1	B2	C1	D1	E1	F2	G1	H1	I2	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	TransferOrder
A1	B2	C1	D1	E1	F1	G3	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	TransferOrder
A1	B1	C3	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	OrderOfDismissal
A1	B1	C2	D1	E1	F1	G2	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	OrderOfDismissal
A1	B1	C1	D1	E1	F1	G2	H1	I2	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1	OrderOfDismissal

ПРИЛОЖЕНИЕ Г

Свидетельства о государственной регистрации программы для ЭВМ и
акты внедрения об использовании результатов диссертации

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО
о государственной регистрации программы для ЭВМ
№ 2017613946

**Интеллектуальная система поддержки принятия решений
на основе метода прецедентов в области IT-консалтинга**

Правообладатель: *Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Новосибирский государственный технический университет»
(RU)*

Авторы: *Макарова Екатерина Сергеевна (RU),
Авдеенко Татьяна Владимировна (RU)*

Заявка № **2017611083**
Дата поступления **10 февраля 2017 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ **04 апреля 2017 г.**

Руководитель Федеральной службы
по интеллектуальной собственности



Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2017615280

**Интеллектуальная система управления знаниями в ИТ
подразделении**

Правообладатель: *Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Новосибирский государственный технический университет»
(RU)*

Авторы: *Макарова Екатерина Сергеевна (RU),
Авдеенко Татьяна Владимировна (RU)*



Заявка № 2017612409

Дата поступления 24 марта 2017 г.

Дата государственной регистрации
в Реестре программ для ЭВМ 11 мая 2017 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев Г.П. Ивлиев

УТВЕРЖДАЮ

Начальник ЭУ № 054

(подпись)

В.Г. Галкин

(фамилия, имя, отчество)

2015 г.



Печать организации

АКТ

о внедрении результатов кандидатской диссертационной работы

Макаровой Екатерины Сергеевны

Настоящий акт подтверждает, что методы и алгоритмы, разработанные и исследованные в диссертационной работе аспирантки Макаровой Е.С. «Методы и алгоритмы интеллектуальной поддержки принятия управленческих решений на основе прецедентов» по специальности «05.13.10 – управление в социальных и экономических системах», использовались ООО «ПрофТехУслуги» ЭУ №054 для помощи в работе специалистов кадровой службы. Применение разработанных методов и алгоритмов позволило повысить эффективность при работе с программным продуктом на 15% (на основе экспертных оценок).

Начальник отдела кадров

Т.В. Хлопкова

УТВЕРЖДАЮ

Проректор по учебной работе

доктор технических наук, доцент

Брованов С.В.

13 марта 2017 г.



АКТ

**о внедрении результатов кандидатской диссертации
Макаровой Екатерины Сергеевны в учебный процесс НГТУ**

Настоящим актом подтверждается, что результаты диссертации Макаровой Е.С. по специальности 05.13.10 – управление в социальных и экономических системах, используются в учебном процессе кафедры экономической информатики Новосибирского государственного технического университета в 2015-2017 гг.

Результаты диссертационной работы (онтологическая модель интеллектуальной подсистемы поддержки принятия решений, база знаний, база прецедентов для эффективного консультирования пользователей) использованы при подготовке практических заданий по следующим дисциплинам «Системы поддержки принятия решений», «Инженерия знаний и интеллектуальные системы», «Интеллектуальные информационные системы», «Интеллектуальные технологии в моделировании бизнес-процессов».

Зав. кафедрой экономической
информатики, к.э.н., доцент

Мамонов В. И.

УТВЕРЖДАЮ



Директор филиала ФГУП «Охрана»
Росгвардии по Новосибирской области

С.Н. Потушинский

«01» августа 2017 г.

АКТ ОПЫТНОГО ВНЕДРЕНИЯ результатов диссертационной работы Макаровой Екатерины Сергеевны

В 2016-2017 годах в рамках инициативной работы была разработана, протестирована, и прошла опытную эксплуатацию подсистема интеллектуальной поддержки принятия решений на основе прецедентов. В процессе работы были получены результаты:

- разработана и накоплена база прецедентов для эффективного консультирования пользователей;
- разработан подход к извлечению прецедентов из базы знаний, базовым компонентом которого является индексирование на основе онтологии предметной области;
- разработана методика классификации прецедентов на основе теории нечетких множеств;
- разработана новая технология работы с заявками пользователей.

При создании данной подсистемы использовались методы, алгоритмы и программное обеспечение, разработанные аспиранткой Макаровой Е.С. в ходе проведения ее диссертационного исследования.

Опытная эксплуатация созданной подсистемы в ИТ-отделе филиала ФГУП «Охрана» Росгвардии по Новосибирской области улучшила работу отдела кадров, и сократила время на обработку запросов пользователей.

Применение разработанных методов и алгоритмов позволило повысить эффективность при работе с программным продуктом на 20 % (на основе экспертных оценок).

Начальник отдела кадров

С.Н. Никольская