

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

На правах рукописи

Субхангулов Руслан Айратович

**ОНТОЛОГИЧЕСКАЯ ИНФОРМАЦИОННАЯ
ПОДДЕРЖКА ПРОЕКТИРОВАНИЯ В
ЭЛЕКТРОННЫХ АРХИВАХ ТЕХНИЧЕСКОЙ
ДОКУМЕНТАЦИИ**

Специальность 05.13.12 – Системы автоматизации
проектирования (промышленность)

Диссертация на соискание ученой степени
кандидата технических наук

научный руководитель – кандидат технических наук, доцент
Наместников Алексей Михайлович

Ульяновск – 2015

Оглавление

Список сокращений	5
Введение	6
1. Анализ применимости интеллектуальных методов поддержки проектных решений в электронных архивах технической документации	14
1.1. Обзор методов информационной поддержки проектных решений в электронных архивах	16
1.1.1. Информационный поиск документов	16
1.1.2. Кластеризация документов	20
1.1.3. Классификации документов	21
1.2. Понятие информационной потребности в электронных архивах. Формальные модели информационной потребности в интеллектуальных системах	25
1.3. Понятие профиля пользователя	29
1.4. Применение прикладных онтологий в задачах информационной поддержки	30
1.5. Анализ существующих программных систем поддержки проектных решений в электронных архивах технических документов	40
1.6. Цели и задачи исследования	43
2. Онтологические модели и алгоритмы информационной поддержки процесса проектирования	45

2.1. Модель онтологии информационной поддержки процесса проектирования автоматизированных систем	46
2.1.1. Задачи прикладной онтологии при проектировании автоматизированных систем	46
2.1.2. Структурно-функциональная модель информационной поддержки проектировщика.....	50
2.1.3. Структурно-функциональная модель прикладной онтологии	52
2.2. Онтологическая модель профиля проектировщика	58
2.2.1. Формализация профиля проектировщика с учетом информационных потребностей.....	58
2.2.2. Способ формирования онтологического контекста проекта.....	61
2.3. Формирование нечетких контекстно-ориентированных запросов к электронному архиву	66
2.3.1. Классификация запросов на основе Байесовской модели	66
2.3.2. Нечеткая онтологическая модель поискового запроса	68
2.3.3. Алгоритм формирования нечетких контекстно-ориентированных запросов к электронному архиву	73
2.4. Методика интеллектуальной информационной поддержки проектирования АС	77
3. Разработка интеллектуальной системы информационной поддержки проектирования АС.....	81
3.1. Структурно-функциональное решение программной системы	81
3.2. Описание функциональных возможностей подсистем	83
3.2.1. Описание подсистемы хранения данных.....	83
3.2.2. Описание электронной библиотеки Wiki-ресурсов	87
3.3. Описание функциональных возможностей подсистемы информационной поддержки.....	88
3.4. Описание проекта программной системы.....	93
3.4.1. Иерархия классов.....	93
3.4.2. Основные алгоритмы программной системы.....	96

3.5. Выводы по главе	99
4. Анализ адекватности разработанных моделей и методов на основе вычислительных экспериментов	101
4.1. Формирование концептуальной сети проектов на основе wiki- ресурсов.....	104
4.2. Формирование пользовательских профилей системы информационной поддержки.....	105
4.3. Математическая модель оценки качества формирования поисковых запросов	108
4.4. Сравнительный анализ результатов вычислительных экспериментов на множестве документов электронного архива ФНПЦ АО «НПО «Марс» ..	109
4.5 Выводы по главе	123
Заключение	124
Библиографический список	126
Приложение А	140
Приложение Б.....	142
Онтология предметной области	142
Б.1 Схема онтологии	142
Б.2 Фрагмент онтологии предметной области	143
Приложение С	146
Исходные коды основных алгоритмов программной системы	146
С.1 Алгоритм формирования концептуальной сети из внешних профессиональных wiki-ресурсов.....	146
С.2 Алгоритм формирования контекстно-ориентированных проектных запросов.....	147
С.3 Алгоритм вычисления меры сходства между контекстно- ориентированным проектным запросом и ТД	150

Список сокращений

АС – автоматизированная система.

ЭА – электронный архив.

ЖЦ – жизненный цикл.

ИНС – искусственная нейронная сеть.

НИОКР – научно-исследовательские и опытно-конструкторские работы.

ОКР – опытно-конструкторские работы.

ОМ – онтологическая модель.

ОП – онтологическое представление.

САПР – система автоматизированного проектирования.

ТД – технический документ.

Введение

Современные электронные архивы (ЭА) крупных проектных организаций насчитывают десятки и сотни тысяч документов, относящихся к проектно-технической документации. В таких ЭА сосредоточен значительный опыт проектирования сложных технических систем, к которым можно отнести современные автоматизированные системы (АС). Проектирование новой системы крайне редко осуществляется без обращения к предыдущему опыту разработок и поиска аналогов. Такие исследователи, как Норенков И.П., Тарасов В.Б. Collins Н., отмечают тот факт, что при увеличении объема ЭА затрудняется анализ документов по заранее заданным реквизитам. Вследствие распределенного проектирования от участников проекта часто требуются навыки в области семантического анализа технической документации и выполнение корректных предметно-ориентированных проектных запросов к ЭА.

В основе модели информационной поддержки лежат механизмы выполнения интеллектуальных запросов, которые предполагают способность поисковой системы к самоорганизации, осуществление независимого общения с пользователем, эффективный поиск текстовых документов, реагирующий на изменения информационной потребности пользователя.

Учет специфики проектных знаний приводит к необходимости формирования онтологии электронного архива особой структуры, позволяющей учитывать контекст принимаемых проектных решений, жизненный цикл проектируемой АС и систему индивидуальных предпочтений проектировщиков. Таким образом, система информационной

поддержки, как неотъемлемая часть электронного архива проектной организации, должна обладать свойствами интеллектуальной системы. Известные исследователи в области онтологических систем, такие как Загоруйко Ю.А., Гаврилова Т.А., Соловьев В.Д., Лукашевич Н.В., Добров Б.В., Ландэ Д.В., Смирнов С.В., Соснин П.И., Gruber T.R., Berners-Lee T., Uschold M. и другие отмечают важность и актуальность исследований, базирующихся на онтологическом подходе в сложно-структурированных предметных областях. В работах Соснина П.И. и Смирнова С.В. отмечается важность применения систем, основанных на знании, в процессе человеко-компьютерного взаимодействия при нахождении проектных решений.

В настоящее время не существует математических методов и алгоритмов, позволяющих выполнять контекстно-ориентированные запросы к ЭА с учетом текущего состояния проекта и индивидуальных предпочтений проектировщика. Следовательно, актуальным является разработка моделей, методов и алгоритмов построения информационной поддержки, которые составляют теоретическую основу интеллектуальной системы формирования запросов к ЭА технической документации с применением предметно-ориентированной онтологий с учетом принципиальной неполноты средств языка запросов.

Цель диссертационной работы

Целью диссертации является совершенствование процессов взаимодействия субъекта проектирования автоматизированных систем с электронным архивом технической документации, обеспечивающее повышение качества выполнения проектных запросов за счет включения в процесс информационной поддержки дополнительных предметных знаний и учета опыта проектировщика.

Предмет исследования

Модели, методы и средства информационной поддержки процесса проектирования АС при анализе и поиске технической документации в электронном архиве.

Объект исследования

Объектом исследования является электронный архив технической документации крупной проектной организации.

Задачи диссертационного исследования

В соответствии с целью работы актуальными являются следующие задачи исследования:

- Провести сравнительный анализ существующих современных методов, алгоритмов и систем обеспечения доступа к массиву текстовой документации. Рассмотреть их ограничение в контексте проектирования АС. Исследовать возможность и оценить целесообразность применения прикладной онтологии в задачах информационной поддержки проектирования АС.

- Разработать структурно-функциональную модель онтологии информационной поддержки автоматизированного проектирования АС, способствующую интеграции знаний из внешних профессиональных ресурсов.

- Разработать онтологическую модель профиля проектировщика для обеспечения возможности сохранения опыта взаимодействия субъекта проектирования с электронным архивом.

- Разработать алгоритм формирования контекстно-ориентированных запросов к электронному архиву технических документов с целью улучшения качества информационной поддержки проектирования АС.

- Разработать необходимые программные средства, позволяющие решать задачу информационной поддержки в процессе проектирования АС на

основе онтологии, провести вычислительные эксперименты, доказывающие их эффективность, внедрить полученные результаты в практику проектной организации.

Методы исследования

В диссертационной работе применяются методы онтологического анализа, теории графов, теории нечетких систем, теории вероятностей и объектно-ориентированного программирования.

Научная новизна

Научная новизна результатов исследования заключается в следующем:

1. Предложена новая структурно-функциональная модель онтологии информационной поддержки автоматизированного проектирования АС, отличающаяся многоуровневой структурой и позволяющая выполнять проектные запросы в контексте этапов жизненного цикла проектируемого изделия и учитывать текущий контекст проекта.

2. Разработана онтологическая модель профиля проектировщика, которая позволяет специфицировать опыт взаимодействия субъекта проектирования с электронным архивом на концептуальном уровне.

3. Разработан алгоритм формирования контекстно-ориентированных запросов к электронному архиву технических документов на основе байесовского классификатора с целью информационной поддержки деятельности проектировщика с учетом моделируемых информационных потребностей.

4. Предложена методика использования онтологических алгоритмов информационной поддержки в жизненном цикле проектирования АС, отличающаяся возможностью интеграции системы профилей субъектов проектирования, онтологического ресурса с электронным архивом технической документации с привлечением дополнительных знаний, извлеченных из внешних профессиональных wiki-ресурсов.

Практическая значимость работы

Созданная программная система информационной поддержки проектировщика применяется в процессе проектирования автоматизированных систем и позволяет достичь улучшенных технико-экономических показателей объектов проектирования за счет сокращения времени выполнения опытно-конструкторских работ.

Разработанные модели и алгоритмы реализованы в форме программной системы и внедрены в деятельность ФНПЦ АО «НПО «Марс» (г. Ульяновск). Практическое использование результатов диссертационной работы подтверждено соответствующими документами.

Основания для выполнения работы

Данная научная работа выполнялась в рамках тематического плана научных исследований Федерального агентства по образованию в 2010 году, была поддержана грантами РФФИ № 10-07-00064-а в 2010, 2011 и 2012 годах, № 14-01-31086 мол_а в 2014 и 2015 годах.

Достоверность результатов диссертационной работы

Достоверность научных положений, выводов и рекомендаций подтверждена результатами математического моделирования, результатами экспериментов и испытаний, а также результатами использования материалов диссертации в проектных подразделениях организации.

Основные положения, выносимые на защиту:

1. Модель прикладной онтологии информационной поддержки проектирования АС является адекватной для выполнения проектных запросов в контексте этапов жизненного цикла проектируемого изделия и учитывает текущий контекст проекта.

2. Онтологическая модель профиля проектировщика является достаточной для адекватного представления опыта взаимодействия субъекта проектирования с электронным архивом на концептуальном уровне.

3. Алгоритм формирования контекстно-ориентированных запросов к электронному архиву технических документов с целью информационной поддержки проектировщика является эффективным по критериям точности и полноты.

4. Разработанный комплекс программ как подсистема информационной поддержки пользователя электронного архива технической документации в полной мере реализует все описанные теоретические положения и позволяет сократить время выполнения поисковых запросов к электронному архиву технических документов.

Апробация работы

Основные положения и результаты диссертации докладывались, обсуждались и получили одобрение на всероссийской школе-семинаре «ИМАП-2011» (г. Ульяновск, 2011 г.); на молодежной научно-технической конференции «Автоматизация процессов управления» (г. Ульяновск, 2011 г.); на 46-й научно-технической конференции УлГТУ (г. Ульяновск, 2012 г.); на 4-й всероссийской научно-технической конференции аспирантов, студентов и молодых ученых «ИВТ-2012» (г. Ульяновск, 2012 г.); на 2-м международном симпозиуме «Гибридные и синергетические интеллектуальные системы: теория и практика» (г. Калининград, 2012 г.); на 13-й национальной конференции по искусственному интеллекту с международным участием «КИИ-2012» (г. Белгород, 2012 г.); на всероссийской школе-семинаре «ИМАП-2012» (г. Ульяновск, 2012 г.); на 47-й научно-технической конференции УлГТУ (г. Ульяновск, 2013 г.); на III международной научно-технической конференции «OSTIS-2013» (г. Минск, 2013 г.); на VII международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном

интеллекте» (г. Коломна, 2013 г.); на IV международной научно-технической конференции «OSTIS-2014» (г. Минск, 2014 г.); на VI всероссийской научно-практической конференции «НСМВ-2014» (г. Санкт-Петербург, 2014); на 3-м международном симпозиуме «Гибридные и синергетические интеллектуальные системы: теория и практика» (г. Калининград, 2014 г.); на 13-й национальной конференции по искусственному интеллекту с международным участием «КИИ-2014» (г. Казань, 2014 г.); на V международной научно-технической конференции «OSTIS-2015» (г. Минск, 2015 г.); на VIII международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (г. Коломна, 2015 г.).

Научные публикации

По результатам работы было опубликовано 21 статья, из которых 6 – в журналах из перечня ВАК, и 3 тезиса докладов. Получено свидетельство (РОСПАТЕНТ) о государственной регистрации программ для ЭВМ №2012617587.

Структура и объем диссертации

Диссертация состоит из введения, четырех глав, заключения, списка использованной литературы и приложений. Основное содержание работ изложено на 152 страницах, включая 45 рисунков и 8 таблиц. Список использованных источников состоит из 121 наименований.

В **первой главе** содержится анализ понятия информационной потребности пользователя электронного архива, представлены способы формализации информационной потребности в интеллектуальных системах. Рассматриваются методы информационной поддержки пользователя для удовлетворения его информационных потребностей в процессе принятия проектных решений. Приведен обзор методов информационного поиска и классификации технических документов электронного архива. Рассмотрено

понятие «онтология», представлены формальные модели онтологии и способы применения ее в задачах информационного поиска и классификации документов. Анализируются существующие программные системы поддержки проектных решений в электронных архивах технических документов.

Во **второй главе** рассмотрено формализованное представление информационной потребности. Описывается структурно-функциональная модель онтологии для решения задачи информационной поддержки в электронном архиве технической документации с учетом жизненного цикла автоматизированного проектирования автоматизированных систем (АС). Представлены методы информационной поддержки проектировщика на основе алгоритмов формирования контекстно-ориентированных запросов к электронному архиву технической документации. Рассмотрен алгоритм онтологической модели информационного поиска, учитывающий информационную потребность пользователя электронного архива и жизненный цикл автоматизированного проектирования АС. Описана методика интеллектуальной информационной поддержки проектирования АС.

В **третьей главе** рассмотрена архитектура и функциональные возможности программной системы, реализующей информационную поддержку автоматизированного проектирования на основе онтологии в электронном архиве технической документации. Описаны структуры входных и выходных данных, программное обеспечение и технологические средства.

В **четвертой главе** проводится анализ адекватности разработанных моделей и методов на основе вычислительных экспериментов.

Личный вклад

Все результаты, составляющие содержание диссертации, получены автором самостоятельно.

Глава 1

Анализ применимости интеллектуальных методов поддержки проектных решений в электронных архивах технической документации

Современные проектные предприятия обладают значительным объемом конструкторско-технологической документации. Одна и та же конструкторская документация может использоваться многократно в различных проектах и одна и та же технологическая документация может быть адаптирована к различным производственным условиям. Таким образом, возникает необходимость в интеграции результатов работ, полученных в ходе профессиональной деятельности проектных отделов, в единую систему для того, чтобы, во-первых, упорядочить знания о документах и выстроить иерархию документов, во-вторых, решить многочисленные практические задачи с документами для их эффективного поиска, использования и хранения. Поэтому в настоящее время многие предприятия переводят накопленный архив документации в электронный формат. В связи с этим появилась потребность в систематизации и автоматизации работы с электронным архивом технической документации.

Единицей обработки и хранения в электронном архиве является технический документ, который рассматривается как информационный ресурс. Информационный ресурс – это файл или совокупность файлов, объединенных общей семантикой и имеющих текстовую аннотацию [69]. Основными функциями электронного архива являются [49]:

- управление документами и иерархической структурой архива;
- оцифровка, трансформация и представление бумажных документов в разных форматах;
- ускорение занесения большого массива типовых и разнородных документов в систему;
- управление Web-контентом;
- управление задачами и отслеживание статуса их выполнения;
- быстрый и удобный поиск документов.

Информационная поддержка в электронном архиве – процесс информационного обеспечения, ориентированный на пользователей информации, занятых в процессе проектировании сложных объектов. Задачей информационной поддержки электронного архива является максимальное удовлетворение информационной потребности специалистов-проектировщиков. Основными методами, с помощью которых достигается удовлетворение информационной потребности, являются:

- информационный поиск (поиск документов по запросу пользователя);
- автоматическая кластеризация документов;
- автоматическая классификация документов.

Для эффективного поиска документов в электронных архивах применяются различные методы предобработки [3]: удаление стоп-слов, стемминг (морфологический анализ), N-граммы, приведение регистра, извлечение ключевых понятий из текста. Данные методы применяются для снижения времени поиска информации и устранения незначущих слов [3]. Совокупность характеристик моделей документов, запросов и функций соответствия запроса документу называется моделью информационного поиска.

1.1. Обзор методов информационной поддержки проектных решений в электронных архивах

1.1.1. Информационный поиск документов

Самым известным методом информационной поддержки пользователя является информационный поиск документов. Будем использовать следующее определение информационного поиска, как процесса поиска в большой коллекции (хранящейся, как правило, в памяти компьютера) некоего неструктурированного материала (обычно – документа), удовлетворяющего информационные потребности [29]. Под неструктурированными данными следует понимать набор документов, представляющих собой логически объединенный текст без каких-либо ограничений на его структуру. В научной литературе используется понятие классической модели поиска, которая включает:

- булеву модель поиска;
- векторно-пространственную модель поиска;
- вероятностную модель поиска.

Перед тем, как рассмотреть классические модели информационного поиска, введем ряд обозначений, которые характерны для них: T – множество термов, D – множество документов, по которым выполняется поиск информации. Каждый документ представляется как набор ключевых слов, называемый индексом. Обычно индексы документа представляются, как множество $D = \{t_i, w_i\}$, где t_i – терм из множества термов T , w_i – вес терма в документе, характеризующий частоту встречаемости терма в документе. Чаще всего термами являются существительные, т. к. они описывают сами себя и, следовательно, позволяют без труда семантически идентифицироваться. Прилагательные, наречия, глаголы менее полезны для индексов документа, т. к. служат обычно дополнениями к существительным. Информационный запрос q так же, как и документ представляется в виде множества термов с весами $q = \{t_i, w_i\}$.

Модель булева поиска

Одна из известных моделей информационного поиска базируется на теории множеств и Булевой алгебре. В работе [29] дается следующее определение модели булевого поиска – это модель информационного поиска, в ходе которого можно обрабатывать любой запрос, имеющий вид булева выражения, т. е. выражения, в котором термины используются в сочетании с операциями AND, OR и NOT. Данная модель основывается на том, что терм либо присутствует в документе, либо отсутствует. Как результат, вес термина в документе имеет бинарное значение $w_i \in \{0,1\}$.

Булева модель поиска имеет следующие недостатки [26], [29], [104], [111].

- На заданный запрос поисковая машина может вернуть очень много документов (или даже все документы коллекции). В этом случае пользователь вынужден последовательно добавлять условия в запрос, чтобы уменьшить результирующую выборку. Поиск производится методом проб и ошибок. В результате часто возникает ситуация, когда условия булевого запроса оказываются противоречивы и пользователь не получает ни одного документа.
- Как правило, полезную выборку обозримого размера можно получить, задав сложную логическую формулу. При этом от пользователя требуется не только знание правил построения формул, но и достаточно хорошее знакомство с «языком» предметной области.
- Вследствие того, что существуют только два значения релевантности: «релевантен» (true) и «нерелевантен» (false), результирующая выборка не может быть упорядочена по релевантности. Все документы одинаково релевантны. Данная проблема решается в расширенной булевой модели поиска [110].
- Все атомы формулы имеют одинаковую важность (вес), хотя некоторые из них могут быть «ключевыми», другие – вспомогательными.

Векторная модель поиска

Наиболее популярной моделью поиска является векторная модель поиска (Vector Model Space) [9], [12], [79], [107], [108], [109]. Векторная модель была реализована в 1968 году Джерардом Солтоном (Gerard Salton) в поисковой системе SMART (Salton's Magical Automatic Retriever of Text). В данной модели поиска документ представляется «мешком слов» (bag of words). Порядок следования слов в документе не является значимым. Предполагается, что документы с одинаковыми наборами слов сходны. В данной модели каждому терму в документе t_i соответствует вес w_i , который характеризует частоту появления терма в документе. Запрос q также представляется как множество термов со своими весами. Таким образом, запрос и документ представляются, n -мерными векторами $\vec{q} = (w_{1,q}, w_{2,q}, \dots, w_{n,q})$ и $\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{n,j})$, где n – количество термов в словаре модели.

Существует множество подходов к определению весов терма, но большинство из них исходит из двух эмпирических наблюдений, справедливых для текстов [29]:

- 1) Чем чаще встречается слово в тексте, тем оно более релевантно по отношению к теме документа.
- 2) Чем чаще встречается слово среди всех документов коллекции, тем хуже оно отражает различие между документами.

Для нахождения меры сходства между документом d_j и пользовательским запросом q используется простое скалярное произведение двух векторов $\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{n,j})$ и $\vec{q} = (w_{1,q}, w_{2,q}, \dots, w_{n,q})$, которое соответствует косинусу угла между векторами.

Недостатками векторной модели поиска являются следующие [26], [29], [54]:

- данная модель не справляется с синонимией (когда разные слова имеют одно значение) и полисемией (когда одно слово имеет разные значения);
- векторы и массивы имеют высокую размерность, что приводит к сложности обработки.

Вероятностная модель поиска

Вероятностная модель поиска опирается на теоретические подходы байесовских условных вероятностей. В 1977 году С. Э. Робертсон (S.E. Robertson) и К. Спарк-Джонс (K. Sparck Jones) обосновали и реализовали вероятностную модель, предложенную в 1960 году [26], [105], [106].

Основной вопрос, который решается с помощью модели: как велика вероятность того, что документ d релевантен запросу q [26]? Релевантность при этом рассматривается как вероятность того, что данный документ может оказаться интересным пользователю. Функционирование модели базируется как на экспертных оценках, получаемых в результате обучения модели, которые признают документы из учебной коллекции релевантными/нерелевантными, так и на последующих оценках вероятности того, что документ является релевантным запросу, исходя из состава его термов.

Если для запроса известны данные оценки вероятностей для всех документов, то документы можно сортировать по ним и выводить пользователям в отсортированном порядке. Другими словами, вероятностная модель поиска предусматривает определение вероятностей соответствия запроса для документов, сортировку и предоставление документов с ненулевой вероятностью пользователю.

С самого начала в вероятностной модели использовалось упрощение, которое допускает независимость вхождения в документ любой пары термов (поэтому такой подход называется «наивным» байесовским).

При этом в вероятностной модели поиска предполагается наличие учебных наборов релевантных и нерелевантных документов, выбранных пользователем или полученных автоматически при каком-то начальном предположении. Вероятность того, что поступивший документ является релевантным, рассчитывается на основании соотношения появления термов в релевантном и нерелевантном массиве документов.

В случае применения экспертных оценок процесс поиска является итерационным. На каждом шаге итерации благодаря режиму обратной связи определяется множество документов, отмеченных пользователем как удовлетворяющие его информационным потребностям.

Недостатками данной модели являются:

- низкая вычислительная масштабируемость;
- необходимость постоянного обучения системы.

1.1.2. Кластеризация документов

Кластеризация – способ группировки многомерных объектов, основанный на представлении результатов отдельных наблюдений точками подходящего геометрического пространства с последующим выделением групп, как сгустков этих точек [28].

Для возможности применения кластеризации в задачах информационной поддержки используется кластерная гипотеза [29].

Кластерная гипотеза. Документы, принадлежащие одному и тому же кластеру, являются примерно одинаково релевантными по отношению к информационным потребностям.

Данная гипотеза утверждает, что если документ принадлежит кластеру и удовлетворяет информационным потребностям пользователей, то, возможно, и другие документы этого же кластера также удовлетворяют информационным потребностям.

Постановка задачи кластеризации документов выглядит следующим образом: дано множество документов $D = \{d_1, d_2, \dots, d_n\}$, конечное множество

кластеров K и целевая функция $\rho(d, d')$, которая определяется в терминах сходства или расстояния между документами. Задача алгоритма кластеризации состоит в вычислении величины $\gamma: D \rightarrow \{1, \dots, K\}$, минимизирующей (или максимизирующей) целевую функцию.

В различных алгоритмах используются различные критерия близости между документами. Чаще всего применяются следующие метрики [7], [17], [51], [71]:

- евклидово расстояние $d_2(x_i, x_j) = \sqrt{\sum_{t=1}^m (x_{it} - x_{jt})^2}$,
- расстояние по Хэммингу $d_H(x_i, x_j) = \sum_{t=1}^m |x_{it} - x_{jt}|$,
- расстояние Чебышева $d_\infty(x_i, x_j) = \max_{1 \leq t \leq m} |x_{it} - x_{jt}|$,
- расстояние Махаланобиса $d_M(x_i, x_j) = (x_i - x_j)^t S^{-1} (x_i - x_j)^t$.

1.1.3. Классификации документов

Задача классификации текстовых документов является частным случаем задачи классификации объектов по различным признакам, работы по которым начались в 60-х годах прошлого века [52], [68], [70]. В настоящее время разработаны различные методы классификации, которые применяются в широком круге задач [5], [6]. Современные ЭА содержат классификационные функционалы, которые позволяют распределять технические документы по заранее определенным классам (категориям). Задача категоризации текстовых документов заключается в установлении принадлежности документа к определенным классам. Принадлежность к классу определяется на основании принадлежности содержания текстовых документов к конкретному классу по заранее определенным признакам. К классификационным признакам можно отнести [29]:

- 1) общую тематику содержания текстов;
- 2) наличие определенных дескрипторов;

3) выполнение определенных сложных условий.

Формальное описание задачи классификации текстовых документов выглядит следующим образом: пусть имеется множество документов $D = \{d_1, \dots, d_n\}$ и фиксированное множество классов (категорий) $C = \{c_1, \dots, c_m\}$. Размерность множества классов определяется экспертами предметной области. Множество документов имеет большую размерность, чем множество категорий. Кроме того, задано обучающее множество D' документов, которые распределены по определенным классам, т.е. $\langle d, c \rangle \in D' \times C$, данное множество документов заранее подготовлено экспертами предметной области. Используя алгоритм обучения необходимо построить функцию классификации γ , которая отображает документы в классы:

$$\gamma : D \rightarrow C.$$

Коллекцию заранее классифицированных экспертами документов обычно разделяют на две коллекции [29]:

1. Учебная коллекция, по которой строится функция классификации γ ;
2. Тестовая коллекция, которая используется для проверки качества построенной функции классификации γ .

Рассмотрим подробно различные методы классификации текстовых документов, которые применяются в электронных архивах.

Классификация в векторном пространстве

В основе таких моделей классификации лежит векторное представление документов, в котором каждый документ рассматривается в виде вектора терминов с весами. В данных моделях используется следующая гипотеза [25]:

Гипотеза компактности. Документы, принадлежащие одному и тому же классу, образуют компактную область, причем области, соответствующие разным классам, не пересекаются.

К данной модели относятся методы классификации Роккио [89], [107] и К-ближайших соседей.

Метод опорных векторов

Метод опорных векторов (Support Vector Machines, SVM), предложенный советским ученым В. Н. Вапником [73], [91], [118], [119], относится к группе граничных методов классификации. В наиболее простом случае бинарной классификации задача сводится к нахождению гиперплоскости, разделяющей два множества объектов (документов): одно множество документов принадлежит категории, другое множество документов не принадлежит категории. При этом найдется такая гиперплоскость, для которой минимальное расстояние до ближайших документов максимальна.

Нахождение плоскости методом опорных векторов сводится к решению оптимизационной задачи [73]:

$$\begin{cases} \sum_i \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j (\vec{x}_i \bullet \vec{x}_j) \rightarrow \min_{\lambda}; \\ \sum_{i=1} \lambda_i y_i = 0; \\ C \geq \lambda_i \geq 0, i = 1, \dots, n \end{cases}$$

Для решения этой задачи разработаны различные методы [90].

Классификация на основе нейронных сетей

В основе данной модели лежит искусственная нейронная сеть (ИНС). ИНС – это сеть, состоящая из набора нейронов, соединенных между собой [86]. Интеллектуальные системы на основе ИНС позволяют решать задачи

распознавания образов, выполнения прогнозов, оптимизации ассоциативной памяти и управления [30], [86].

Нейронную сеть можно рассматривать как классификатор, где на вход поступают термины документа с весовыми значениями. Величины, полученные на выходе, будут являться результатами классификации. При этом часть нейронов отмечены как входные (к ним поступают документы), а часть как выходные, которые выдают результативную информацию. Для того чтобы данная система могла качественно функционировать, необходимо прежде «натренировать» ее на известных данных, в которых известна входная информация и правильный результат, получаемый на выходе ИНС. Тренировка состоит в подборе весов межнейронных связей, обеспечивающих наибольшую близость ответов сети к известным правильным ответам.

Байесовская модель классификации

В основе данного метода лежит математический аппарат теории вероятности [10]. Основная идея данного метода заключается в определении вероятности принадлежности анализируемого документа к конкретной категории. Для этих целей используется следующая формула:

$$P(c | d) = \frac{P(d | c)P(c)}{P(d)},$$

где $P(c | d)$ – условная вероятность того, что документ d принадлежит классу c , $P(d | c)$ – условная вероятность встретить документ d среди всех документов класса c , $P(c)$ – безусловная вероятность того, что документ принадлежит классу c , $P(d)$ – безусловная вероятность встретить документ d в коллекции документов.

Выдвигается предположение, что события (в нашем случае наборы слов) являются независимыми. Исходя из предположения о независимости слов, условная вероятность документа аппроксимируется произведением условных вероятностей всех слов, входящих в документ. В работе [91]

рассматриваются различные варианты формул для аппроксимации и их влияние на результат классификации.

1.2. Понятие информационной потребности в электронных архивах. Формальные модели информационной потребности в интеллектуальных системах

Информационная потребность – характеристика предметной области, значение которой необходимо установить для выполнения поставленной задачи в практической деятельности. Данное определение используется в серии стандартов ГОСТ 7.73-96.

Выделяются четыре этапа формирования информационной потребности:

- реальная информационная потребность – это неосознанная истинная информационная потребность пользователя (потребность в некоторой новой информации при решении стоящей перед пользователем задачи);
- осознанная информационная потребность – появляется после осознания пользователем стоящей перед ним проблемы (осознанная потребность отличается от реальной, более того, пользователь может понимать имеющуюся проблему неправильно);
- выраженная информационная потребность – результат описания осознанной информационной потребности с помощью естественного языка;
- формализованная информационная потребность – это результат представления выраженной потребности средствами формального поискового языка.

На рисунке 1.1 представлен процесс формирования информационной потребности [24]:



Рисунок 1.1 Формирование информационной потребности

Процесс формирования информационной потребности состоит из ряда этапов, где конечным является формализация потребности. Чаще всего информационная потребность формализуется в виде поискового запроса на естественном языке, который состоит из набора терминов предметной области. Формально такой запрос можно представить в следующем виде:

$$\bar{q} = \{t_i\}.$$

Данное представление информационной потребности характерно для традиционных моделей поиска. Однако такие модели содержат недостатки, которые мы рассматривали выше, отметим лишь наиболее важные из них [13]:

- избыточность — в пословном индексе используются слова-синонимы, выражающие одни и те же понятия;
- предположение о независимости слов текста — слова текста считаются независимыми друг от друга, что не соответствует свойствам связного текста;
- многозначность слов — поскольку многозначные слова могут рассматриваться как дизъюнкция двух или более понятий, выражающих различные значения многозначного слова, то маловероятно, что все элементы этой дизъюнкции интересуют пользователя.

Для решения подобного рода проблем применяются интеллектуальные модели поиска. Соответственно, формализованная информационная потребность претерпевает изменения. Рассмотрим методы, которые применяются в процессе формализации потребности в интеллектуальных моделях информационной поддержки [76]:

- обратная связь по релевантности;
- обратная связь по псевдорелевантности;
- расширение/новая формулировка запроса с помощью специального тезауруса.

Обратная связь по релевантности

Алгоритм с обратной связью выглядит следующим образом [76]:

1. Пользователь вводит запрос.
2. Система возвращает первоначальный список найденных документов.
3. Пользователь отмечает некоторые из найденных документов как релевантные или нерелевантные.
4. Система определяет улучшенное представление информационной потребности, основываясь на обратной связи с пользователем.
5. Система выводит пользователю уточненный набор найденных документов.

Данный алгоритм выполняется один раз или несколькими итерациями. Основная цель данного алгоритма состоит в переходе от запроса $\vec{q}$ к виду формализованного представления информационной потребности $\vec{q}_{раси}$ и нахождения документов, удовлетворяющих новому запросу. Запрос $\vec{q}_{раси}$ может вычисляться с помощью различных выражений, например [29]:

- Применение косинусной метрики $\vec{q}_{раси} = \frac{1}{|D_r|} \sum_{\vec{d} \in D} \vec{d}_r - \frac{1}{|D_{nr}|} \sum_{\vec{d} \in D} \vec{d}_{nr}$,
- Метод Роккио $\vec{q}_{раси} = \alpha \vec{q} + \beta \frac{1}{|D_r|} \sum_{\vec{d} \in D} \vec{d}_r - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d} \in D} \vec{d}_{nr}$.

Обратная связь по псевдорелевантности

Алгоритм с обратной связью по псевдорелевантностью [76]:

1. Пользователь вводит запрос.
2. Находится первоначальный список документов.
3. Предполагается, что первые k документы являются релевантными.
4. Затем к k документам применяется алгоритм с обратной связью по релевантности.

Расширение/новая формулировка запроса с помощью специального тезауруса

Уточнение информационной потребности достигается путем использования различных видов словарей и тезаурусов. Для каждого термина запроса предлагается дополнительный термин из тезауруса, который является синонимом, либо семантически близок к термину запроса. Таким образом, формализованная информационная потребность принимает новый вид и используется подсистемой информационной поддержки электронного архива в процессе поиска релевантных документов. Данные тезаурусы могут быть созданы вручную экспертами предметной области, либо сгенерированы

автоматически, путем анализа коллекции документов. Существует два основных метода автоматического создания тезауруса [29], [76]:

1. Используется совместная встречаемость слов. Считается, что, если слова встречаются в одном абзаце, то они имеют семантическую близость или связаны между собой, поэтому для этого случая применяется статистический подсчет текстовых показателей.

2. Подход, основанный на использовании грамматического анализа текста, а также грамматических отношений и зависимостей.

Преимущества методов, основанных на интеллектуальных моделях информационных потребностях – это более качественный результат поиска вследствие того, что система информационной поддержки обладает знаниями о реальной информационной потребности пользователя.

Недостатками данных методов являются следующие.

- Метод с обратной связью релевантности неудобен для пользователя, т.к. необходимо реагировать на запросы системы и выполнять действия для получения документов, удовлетворяющих информационной потребности.

- Метод расширения запроса с помощью тезауруса требует построения качественного лингвистического ресурса. Однако, любая предметная область развивается и, следовательно, всегда требуется пополнение тезауруса. Данная модель требует высокого качества разрешения многозначности слов. В работах [82], [112] обосновывается предположение, что точность многозначности терминов должна достигать не менее 90% для получения положительного эффекта в информационном поиске.

1.3. Понятие профиля пользователя

В современных аналитических системах при решении задач информационной поддержки активно используется понятие профиля пользователя. Профиль пользователя – совокупность данных, отражающих

информационную потребность пользователя. Системы, применяющие профили пользователей, получили название рекомендующих систем.

Рекомендующие системы – это класс систем принятия решений, которые используют знания об интересах и предпочтения пользователя для оценки/прогнозирования его реакции на рекомендации [11]. Отметим, что в ряде работ [81], [93], [94], [120] в качестве содержания профиля было предложено использовать понятия естественного языка. Данные методы получили название методов на основе понятий (англ. *concept-based methods*). В данных методах выдвигается гипотеза, что понятия отражают с одной стороны семантику запроса пользователя, а с другой – информационную потребность пользователя. При этом понятие сопровождается списком ключевых слов, которые семантически близки к данному понятию. Соответственно, поиск документов осуществляется по тем понятиям и ключевым словам профиля, которые содержатся в документах.

Профили пользователей находят широкое применение в задачах информационной поддержки. Дальнейшим развитием данных систем является применение в качестве содержания профилей базы знаний или прикладных онтологий.

1.4. Применение прикладных онтологий в задачах информационной поддержки

Для профессионального, в том числе научно-технического поиска информации, требуется обеспечение поиска, основанного на знаниях, – использование синонимов, возможности автоматического расширения запроса, возможности автоматического анализа результатов запроса и помощь в интерактивном поиске. Для решения подобного рода задач применяют интеллектуальные модели поиска, функционирование которых основано на предметно-ориентированных знаниях. Все более широкое распространение в научной сфере находят методы извлечения знаний из

слабоструктурированных документов, основанные на применении словарей, тезаурусов, онтологий [2].

Понятие онтологии имеет много различных определений [20], [25], [34], [70], [83], [84]. Неформально, онтология представляет собой некоторое описание взгляда на мир, применительно к конкретной области интересов. Это описание состоит из терминов и правил использования этих терминов, ограничивающих их значение в рамках конкретной области.

На формальном уровне, онтология – это система, состоящая из набора понятий и набора утверждений об этих понятиях, на основе которых можно строить классы, объекты, отношения, функции и теории.

В качестве определения онтологии может использоваться следующее: онтология – артефакт, структура, описывающая значения элементов некоторой системы [20], [21], [22]. Еще одно из определений онтологии, получившее широкое распространение в научных кругах и предложенная Томасом Грубером: онтология – явная спецификация концептуализации [83]. Это определение является обобщением множества различных определений. В данном определении центральное место занимает понятие концептуализации. Концептуализация – это структура реальности, рассматриваемая независимо от словаря предметной области и конкретной ситуации [14], [16], [56], [85].

В настоящее время не существует единого мнения о формальном представлении онтологии. В работе [8] онтологию записывают в следующем виде:

$$O = \langle X, R, F \rangle,$$

где X – конечное множество концептов (понятий, терминов) предметной области, которую представляет онтология O ; R – конечное множество отношений между концептами (понятиями, терминами) заданной предметной области; F – конечное множество функций интерпретации (аксиоматизации).

При наложении ограничений $R = \emptyset$ и $F = \emptyset$ онтология трансформируется в простой словарь:

$$O = V = \langle X, \{\}, \{\} \rangle.$$

В рамках проекта Karlsruhe Ontology Framework (KAON) [113] было сформировано следующее формальное представление онтологии:

Ядро онтологии с аксиомами это структура

$$O := (C, \leq_c, R, \sigma \leq R, A),$$

состоящая из

- двух непересекающихся множеств C и R , элементы которых есть *идентификаторы понятий* и *идентификаторы отношений*, соответственно;
- частичного порядка $\leq_c$ на C , называемого *иерархией понятий* или *таксономией*;
- функции $\sigma : R \rightarrow C^+$, называемой *сигнатурой*;
- множества A логических аксиом.

В работе [102] онтология имеет вид:

$$\langle C, A, T, D, R, F \rangle,$$

где C – множество классов, описывающих понятия некоторой предметной или проблемной области; A – множество атрибутов, описывающих свойства понятий; T – множество типов значений атрибутов; D – множество доменов; R – множество отношений, заданных на классах (понятиях); F – множество ограничений на значения атрибутов.

В работе [4] определяют следующие свойства онтологии:

1. Онтологии представляют собой спецификации на формальном языке, в которых фиксируются договоренности группы специалистов о том, что и как называется в их области и каким свойствам (соотношениям) удовлетворяет.
2. На логическом уровне каждой онтологии соответствует некоторая теория (сигнатура+аксиомы), а иногда и некоторая фиксированная модель (множества+операции+отношения). Вопросы к онтологии интерпретируются как запросы к соответствующей ей теории (модели).

3. Онтологии, как правило, строятся по модульному принципу: при определении новой онтологии могут использоваться уже ранее построенные онтологии.

4. Онтологии должны быть удобны для понимания специалистами и интерпретироваться системами при использовании.

В качестве баз знаний, помимо онтологий, также могут рассматриваться [56]:

- 1) словарь с определениями,
- 2) простая таксономия,
- 3) тезаурус (таксономия с терминами),
- 4) модель с произвольным набором отношений,
- 5) таксономия и произвольный набор отношений,
- 6) полностью аксиоматизированная теория.

Онтология может использоваться во многих задачах, некоторые из которых определены в работе [99]:

- для совместного использования людьми или программными агентами общего понимания структуры информации;
- для возможности повторного использования знаний в предметной области;
- для отделения знаний в предметной области от оперативных знаний;
- для анализа знаний в предметной области и т.д.

Для реализации возможности использования онтологии при извлечении знаний необходимо, чтобы в онтологии понятия были связаны с терминами предметной области. Такого рода онтологии называются лингвистическими [14], [56]. Одной из известных лингвистических онтологий является ресурс WordNet.

В настоящее время в научной среде ведутся работы по интеграции баз знаний в методах информационный поддержки:

- методы информационного поиска [13], [19], [23], [48], [53], [75], [77], [92], [98], [102], [103];

- методы классификации документов [15], [18], [50];
- методы кластеризации документов [87], [92];
- методы онтологической персонификации пользователей [72], [74], [96], [100], [114], [117], [121].

Рассмотрим ряд работ, в которых онтология применяется в задачах информационной поддержки. В работе [102] рассматривает применение прикладной онтологии в задачах информационного поиска. Онтологическая модель состоит из двух слоев: первый слой содержит категорию имен, в то время как второй слой содержит категорию ключевых слов, связанные с категориями имен из первого слова. На рисунке 1.2 представлена структурная модель онтологии

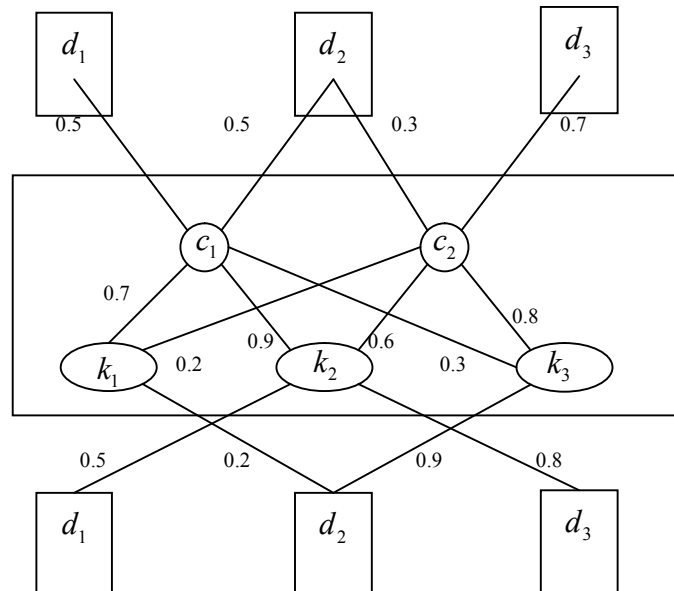


Рис. 1.2. Нечеткое отношение в онтологии

На рисунке представлены две категории c_1 и c_2 , и три ключевых слова k_1 , k_2 и k_3 . Каждая категория c_i связана с ключевым словом k_j отношением с весом, равным величине нечеткой ассоциации $r_{i,j} \in [0,1]$. На рис. 1.2 документы d_1 , d_2 и d_3 соединены с онтологией.

Онтология определена в виде матрицы релевантности R следующим образом:

$$R = \begin{matrix} & c_1 & c_2 & \cdots & c_m \\ \begin{matrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{matrix} & \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix} \end{matrix},$$

где

n – количество ключевых слов второго слоя $1 \leq i \leq n$;

m – количество категории первого слоя, $1 \leq i \leq m$;

$r_{i,j} \in [0,1]$ – степень релевантности между k_j и c_i .

Запрос представляется в следующем виде: $Q = \{k_i, c_j\}$, как элементы множество слов и категории, соединенные операторами AND или OR, где Q есть вектор $x = [x_1, x_2, \dots, x_i, \dots, x_n]$ $1 \leq i \leq n$ и $y = [y_1, y_2, \dots, y_j, \dots, y_m]$, $1 \leq i \leq m$ такие что:

$$x_i = \begin{cases} 1 & \text{если } k_i \in Q \\ 0 & \end{cases}.$$

$$y_j = \begin{cases} 1 & \text{если } c_j \in Q \\ 0 & \end{cases}.$$

Документ D представлен в виде коллекции документов $D = \{d_1, d_2, \dots, d_{doc}, \dots, d_u\}$, рассматривая множество $K = \{k_1, k_2, \dots, k_i, \dots, k_n\}$, как множество ключевых слов и $C = \{c_1, c_2, \dots, c_j, \dots, c_m\}$, как множество категорий.

Документы можно представить в виде матриц T_k и T_c :

$$T_k = \begin{matrix} & k_1 & k_2 & \cdots & k_n \\ \begin{matrix} d_1 \\ d_2 \\ \vdots \\ d_u \end{matrix} & \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{u1} & \alpha_{u2} & \cdots & \alpha_{un} \end{bmatrix} \end{matrix},$$

где $\alpha_{doc,i} \in [0,1]$ – степень релевантности между документом d_{doc} и ключевым словом k_i , $1 \leq doc \leq u$, $1 \leq i \leq n$;

$$T_c = \begin{matrix} & c_1 & c_2 & \cdots & c_m \\ \begin{matrix} d_1 \\ d_2 \\ \vdots \\ d_u \end{matrix} & \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{u1} & \beta_{u2} & \cdots & \beta_{um} \end{bmatrix} \end{matrix},$$

где $\beta_{doc,j} \in [0,1]$ – степень релевантности между документом d_{doc} и категорией $c_j, 1 \leq doc \leq u, 1 \leq j \leq m$.

Определение 1. Пусть x будет вектор запроса, состоящий из ключевых слов, и пусть R будет онтология нечетких отношений, тогда композиция из x и R – это нечеткое множество G_c , которая представляет собой категорию, ассоциированную с термами запроса, т.е. $G_c = x \circ R$.

Определение 2. Пусть y будет вектор запроса, состоящий из категорий, и пусть R онтология нечетких отношений. Композиция из R и y – это нечеткое множество G_k , которое представляет собой набор ключевых слов, ассоциированных с термами запроса, т.е. $G_k = R \circ y$, где символ $\circ$ обозначает минимаксную композицию.

Возможны три ситуации нахождения релевантных документов [87]:

1. Q содержит только элементы из множества K ;
2. Q содержит только элементы из множества C ;
3. Q содержит как элементы множества K , так и C .

Формальное представление ранжирования документов выполняется следующим образом:

Пусть $F_k = [f_{k1}, f_{k2}, \dots, f_{ki}, \dots, f_{kn}]$, $1 \leq i \leq n$ – нечеткое множество, полученное из G_k и пусть $F_c = [f_{c1}, f_{c2}, \dots, f_{cj}, \dots, f_{cm}]$ $1 \leq j \leq m$ – нечеткое множество, полученное из G_c такие, что:

$$f_{ki} = \begin{cases} g_{ki} & \text{если } g_{ki} > z_1 \\ 0 & \end{cases},$$

$$f_{cj} = \begin{cases} g_{cj} & \text{если } g_{cj} > z_1 \\ 0 & \end{cases}$$

где

g_{ki} – i -й компонент G_k ;

g_{cj} – j -й компонент G_c ;

z_1 – порог, установленный разработчиком системы.

Определение 3. Композиция множества F_k и матрицы T_k дает нечеткое множество V_{dk} , которое представляет собой степень релевантности между коллекцией документов и ключевым словом k_i т.е. $V_{DK} = F_k \circ T_k$.

Определение 4. Композиция множества F_c и матрицы T_c дает нечеткое множество V_{DC} , которое представляет собой степень релевантности между коллекцией документов и категорией c_j т.е. $V_{DC} = T_c \circ F_c'$, где ' означает транспонирование.

В работе [1] для процесса классификации текстовых документов в качестве источника знаний используется Общественно-политический тезаурус РуТез [15] (далее – Тезаурус РуТез). РуТез был разработан АНО Центр Информационных Исследований и является основой тематического анализа в рамках Автоматизированной лингвистической обработки текстов (АЛОТ) [15], используемого в УИС РОССИЯ. РуТез – это иерархическая сеть понятий, которая включает значительно больше понятий, отношений, синонимов, чем традиционные тезаурусы для ручного индексирования (75 тысяч терминов, 29 тысяч понятий, связанных более чем 100 тысячами непосредственных отношений).

В данном подходе значимость термина для содержания текста определяется в результате построения, так называемого тематического представления текста, независимом от величины и типа текстов. Основные этапы построения тематического представления текста [13], [15]:

1. Сопоставление текста с Тезаурусом создает для текста дескрипторный индекс, в котором указывается, какие дескрипторы Тезауруса и в каком месте текста обнаружены.

2. Нахождение для каждого дескриптора текста тематически близких дескрипторов и отображение этой информации в так называемой тезаурусной проекции текста.

3. Использование связей дескрипторов в тезаурусной проекции для разрешения многозначности терминов.

4. Построение текстовых связей для каждого дескриптора текста, то есть фиксация для каждого вхождения каждого дескриптора трех соседних дескрипторов вправо и трех влево. Выбор таких цифр – величина экспериментальная, однако, согласуется с экспериментами в области исследования кратковременной памяти.

5. Построение тематических узлов – групп близких по смыслу терминов. Тематические узлы строятся для тех дескрипторов, которые отличаются своей частотностью или местоположением в заголовках или в начале текста.

6. Выбор среди построенных тематических узлов – основных тематических узлов, то есть тех, которые моделируют элементы основной темы текста. Выбор производится на основе анализа суммированных текстовых связей тематических узлов.

В зависимости от того, элементом какой структуры тематического представления оказывается понятие d тезауруса, формируется оценка значимости $\omega(d; D)$. Типичные значения – 0.9 для центра основного тематического узла, 0.7 – для элемента основного тематического узла, 0.75 – для центра локального тематического узла и т.д. Окончательно вес понятия для текста определяется добавлением стабилизирующего фактора, учитывающего частотность понятия в документе:

$$\theta(d) = \alpha \cdot \omega(d; D) + (1 - \alpha) \cdot \frac{\text{freq}(d; D)}{\max_c \text{freq}(c; D)}, \quad \alpha = 0.7$$

Каждая рубрика R описывается дизъюнкцией альтернатив, каждый дизъюнкт представляет собой конъюнкцию:

$$R = \bigcup_i D_i = \bigcup_i \left[\bigcap_j K_{ij} \right] = \bigcup_i \left[\bigcap_j \left(\bigcup_k d_{ijk} \right) \right].$$

Конъюнкты K_{ij} описываются экспертами с помощью так называемых «опорных» понятий тезауруса d_{ijk} . Для каждого опорного понятия задается правило его расширения $f(\cdot)$, определяющее каким образом вместе с опорным понятием учитывать подчиненные ему по иерархии понятия. Выделяются три случая – без расширения, полное расширение по дереву иерархии тезауруса и расширение только по родовидовым связям.

Оценка релевантности содержания текста в рубрике (вес рубрики) может быть рассчитана на основе информации о весах понятий в тексте, входящих в ее описание.

В УИС РОССИЯ вес конъюнкта рассчитывается по формуле:

$$\theta(K_{ij}) = \min \{1.0; \max \{ \theta(d_{ijk}), \chi \cdot \theta(p_{ijm}) \} \} \quad (1.1)$$

где $d_{i,j,k}$ – понятия, не требующие подтверждения; $p_{i,j,m}$ – понятия, требующие подтверждения, χ – множитель, равный единице, если имеются понятия, не требующие подтверждения, и нулю – иначе.

Вес дизъюнкта предназначен учитывать не только сумму весов составляющих его конъюнктов, но и меру близости конъюнктов в тексте:

$$\theta(D_i) = \frac{\sum_{j=1}^m \theta(K_{i,j}) + \sum_{j < k} S(K_{i,j}, K_{i,k})}{m + C_m^2}, \quad (1.2)$$

где

$$S(K_{i,j}, K_{i,k}) = \min \left\{ 1.0; \frac{\sum s(c_{i,j,q} \in K_{i,j}, d_{i,k,w} \in K_{i,k})}{\max s(c \in D, d \in D)} \right\} \quad (1.3)$$

– сумма всех текстовых связей между понятиями одного конъюнкта и понятиями другого, деленная на значение максимальной текстовой связи между любыми двумя понятиями текста. Этот член равен обычно единице, для сильно связанных конъюнктов, и принимает малое значение, если понятия различных конъюнктов обсуждались в разных местах текста.

Вес рубрики представляет максимум весов входящих в описание рубрики альтернатив. В случае имеющихся иерархических связей между рубриками, оценка релевантности нижестоящих рубрик переносится на вышестоящие. При запросе по вышестоящей рубрике будут выходить документы, к которым были приписаны нижестоящие рубрики.

Алгоритм рубрицирования работает следующим образом: для всех понятий тезауруса, найденных в тексте, определяется множество рубрик, которые могут быть определены в тексте. Для каждой рубрики происходит расчет ее веса по формулам (1.1) – (1.3). В результирующем множестве остаются рубрики, вес которых превосходит задаваемый заранее для коллекции порог.

В задачах персонификации профилей пользователей онтология также нашло широкое применение [78], [81], [96], [115]. При этом основной идеей является применение существующих базы коллективных знаний. Так в работах [101] применяются известные базы знаний DBpedia, ODP. Также рассматриваются возможности автоматического формирования прикладной онтологии с использованием дампа категорий «Википедия», в качестве словаря понятий с возможностью получения статей, представляющие семантику этого понятия. Данная технология получила название викификации [97], [104], [116].

1.5. Анализ существующих программных систем поддержки проектных решений в электронных архивах технических документов

В настоящее время многие ведущие производители программного обеспечения предлагают свои продукты и решения в области анализа текстовых документов [27]. Рассмотрим несколько из них:

- TDMS 4.0;
- Архив ПСД;

- TextAnalyst (Megaputer Intelligence);
- Text Miner (SAS Institute);
- Oracle Text (Oracle).

Платформа TDMS 4.0 – программный продукт нового поколения для построения решений по управлению информационным содержанием и проектным производством.

Основой системы TDMS является объектно-ориентированное ядро, позволяющее гибко настроить среду разработки объектов проектирования (проектов, изделий, сооружений) и связать в единое информационное пространство разнородные электронные документы, внешнюю и внутреннюю почту, систему управления, отслеживание хода выполнения работ и обеспечение интеграции с прикладными программами, установленными на предприятии заказчика.

Архив ПСД – это типовое решение, предназначенное для централизованного учета, структурированного хранения и управления проектной документацией. Информационная система Архив ПСД реализована на платформе AS Archive.

Типовое решение Архив ПСД решает все основные задачи управления проектно-сметной документацией, в том числе:

- создание и хранение электронных копий документов – ввод бумажных документов со сканера, импорт электронных документов из файлового каталога и их размещение в едином файловом хранилище;
- регистрация и индексирование документов – автоматизированный учет поступления и хранения документации;
- систематизация документов – систематизация, каталогизация и многоуровневое описание информации и документации. Хранение, создание и редактирование словарей, классификаторов и рубрикаторов, необходимых для систематизации документов;

- навигация по архиву и поиск документов – поиск документов по атрибутам, сложным запросам и тексту распознанных документов. Различные представления электронного каталога документации;
- обеспечение доступа к документам – разграничение прав доступа к электронным версиям документов. Информирование пользователей о новых поступлениях. Формирование и работа с заявками на выпуск/печать документации;
- работа с электронными версиями документов – просмотр и редактирование электронных документов в среде приложения управления электронным архивом документов;
- подготовка (комплектование) электронных документов – формирование электронных версий комплектов проектно-сметной документации. Экспорт в форматы сторонних приложений и другие информационные системы;
- обеспечение защиты информации – конфиденциальность документов, атрибутивной и служебной информации. Протоколирование работы пользователей;
- формирование отчетности – формирование стандартных и пользовательских отчетов с помощью визуально простого интерфейса. Автоматизированное формирование типовых документов и справок.

Система TextAnalyst решает следующие задачи Text Mining:

- создание семантической сети текстового документа (поиск ключевых понятий в текстовом документе и установление взаимоотношений между ними);
- подготовка резюме текстового документа (выделение в текстовом документе предложений, содержащих значимые для данного текстового документа термины);
- поиск по текстовому документу (предусмотрено использование запросов на естественном языке);

- автоматическая классификация и кластеризация текстовых документов.

Text Miner SAS – система, для сравнения определенных грамматических и словесных рядов в текстовом документе.

Text Miner позволяет определять, насколько правдив тот или иной текстовый документ. Обнаружение лжи в текстовых документах производится путем анализа содержимого текстового документа и выявления изменений стиля письма, которые могут возникать при попытке исказить или скрыть информацию.

Oracle Text – программный комплекс, интегрированный в СУБД, позволяющий эффективно работать с запросами к текстовым документам (в том числе, с помощью PL/SQL). Основной задачей является поиск текстовых документов по их содержимому – словам или фразам, которые при необходимости комбинируются с использованием булевых операций. Результаты поиска ранжируются по релевантности, с учетом частоты встречаемости терминов запроса в найденных текстовых документах. Для повышения полноты поиска используются следующие методы:

- расширение терминов запроса всеми морфологическими формами, что реализуется привлечением знаний о морфологии языка;
- расширение терминов запроса близкими по смыслу терминами за счет подключения тезауруса;
- расширение запроса терминами близкими по написанию и по звучанию.

Возможности обработки текстовой информации на русском языке в Oracle Text достаточно ограничены.

1.6. Цели и задачи исследования

При анализе существующих моделей поиска и классификации текстовых документов выявлен ряд проблем:

- отсутствуют методы автоматизированного формирования онтологии предметной области для применения в задачах информационной поддержки проектирования АС;
- в традиционных моделях поиска и классификации не учитывается реальная информационная потребность пользователя;
- большая вычислительная сложность традиционных методов информационного поиска и классификации.

Таким образом, необходимо рассмотреть возможность автоматизированного формирования онтологии предметной области и применение ее в задачах интеллектуального анализа технических документов с учетом модели жизненного цикла и стандартов, применяемых на предприятии при проектировании. Разработать средства автоматизированного формирования предметной онтологии, формирования контекстно-ориентированных профессиональных поисковых запросов к электронному архиву, провести вычислительные эксперименты, доказывающие их эффективность, внедрить полученные результаты в практику проектной организации.

Глава 2

Онтологические модели и алгоритмы информационной поддержки процесса проектирования

Современная крупная проектная организация обладает значительным по объему электронным архивом конструкторской и технической документации, большая часть которой представлена в текстовом неструктурированном виде. Фактически такой электронный архив текстовой документации содержит в себе опыт и знания большого количества высококвалифицированных специалистов, которые на протяжении многих лет занимались разработкой и проектированием сложных (АС).

Информационная поддержка реализации проектных процедур в электронных архивах осуществляется с использованием традиционных моделей формирования поисковых запросов и классификации документов. Однако данные модели не всегда отражают информационные потребности пользователя, и, тем самым, всегда присутствует вероятность того, что релевантные документы, которые были отобраны, не уменьшают информационную неопределенность пользователя. Для профессионального, в том числе научно-технического поиска информации, требуется привлечение специальных экспертных знаний, использование синонимов, возможности автоматического расширения запроса, возможностей автоматического анализа результатов запроса и помощь в интерактивном поиске. Для решения подобного рода задач применяются интеллектуальные модели информационной поддержки, функционирование которых основано на предметно-ориентированных знаниях. Эти знания могут быть представлены в виде онтологии предметной области [13].

В данной работе утверждается, что онтология должна учитывать специфику предметной области проектирования и иметь структуру, которая учитывает особенности процесса проектирования АС, понятийный аппарат

предметной области, отношение между концептами предметной области и конечным множеством интерпретации.

2.1. Модель онтологии информационной поддержки процесса проектирования автоматизированных систем

2.1.1. Задачи прикладной онтологии при проектировании автоматизированных систем

Прежде чем рассмотреть структуру онтологии, необходимо определить, какие цели преследует использование онтологии в задачах информационной поддержки проектирования АС и выявить, какие ограничения накладываются на нее в процессе разработки. Рассмотрим определение проектирования и особенности, которые присущи данному процессу. Следующее определение дано в работе [49]: «Проектирование технического объекта – это создание, преобразование и представление в принятой форме образа этого еще не существующего объекта».

Проектная деятельность имеет ряд специфических особенностей [63]:

1. Продуктом проектной деятельности является упорядоченная совокупность сведений, служащих знаковой моделью объекта, в момент проектирования реально еще не существующего.

2. Процедуры проектирования реального объекта соответствуют преобразованию его исходного описания в некотором конечном пространстве.

3. Способы преобразования информации при проектировании нельзя отразить в виде математических соотношений, т. е. невозможно построить строгую математическую модель такого процесса преобразования.

4. Ввиду сложности проектируемых объектов на каждом этапе разработки вовлекаются различные специалисты, что придает проектированию характер коллективной деятельности.

5. Проектируемый объект входит в упорядоченную иерархию объектов и, с одной стороны, выступает как часть системы более высокого уровня, а с другой — как система для объектов более низкого уровня. В соответствии с

этим процесс проектирования можно разделить на два этапа: внешнего (объект – часть системы более высокого ранга) и внутреннего проектирования (объект – совокупность компонентов).

6. Проектирование, как правило, имеет итерационный многовариантный характер, для принятия проектных решений используются различные научно-технические знания.

Рассмотрим особенности АС как подкласса сложных технических систем. Согласно ГОСТ 34.003-90: Автоматизированная система – это система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций. Стандарт IEEE 1471 такой тип систем определяет как "software intensive systems" – системы, интенсивно использующие программное обеспечение, что указывает на существенное влияние программного обеспечения, входящего в их состав.

Одним из основных принципов разработки и внедрения АС является принцип единства информационной базы, реализация которого предполагает создание единой динамической информационной модели объекта проектирования, содержащий необходимый и достаточный перечень показателей по информационной поддержке всех этапов жизненного цикла разрабатываемой системы.

Будем рассматривать определение онтологии проектирования как формализованное описание знаний субъектов проектирования о процессе проектирования новых или модернизаций уже известных артефактов, включая знания о самом объекте проектирования и близких к нему по свойствам артефактов, а также тезаурус предметной области. Таким образом, для того чтобы описать процесс проектирование необходимо включить в прикладную онтологию понятие жизненного цикла (ЖЦ), которая отражает различные состояния проектируемой системы, начиная с момента возникновения необходимости в данной системе и заканчивая в моменте снятия с эксплуатации. Каждый такой этап определяется различными

конструкторскими, регламентными и нормативными документами. Основными целями применения онтологии информационной поддержки проектирования АС являются:

1. Возможность расширения знания об исследуемом объекте и среде, которая его окружает;
2. Сокращение времени выполнения контекстно-ориентированных проектных запросов к электронным архивам и удовлетворение информационной потребности проектировщика.

В данной работе предлагается рассматривать информационную потребность проектировщика не на уровне терминов поискового запроса, а на уровне концептуального описания предметной области в виде прикладной онтологии.

Основными требованиями к прикладной онтологии информационной поддержки проектирования АС являются следующие:

1. Онтология должна включать ограничения на состав и последовательность основных стадий и этапов проектирования АС, а также их отношения с артефактами проектирования в виде технических документов.
2. Структура онтологии должна основываться на разделении понятийного аппарата и множества терминов, связанных отношениями.
3. Множество понятий онтологии должно иметь возможность автоматического доопределения, для обеспечения настройки на тематику проекта АС с использованием открытых общедоступных информационных ресурсов.

Рассмотрим систему контекстов, в которых происходит проектирование современной АС. *Контекст процесса проектирования* является внешним контекстом относительно проектной организации. Информационные ресурсы данного контекста определяются на основе государственных стандартов, отраслевых стандартов и т.д. и описывают методологию проектирования, а также понятийный аппарат в самом общем

виде (например, понятия "проект", "технический документ", "стадия проектирования" и другие).

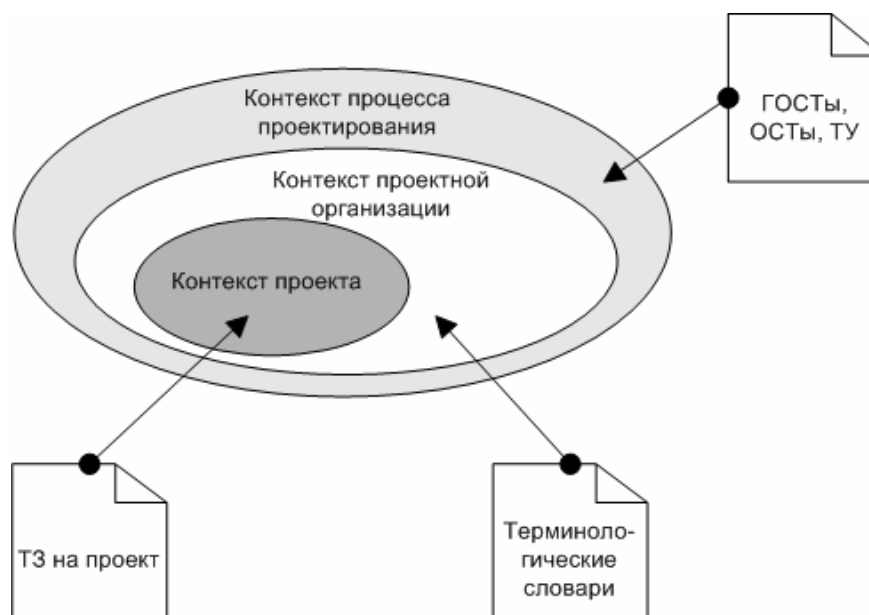


Рис.2.1 Система вложенных контекстов проектной организации

Контекст проектной организации определяется системой понятий и терминов, которые используются внутри организации в процессе проектной деятельности. Данный контекст формируется на основе существующих в организации терминологических словарей и (или) термины, и понятия извлекаются из технических документов электронных архивов. Отношение вложенности контекста проектной организации в контекст процесса проектирования определяет зависимость внутреннего контекста от внешнего. Внешний контекст выступает в роли ограничений, которым должен удовлетворять внутренний контекст (например, контекст процесса проектирования формируется с использованием ГОСТ 27.002-89 "Надежность в технике. Основные понятия. Термины и определения", тогда в состав понятий контекста проектной организации может быть включен концепт "Внезапный отказ").

Контекст проекта образуют термины документа, на основе которого проектируется АС (техническое задание и (или) технико-экономическое обоснование).

2.1.2. Структурно-функциональная модель информационной поддержки проектировщика

В процессе проектирования АС у проектировщика возникают различные информационные потребности в зависимости от вида проекта, стадии жизненного цикла проектируемого изделия. Удовлетворение информационной потребности в различные моменты времени направлено на решение конкретной проектной задачи.

На рис. 2.2 представлена схема уточнения проектных запросов к электронному архиву технической документации. Поскольку в общем случае у разных проектировщиков в различные моменты времени возникают разные информационные потребности, то их учет при формировании проектных запросов способен привести к повышению показателей качества получаемых результатов (повышение точности и полноты откликов на проектные запросы). Для этого будем применять систему *профилей проектировщиков*.

Формальное описание профиля проектировщика, основанное на системе онтологий, будет представлено в работе далее. Основная цель применения профиля проектировщика при формировании запросов к электронному архиву состоит в определении подходящего контекста решения проектной задачи в соответствии с системой вложенных контекстов проектной организации (рис. 2.1).

При уточнении проектных запросов $\{Q\}^i$ используются информационные ресурсы из онтологии методологии проектирования и онтологии предметной области (рис.2.2). Первая из них позволяет определить стадию проектирования на основе анализа текущего состояния st_i^P реализуемого проекта P , используя знания о том, какие типы документов формируются в рамках проектных стадий.

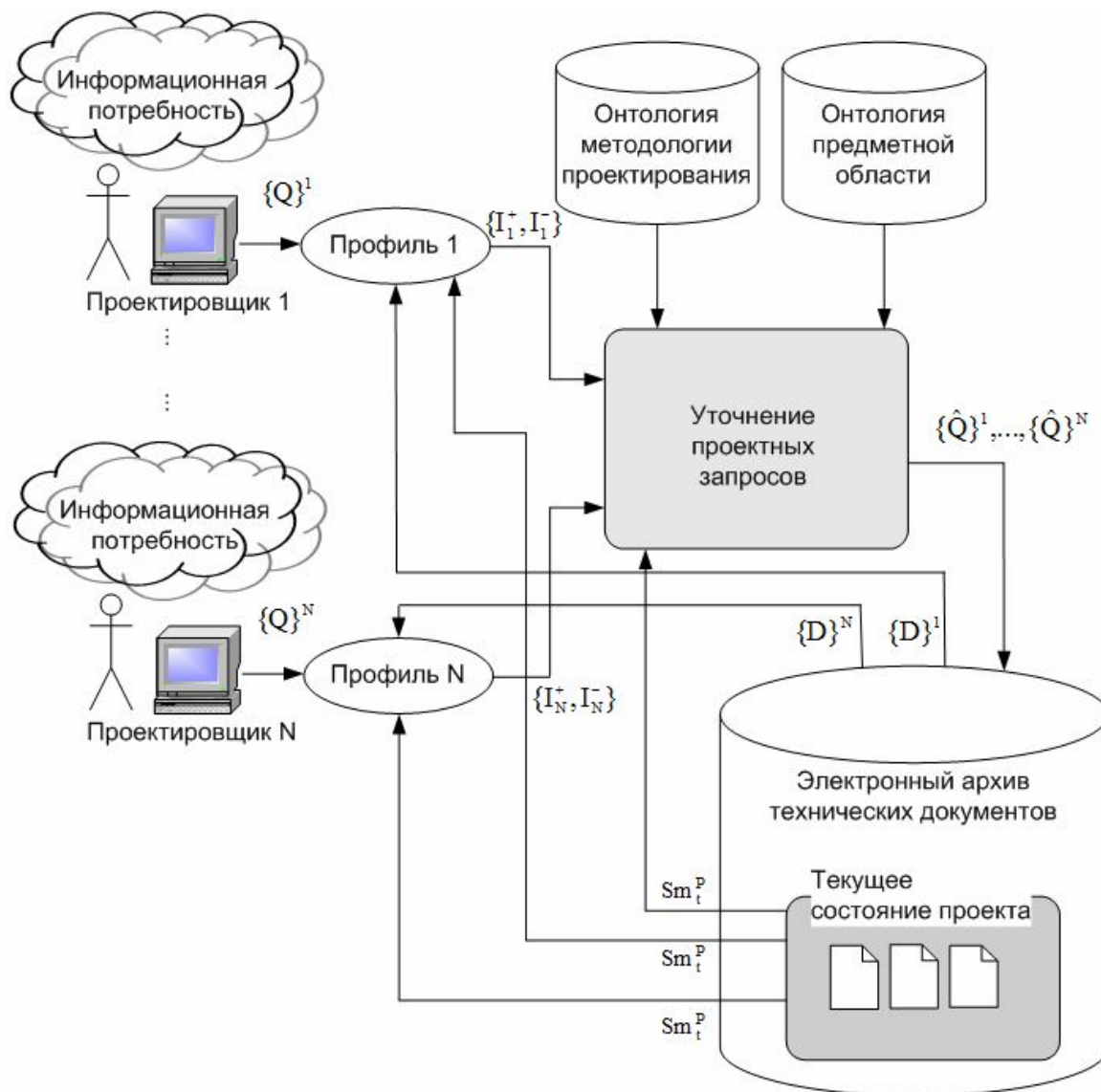


Рис.2.2 Структурно-функциональная схема уточнения проектных запросов

Онтология предметной области определяет контекст проектной организации и контекст проекта. При формировании профиля проектировщика указанные контексты используются для определения понятий предметной области, которые явно соответствуют или явно не соответствуют текущей информационной потребности (множество $\{I_i^+, I_i^-\}$ на рис. 2.2).

Уточнение проектных запросов состоит в определении понятий онтологии предметной области, которые соответствуют текущей информационной потребности. Используя соответствующие терминологические окружения понятий, происходит уточнение набора

ключевых терминов запросов $\{Q\}^i$. В результате получаем уточненное множество $\{\hat{Q}\}^i$.

2.1.3. Структурно-функциональная модель прикладной онтологии

Принимая во внимание тот факт, что проектирование АС выполняется в различных контекстах, будем рассматривать структуру прикладной онтологии информационной поддержки проектирования АС, состоящую из нескольких уровней (рис.2.3). Онтологический базис, состоящий из классов, отношений и функций интерпретации, был выбран как наиболее часто применяемый при использовании таких языков представления онтологий, как RDF/RDFSchema и OWL.

В общем виде формально онтологию будем представлять в виде кортежа:

$$O = \langle O^{ML}, O^{Dm}, G^D, \{G^P\}, R, F \rangle, \quad (2.1)$$

где O^{ML} – онтология методологии проектирования; O^{Dm} – онтология предметной области проектирования АС; G^D – концептуальная сеть организации; $\{G^P\}$ – множество концептуальных сетей реализуемых проектов, R – множество отношений между фрагментами онтологий, F – множество интерпретирующих функций.

Современные методологии проектирования АС предполагают четкую структуризацию процесса проектирования, включающую в себя деление на стадии, этапы и проектные процедуры. В частности, процесс разработки по RUP (Rational Unified Process – рациональный унифицированный процесс) предполагает следующую классификацию: деятельность, состоящая из процессов; цели; ход процесса; работы; основные артефакты и роли. Каждый процесс и отдельно взятая работа предполагает на выходе в качестве результатов некоторый набор артефактов, среди которых нам будут важны технические документы и их фрагменты, сохраняемые в электронном архиве проектной организации.

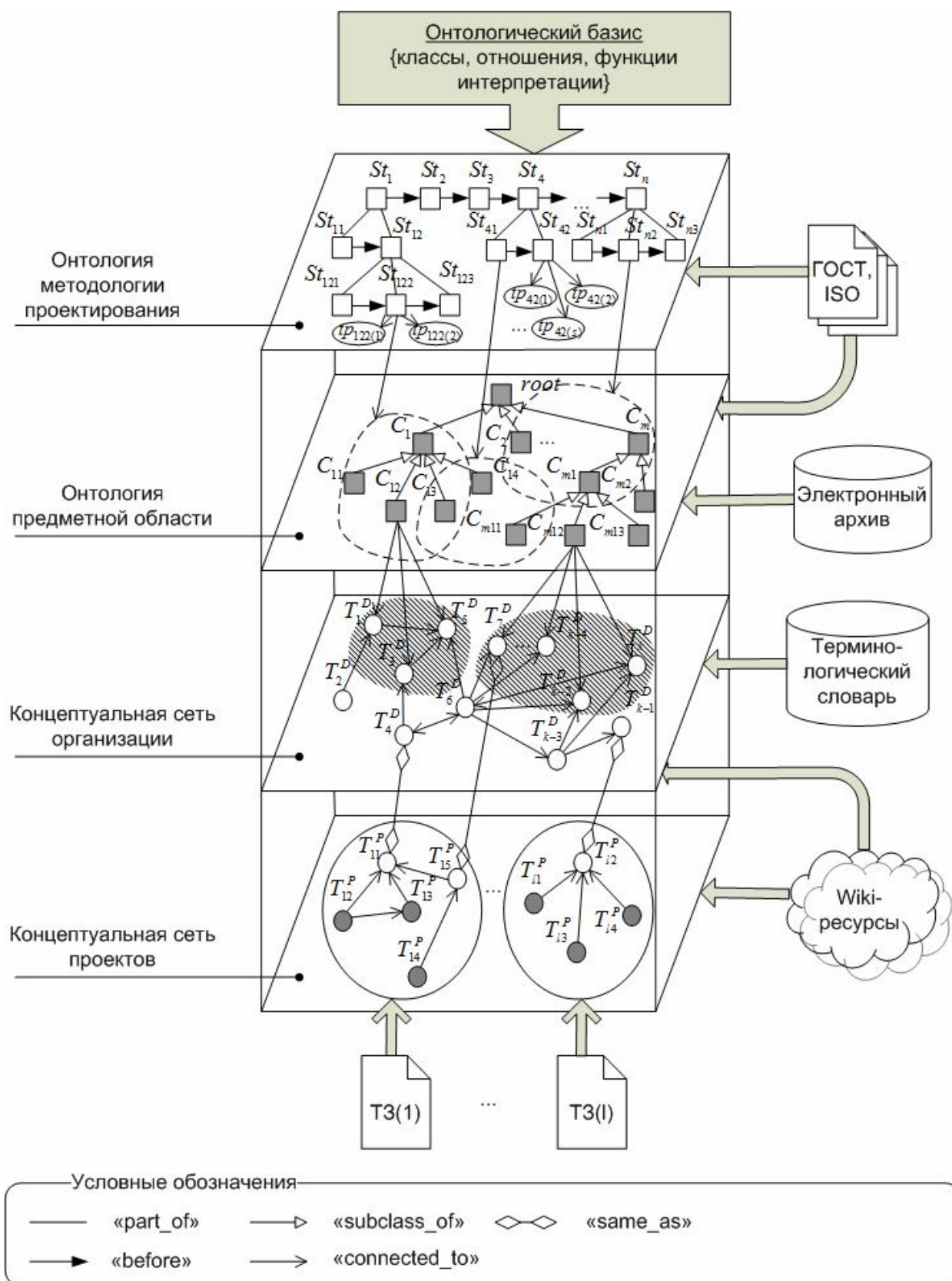


Рис.2.3 Структура прикладной онтологии информационной поддержки проектирования АС

Онтологию методологии проектирования O^{ML} (2.2) представим в виде множества:

$$O^{ML} = \{St, Tp, R^{ML}, F^{ML}\}, \quad (2.2)$$

где St – множество стадий (этапов) проектирования, соответствующее используемой методологии проектирования АС (например, RUP); Tp – множество типов ТД, являющихся артефактами определенных стадий (этапов); R^{ML} – множество семантических отношений онтологии O^{ML} ; F^{ML} – множество функций интерпретации в O^{ML} .

Множество отношений R^{ML} включает в себя три класса отношений:

$$R^{ML} = \{R_B^{ML}, R_{Cn}^{ML}, R_{Pt}^{ML}\}, \quad (2.3)$$

где R_B^{ML} – бинарное транзитивное отношение, имеющее семантику «before» и связывающее две стадии проектирования, следующие по времени друг за другом; R_{Cn}^{ML} – бинарное отношение с семантикой «connected_to», соединяющее стадию или этап проектирования с одним или несколькими классами (видами) ТД из множества артефактов Tp .

Таким образом, запись вида $St_i R_{Cn}^{ML} \{tp_j, tp_k\}$ означает, что в качестве результатов стадии проектирования St_i понимаются документальные артефакты типов tp_j и tp_k . То же самое, но более кратко будем записывать как $St_i | tp_j, tp_k$.

Представим множество стадий проектирования АС St и множество возможных классов артефактов Tp в виде четкого соответствия $\Gamma_{St/Tp} = (St, Tp, F_{St/Tp})$, которое в матричном виде задается с помощью матрицы инцидентий $R_{\Gamma_{St/Tp}}$, строки, которой помечены элементами $St_i \in St$, $i \in I = \{1, 2, \dots, n\}$, столбцы – элементами $tp_j \in Tp$, $j \in J = \{1, 2, \dots, m\}$, а на пересечении St_i строки и tp_j столбца ставится элемент $r_{ij} = \gamma_{F_{St/Tp}} \langle St_i, tp_j \rangle$,

где $\gamma_{F_{St/TP}} = \{0, 1\}$ – характеристическая функция, определяющая, существует ли отношение $St_i | tp_j$ ($\gamma_{F_{St/TP}} = 1$) или такого отношения нет ($\gamma_{F_{St/TP}} = 0$).

Введем понятие *текущего состояния проекта* P , под которым будем понимать множество типов ТД $Sm_t^P = \{tp\}$, где t – текущий момент времени, $tp \in \{tp_1, tp_2, \dots, tp_\lambda\}$ – тип ТД в составе проекта из перечня классов документальных артефактов длиной λ .

Класс отношений R_{Pt}^{ML} определяет бинарные транзитивные отношения «part_of» между стадиями, этапами и проектными процедурами (определяются в соответствии с используемой в организации методологией проектирования). Запись вида $St_{ijk} R_{Pt}^{ML} St_{ij}$ будет означать включение этапа St_{ijk} в состав стадии St_{ij} .

Множество интерпретирующих функций F^{ML} включает в себя два класса функций:

$$F^{ML} = \{F_{St}^{ML}, F_{tp}^{ML}\}, \quad (2.4)$$

где $F_{St}^{ML} : St_{ijk} \rightarrow St_{ij}$ – функция, сопоставляющая этапу (стадии) проектирования соответствующую стадию (проектную процедуру); $F_{tp}^{ML} : \{tp\}_i \rightarrow St_i$ – функция, сопоставляющая множеству типов документальных артефактов соответствующую стадию, этап или проектную процедуру.

Рассмотрим алгоритм определения текущей стадии проектирования на основе состояния проекта в некоторый момент времени t :

Шаг 1. Определение множества технических документов $D_t^P = \{d_1^P, d_2^P, \dots, d_m^P\}$ в репозитории реализуемого проекта в момент времени t .

Шаг 2. Определение текущего состояния проекта P : $Sm_t^P = \{tp_1, tp_2, \dots, tp_\lambda\}_t^P$ как отображение D_t^P на множество возможных типов артефактов $\{tp_1, tp_2, \dots, tp_\lambda\}$.

Шаг 3. Формирование множества $\{St_t^P\}$, элементы которого удовлетворяют условию:

$$St_{i(j)(k)} \mid \{tp\} \supset \{tp_1, tp_2, \dots, tp_l\}_t^P.$$

Шаг 4. Используя отношения R_B^{ML} и R_{Pt}^{ML} , определяется $St_{i(j)(k)}$ такая, для которой не выполняется следующее условие:

$$\forall St_{i(j)(k)}, St_{x(y)(z)} \in \{St_t^P : St_{i(j)(k)} R_B^{ML} St_{x(y)(z)}\}.$$

Шаг 5. Для определенного на предыдущем шаге $St_{i(j)(k)}$ находится стадия (этап или проектная процедура), которая следует за ней. Формально должно выполняться условие: $\exists St_t^P : St_{i(j)(k)} R_B^{ML} St_t^P$.

Онтологию предметной области представим в виде множества:

$$O^{Dm} = \{root, C, R_G^D, F^{Dm}\},$$

где $root$ – корневая вершина таксономии понятий; C – множество понятий предметной области, которая образует основу понятийного аппарата процесса проектирования АС; R_G^D – антисимметричное, транзитивное, нереклексивное бинарное отношение обобщения; $F^{Dm} : \{C\} \rightarrow \{C\}$ – функция интерпретации понятий онтологии предметной области, позволяющая осуществлять переход от потомков к родительским понятиям.

Концептуальную сеть организации представим в виде графа:

$$G^D = \langle T^D, R_{Cn}^D \rangle,$$

где T^D – множество понятий проектной организации, извлекаемых из терминологических словарей и внешних wiki-ресурсов; R_{Cn}^D – отношение, связывающее различные понятия друг с другом.

Концептуальную сеть проектов представим как множество графов:

$$\{G^P\} = \{\langle T^P, R_{Cn}^P \rangle\},$$

где T^P – множество понятий проектной организации, извлекаемых из технических заданий на проектирование АС и внешних wiki-ресурсов; R_{Cn}^P – отношение, связывающее различные понятия друг с другом.

Рассмотрим виды отношений и интерпретирующие функции, которые имеют место между определенными ранее фрагментами прикладной онтологии информационной поддержки.

Множество отношений R включает в себя три класса отношений:

$$R = \{ R^{SC}, R^{CD}, R^{DP} \},$$

где R^{SC} – бинарное отношение, имеющее семантику «connected_to» и связывающее стадию (этап или проектную процедуру) процесса проектирования с подмножеством понятий прикладной онтологии; R^{CD} – бинарное отношение, имеющее семантику «connected_to» и связывающее понятие прикладной онтологии с понятием концептуальной сети организации; R^{DP} – бинарное симметричное отношение, имеющее семантику «same_as», связывающее понятие концептуальной сети проектной организации с понятием из концептуальной сети проектов.

Отношение R^{SC} позволяет на основе знаний о текущей стадии (этапе или проектной процедуре) определить соответствующее ей подмножество понятий предметной области. Интерпретация R^{CD} связана с возможностью определения степени выраженности понятия в некотором фрагменте текстового документа через связанный с ним набор терминов, входящих в *терминологическое окружение*. В работах [46], [64], [65], [66], [67] используется следующее определение близких по смыслу терминов к концептам предметной области: *Под терминологическим окружением понятия предметной области будем понимать множество терминов (слов) из ТД в электронном архиве, которые наиболее близки с данным понятием в семантическом смысле*. Отношение R^{DP} показывает общие термины из двух терминологических сетей: проектной организации и реализуемых проектов.

Множество интерпретирующих функций представлено в виде:

$$F = \{ F_{SC}, F_{DC}, F_{DP} \},$$

где $F_{SC} : St \rightarrow \{ C \}$ – функция, сопоставляющая стадии проектирования (этапу, проектной процедуре) подмножество понятий предметной области,

задаваемая алгоритмически; $F_{DC} : \{T^D\} \rightarrow C$ – функция интерпретации терминов, сопоставляющая каждому концепту набор терминов из словаря (формирует терминологическое окружение понятия); $F_{DP} : T^D \times T^P \rightarrow \text{boolean}$ – функция определения идентичных понятий в концептуальной сети проектной организации и в концептуальных сетях проектов, возвращающая значения: *истина* или *ложь*.

2.2. Онтологическая модель профиля проектировщика

2.2.1. Формализация профиля проектировщика с учетом информационных потребностей

Отдельными участниками процесса проектирования формируются проектные запросы к электронному архиву организации в соответствии со своими *информационными потребностями*, которые основываются как на объективных факторах, так и носят субъективный характер.

Рассмотрим несколько определений понятия «информационная потребность», сформулированных с различных точек зрения. *Информационная потребность* – необходимость в информации, требующая удовлетворения и обычно выражаемая в информационном запросе, одно из центральных понятий в информатике.

Если принять за основу понимание потребности как функциональной системы, то: «информационная потребность – это потребность в информационной деятельности, устраняющей дисбаланс (рассогласование между наличным и нормальным состоянием) информационной сферы субъекта» [55]. В свою очередь информационная деятельность понимается как совокупность процессов создания, сбора, понимания, переработки, хранения, поиска и распространения информации.

Еще одно определение *информационной потребности* согласно ГОСТ 7.73-96 «Поиск и распространение информации»: информационные потребности – характеристики предметной области, значения которых

необходимо установить для выполнения поставленной задачи в практической деятельности.

Формирование поисковых запросов к электронному архиву проектной организации нацелено на удовлетворение информационной потребности проектировщика. Правильно выразить информационную потребность, используя ограниченные возможности набора ключевых слов информационного запроса, является в определенной степени искусством, которое не поддается формализации. Верный набор ключевых слов требует хороших знаний предметной области и знаний о содержимом электронного архива, который может в проектных организациях достигать сотен тысяч документов, накопленных за весьма продолжительный промежуток времени.

Формализация *профиля проектировщика* осуществляется на основе предположения о том, что имеется возможность фиксировать результаты проектных запросов к электронному архиву в виде множества документов, удовлетворяющих информационной потребности, и множества документов, которые текущей информационной потребности не удовлетворяют.

Каждой информационной потребности In_j^i будем ставить в соответствие пару классов понятий онтологии предметной области $C^+ = \{c_1^+, c_2^+, \dots, c_n^+\}$, $C^- = \{c_1^-, c_2^-, \dots, c_m^-\}$, определяющие положительные и отрицательные подмножества понятий онтологии, соответственно (рис. 2.4).

Положительные C^+ и отрицательные C^- классы понятий формируются следующим образом [36], [37]:

- В процессе выполнения конкретным проектировщиком информационных запросов к электронному архиву определяется набор ТД, которые соответствуют его информационной потребности (D^+) и ТД, не соответствующие ей (D^-), с учетом текущей стадии (этапа) проектирования.
- Для каждого документа определяется его концептуальное представление. Другими словами, производится нечеткое

онтологическое индексирование. Запишем нечеткое соответствие между множеством $C^{+(-)}$ и множеством $T^D = T_{in}^D \cup T_{ext}^D$, как $\tilde{G}_{CT} = (C^{+(-)}, T^D, \tilde{F}_{CT})$, где $\tilde{F}_{CT}$ – нечеткое множество в $C^{+(-)} \times T^D$. Определим нечеткое соответствие $\tilde{G}_{CT}$ в виде ориентированного двудольного графа с множеством вершин $C^{+(-)} \cup T^D$, каждой дуге $\langle c_i^+, t_j \rangle$ ($\langle c_i^-, t_j \rangle$) которого приписываем значение функции принадлежности $\mu_{F_{CT}} \langle c_i^+, t_j \rangle$ ($\mu_{F_{CT}} \langle c_i^-, t_j \rangle$). Указанное значение функции принадлежности вычисляется на основе нормализованной частоты встречаемости термина в терминологическом окружении понятия. Поскольку терминологические окружения понятий формируются из внутреннего источника (терминологического словаря) T_{in}^D и из внешнего источника (профессиональных wiki-ресурсов) T_{ext}^D (рис. 2.4), а также $T_{in}^D \cap T_{ext}^D \neq \emptyset$, для определения функции принадлежности применяется следующее правило:

$$\begin{aligned} \mu_{F_{CT}} \langle c_i^+, t_j \rangle &= \mu_{F_{CT}} \langle c_i^+, t_{in(j)} \rangle \in T_{in}^D > \\ \text{if } \mu_{F_{CT}} \langle c_i^+, t_{in(j)} \rangle \in T_{in}^D &= \max(\mu_{F_{CT}} \langle c_i^+, t_{in(j)} \rangle \in T_{in}^D >, \mu_{F_{CT}} \langle c_i^+, t_{ext(j)} \rangle \in T_{ext}^D >) \\ \text{else } \mu_{F_{CT}} \langle c_i^+, t_j \rangle &= \mu_{F_{CT}} \langle c_i^+, t_{ext(j)} \rangle \in T_{ext}^D >. \end{aligned}$$

Образ множества T^D , при соответствии $\tilde{G}_{CT}$, фактически представляет собой нечеткое множество, элементами которого являются концепты с соответствующими степенями выраженности:

$$\tilde{G}_{CT}(T^D) = \{ \mu_{G_{CT}}(c^{+(-)}) / c^{+(-)} \},$$

где $\mu_{G_{CT}}(c^{+(-)}) = \bigvee_{t \in T^D} \mu_{F_{CT}} \langle c_i^{+(-)}, t_j \rangle$.

- В положительные и отрицательные подмножества понятий онтологии предметной области включаются такие понятия из онтологических представлений, степень выраженности которых наибольшая.

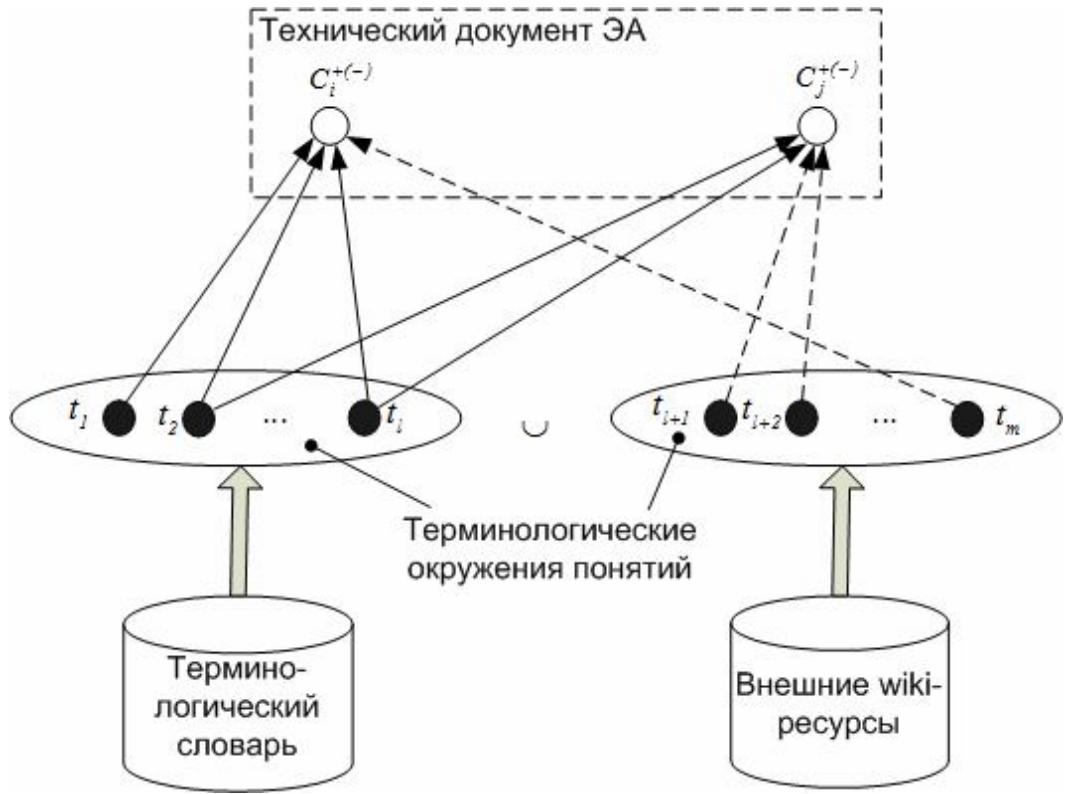


Рис.2.4 Структура терминологических окружений понятий при формировании профиля проектировщика

Формально профиль проектировщика будем представлять в виде кортежа [41]:

$$Ex_j^i = \langle In_j^i, C^+, C^- \rangle, c^{+(-)} = \mu_{F_{CT}}(c^{+(-)}) = \max_{c \in D} (\bigvee_{t \in T^D} \mu_{F_{CT}} \langle c_i^{+(-)}, t_j \rangle), c^{+(-)} \in C^{+(-)},$$

где i – индекс проектировщика, j – индекс стадии жизненного цикла проектирования АС.

2.2.2. Способ формирования онтологического контекста проекта

Контекст проекта ранее был определен как граф $G^P = \langle T^P, R_{Cn}^P \rangle$.

Учитывая интерпретирующую функцию F_{DC} , будем рассматривать контекст проекта не на уровне терминов, а на уровне понятий онтологии. Следовательно, контекст проекта перепишем как граф вида [38], [40], [44], [60], [61]:

$$G^{PT} = \langle C^{PT}, R^{PT} \rangle, \quad (2.5)$$

где C^{PT} – множество вершин-понятий проекта, R^{PT} – множество дуг, соединяющих вершины-понятия.

Множество понятий проекта определяется как результат функции онтологического индексирования технического задания (Tz) на реализуемый проект ($F_{ol}(Tz)$) и функции онтологического доопределения множества C^{Tz} , как результата $F_{ol}(Tz)$ с применением wiki-ресурсов сети Internet ($F_{cAdd}(C^{Tz})$). Алгоритм формирования онтологического контекста проекта АС представим в виде следующих шагов [43], [57]:

Шаг 1. Загрузка файла технического задания (Tz).

Шаг 2. Онтологическое индексирование технического задания:

$$C^{Tz} = F_{ol}(Tz).$$

Шаг 3. Доопределение множества C^{Tz} .

На данном шаге выполняется анализ wiki-ресурса сети Internet и определяется множество дополнительных понятий, имеющих связи с понятиями множества C^{Tz} . Идентификация связей между понятиями определяется на основе существующих гиперссылок на соответствующие страницы сети, содержащие текстовые описания понятий.

Шаг 4. Загрузка терминологического словаря Dis , который формируется на основе технической документации электронного архива, в том числе из набора основных терминов и понятий из ГОСТ, принятых к исполнению в проектной организации.

Шаг 5. Сравнение терминологических окружений $T_{sur}(\hat{C}^{Tz})$ понятий $\hat{C}^{Tz} = F_{cAdd}(C^{Tz})$ с терминами из Dis . Если $\forall \hat{w} \in T_{sur}(\hat{C}^{Tz})$ выполняется условие $\hat{w} \notin Dis$, то необходимо удалить понятие $\hat{c} \in \hat{C}^{Tz}$.

Шаг 6. Проверка очередного $\hat{c} \in \hat{C}^{Tz}$. Если сравнение терминологических окружений со словарем выполнено не для всех элементов множества $\hat{C}^{Tz}$, тогда выполняется переход к шагу 5.

Шаг 7. Определение множества дуг R^{PT} на основе анализа гиперссылок страниц wiki-ресурса.

Шаг 8. Сохранение графа G^{PT} .

Рассмотрим детально процесс формирования концептуальной сети, извлекаемой из «Википедии» – свободной общедоступной мультязычной универсальной интернет-энциклопедии, реализованной на принципах wiki. Концепты в данной библиотеке представлены в виде HTML-страниц. Для связи между страницами используются гиперссылки, которые символизируют семантическую связь между понятиями. Опираясь на систему гиперссылок, существует возможность в автоматическом режиме переходить от одной страницы к другой, извлекая знания о понятиях предметной области.

Рассмотрим определение множества дуг R^{PT} (шаг 7) на основе анализа гиперссылок «Википедии» [39], [58], [59]:

1. На первоначальном этапе извлекаются понятия из онтологии проекта.

2. Выполняется извлечение понятий из «Википедии». В основе данного процесса лежит модифицированный алгоритм волновой трассировки (применяемый при трассировке печатных плат радиоэлектронных устройств). Данный процесс состоит из ряда последовательных шагов:

- 2.1. На вход данного алгоритма поступают множества понятий полученных на этапе 1.

- 2.2. Выполняется поиск страниц в «Википедии», в которых концепты являются заголовками.

- 2.3. Страницы, полученные на этапе (2.2) анализируются с целью нахождения тех концептов, для которых одновременно выполняются условия:

- существует страница, которая описывает концепт;

- анализируемая страница содержит гиперссылки на страницу найденного концепта;

- страница концепта содержит обратную гиперссылку на анализируемую страницу.

2.4. Обнаруженные концепты добавляются в предварительное ядро онтологии.

2.5. Проверяется условие существования маршрута между всеми первичными концептами, которые получены на этапе (1).

2.6. Если условие пункта 2.5 выполняется, то это означает окончание модифицированного алгоритма волновой трассировки, если не выполняется, то пункты 2.2-2.5 выполняются снова для концептов, извлеченных на этапе 2.3.

Таким образом, на выходе второго этапа получаем множество понятий, между которыми существуют не идентифицированные семантические отношения. Однако может оказаться так, что это множество содержит понятия, которые выходят за границы исследуемой предметной области.

3. Полученные на втором этапе концепты приводятся в нормальную форму, т.е. с помощью алгоритма стемминга выделяются словарные основы концептов.

4. Для каждого извлеченного концепта формируется терминологическое окружение. Терминологическое окружение создается путем извлечения терминов из страницы концепта wiki-ресурса и вычисления частоты встречаемости термина на данной странице по следующей формуле:

$$ntf_{t,d} = a + (1 - a) \frac{tf_{t,d}}{tf_{\max}(d)},$$

где $tf_{\max}(d) = \max_{\tau \in d} tf_{\tau,d}$ – максимальная величина tf в документе d , a – сглаживающий коэффициент, принимающий значение между нулем и единицей (экспериментально устанавливаются равной 0,4). Роль данного параметра состоит в уменьшении вклада второго члена. Нормировка частоты термина по максимуму предназначена для того, чтобы избежать следующей

аномалии [29]: в более длинных документах наблюдаются более высокие частоты терминов, так как в более длинных документах чаще содержатся повторяющиеся слова.

Идентификация типов отношений сведем к построению логических правил, определяющих, в каком отношении находятся понятия. Экспертом подготавливается корпус текстов предметной области, которые содержат набор предложений. Каждое такое предложение представляет собой некоторую ситуацию, в котором оказались концепты, свойственные тому или иному отношению, и имеют предикатное представление. В работе [32], [33] предлагается применять в основе отношений трехместные предикаты. Для отношения «обобщение» используется предикат $PHier(a, x, y)$, которое описывает отношение $род \leftrightarrow вид$, в котором значение первого аргумента равно $a \in \{Class, Kind\}$ и отношение $признак \leftrightarrow значение признака$ со значениями первого аргумента $a \in \{Category, Value\}$. Отношению «часть-целое» соответствует предикат $Pwp(a, x, y)$, в котором значение первого параметра равно $a \in \{Whole, Part\}$.

Определим отношение «ассоциация» между концептами, если между ними не существуют отношения «часть-целое» и «обобщение». Для распознавания отношений между концептами, в текстах предметной области, необходимо дополнительная информация об отношениях в виде термов-спутников, которые составляют устойчивые словосочетания с глаголом семантического отношения. Для отношения «обобщение» термы-спутниками являются: «к видам», «родом», «имя», «значения» и другие. Термы-спутники: «целое», «часть», «состав», «элементом» соответствуют отношению «целое-часть». В работе [31] рассмотрены различные виды конструкции с предикатами, выражающие отношения «часть-целое». Используя знания об отношениях, построим логические правила по их идентификации. Формально логическое правило выглядит следующим образом:

$$R = (S = \{P, \{C^W\}, F^C, \{G\}\}),$$

где S – ситуация которая описывается в предложении, $P = \{p\}$ – подмножество предложений из корпусов текста предметной области, C^W – концепты извлеченные из электронной библиотеки «Википедия», F^C – множество конструкции предикатов и терм спутников, G – подмножество описывающий результат исполнения логических правил.

2.3. Формирование нечетких контекстно-ориентированных запросов к электронному архиву

2.3.1. Классификация запросов на основе Байесовской модели

Рассмотрим алгоритм формирования проектных запросов, который используется интеллектуальным агентом проектировщика и позволяет применять его профиль для улучшения показателей точности и полноты запросов к электронным архивам технических документов проектной организации.

Пусть множество $\hat{w} = \{\hat{w}_1, \hat{w}_2, \dots, \hat{w}_n\}$ – есть множество ключевых слов проектного запроса к электронному архиву. Используя функцию $F_{DC} : \{T^D\} \rightarrow C$ для построения терминологических окружений понятий онтологии предметной области, данное множество отображается в нечеткое множество:

$$\tilde{I}_q = \{\mu_1 / c_1, \mu_2 / c_2, \dots, \mu_m / c_m\}, \quad (2.6)$$

как результат онтологического преобразования терминов в набор степеней выраженности понятий онтологии.

Для представления модели формирования проектных запросов на концептуальном уровне будем использовать наивный байесовский классификатор. Вероятность того, что понятие запроса c принадлежит классу $k \in \{C^+, C^-\}$, будем определять по формуле Байеса:

$$P(k|c) = \frac{P(c|k) \cdot P(k)}{P(c)},$$

где $P(c|k)$ – вероятность встретить понятие c среди всех понятий класса k ; $P(k)$ – безусловная вероятность понятия класса k в онтологии предметной области; $P(c)$ – безусловная вероятность понятия c в онтологии предметной области.

Наиболее вероятный класс для понятия запроса определяется, используя оценку апостериорного максимума:

$$k_{map} = \arg \max_{k \in K} \frac{P(c|k) \cdot P(k)}{P(c)}.$$

Поскольку $P(c) = const$ в рамках одной онтологии и учитывая, что

$$P(c|k) \approx P(w_1|k) \cdot P(w_2|k) \cdot \dots \cdot P(w_n|k) = \prod_{i=1}^n P(w_i|k),$$

получаем:

$$k_{map} = \arg \max_{k \in K} [P(k) \cdot \prod_{i=1}^n P(w_i|k)]. \quad (2.7)$$

Для больших документов количество множителей $P(w_i|k)$ в выражении (2.7) может быть большим, а, следовательно, возникает проблема исчезновения порядка вследствие перемножения большого количества малых чисел. Перепишем выражение (2.7) с учетом свойств логарифма:

$$k_{map} = \arg \max_{k \in K} [\log P(k) \cdot \sum_{i=1}^n \log P(w_i|k)].$$

Оценка вероятностей $P(k)$ и $P(w_i|k)$ выполняется на основе обучающей выборки, сформированной для каждой информационной потребности In_j^i . Вероятность класса будем записывать как:

$$P(k) = \frac{D_k}{D},$$

где D_k – количество документов, принадлежащих классу k и определяемое на основе результатов выполнения запросов проектировщика к электронному архиву; D – общее количество документов в обучающей выборке.

Величина $P(w_i|k)$ определяет вероятность встретить термин w_i среди терминов документов, принадлежащих классу k . Значение данной величины

будем определять с учетом того, что термин из окружения понятия, включенного в проектный запрос, может отсутствовать в документах анализируемого класса. Применяя метод аддитивного сглаживания (сглаживания Лапласа), получаем:

$$P(w_i | k) = \frac{f_{ik} + 1}{\sum_{i' \in V} (f_{ik'} + 1)},$$

где f_{ik} – частота встречаемости i -го термина в документах класса k ; V – терминологический словарь проектной организации (список всех уникальных терминов).

В результате классификации понятий запроса (2.6) понятия, принадлежащие к положительному классу понятий информационной потребности проектировщика, остаются в составе запроса. Понятия, которые классифицированы как отрицательные, исключаются из исходного запроса.

2.3.2. Нечеткая онтологическая модель поискового запроса

Поисковый запрос, сформулированный в терминах на естественном языке, преобразуется к онтологическому виду (2.6), но прежде выполняется ряд предобработок [35], [42], [47], [58], [62]:

- удаление стоп-слов,
- стемминг (выделение основы слова, получение термов).

С полученным запросом $Q = \{\hat{w}\}$ возможны два варианта обработки:

- 1) $Q = C$ – поисковый запрос полностью совпадает с названием концепта предметной онтологией
- 2) $Q = W$ – поисковый запрос совпадает с терминами предметной онтологией

В первом случае расширим поисковый запрос, используя функцию интерпретации концептов, т.е. дополним запрос терминами из терминологического окружения концепта:

$$Q = Q \cup F_{DC}(c_i).$$

Во втором случае применим функцию, сопоставляющую набору терминов подмножество понятий предметной области, получая множество концептов, в терминологическом окружении которых присутствуют термины запроса. К полученному множеству концептов применим функцию интерпретации концептов, дополняя запрос терминами, семантически связанными с концептами:

$$Q = Q \cup F_{DC}(F_{DC}^{-1}(\hat{w})).$$

Таким образом, расширение запроса сводится к заданию прямой и обратной функции интерпретации. Полученный после обработки расширенный запрос преобразуется к онтологическому виду (2.6).

Следующим шагом является исключение тех концептов, которые имеют для пользователя низкую степень принадлежности. Данный механизм использует байесовскую модель классификаций и профиль пользователя. Классификатор определяет степень принадлежности концепта к профилю пользователя. Если концепт имеет низкую степень принадлежности, то он исключается из проектно-контекстного запроса вида (2.6) и не используется в дальнейшем в процессе поиска ТД. Уточняя запрос с применением пользовательского профиля, тем самым достигается исключение влияния на результат поиска тех концептов, которые имеют отрицательную оценку в модели информационной потребности.

Таким образом, сформированный запрос представляется в виде графа:

$$G^q = \langle C^q, R^q \rangle \quad (2.8)$$

где C^q – множество вершин-понятий проекта, R^q – множество дуг, соединяющих вершины-понятия. Понятия в контекстно-ориентированном проектном запросе соединены двумя видами отношения: часть-целое и род-вид.

В случае достаточно большой онтологии и неявной принадлежности запроса конкретному фрагменту предметной области проектирования мощность множества C^q (количество понятий, включенных в нечеткое представление проектного запроса) может быть большой. Поэтому возникает необходимость в процедуре редукции запроса (2.8).

Способ редукции понятий проектного запроса основывается на разделении исходного графа $G^q = \langle C^q, R^q \rangle$ на несколько подграфов, каждый из которых содержит только вершины, соединенные дугами одной семантической категории. В данной работе используются две семантические категории: «обобщение» («isA») и «часть-целое» («part_of»). На рисунке 2.9 представлен иллюстративный пример графа проектного запроса вида (2.8), в котором присутствуют вершины $c_1, \dots, c_{14}$ – понятия онтологии предметной области и два типа дуг: дуги «isA» (штриховая стрелка) и дуги «part_of» (сплошная стрелка).

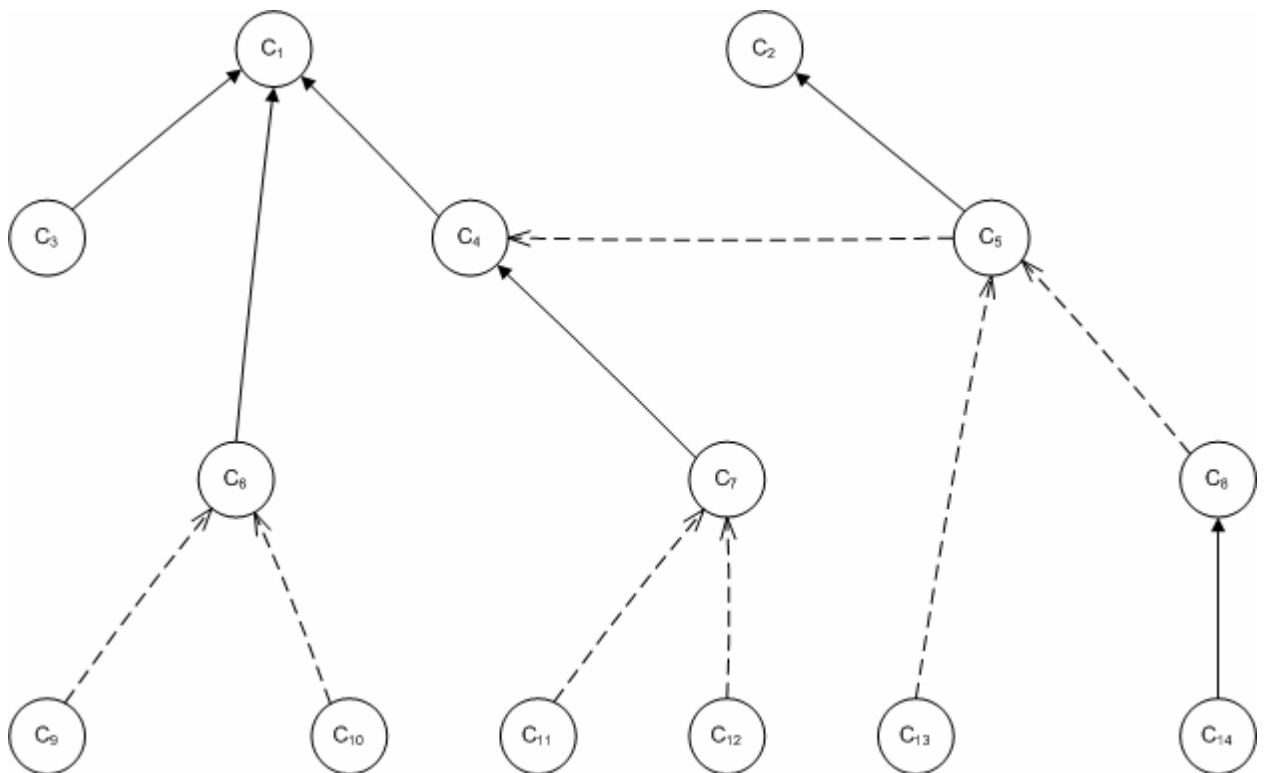


Рис. 2.5. Иллюстративное представление проектного запроса в виде графа

Редукция понятий графа проектного запроса предполагает выполнения ряда следующих шагов.

Шаг 1. Разбиение графа проектного запроса на несколько подграфов с учетом семантических категорий («isA» или «part_of»). На рисунке 2.6 показан подграф проектного запроса $G_{isA}^q = \langle C_{isA}^q, R_{isA}^q \rangle$, понятия которого связаны между собой отношениями типа «isA». Соответственно, на рисунке 2.7 представлен подграф $G_p^q = \langle C_p^q, R_p^q \rangle$ с отношениями «part_of».

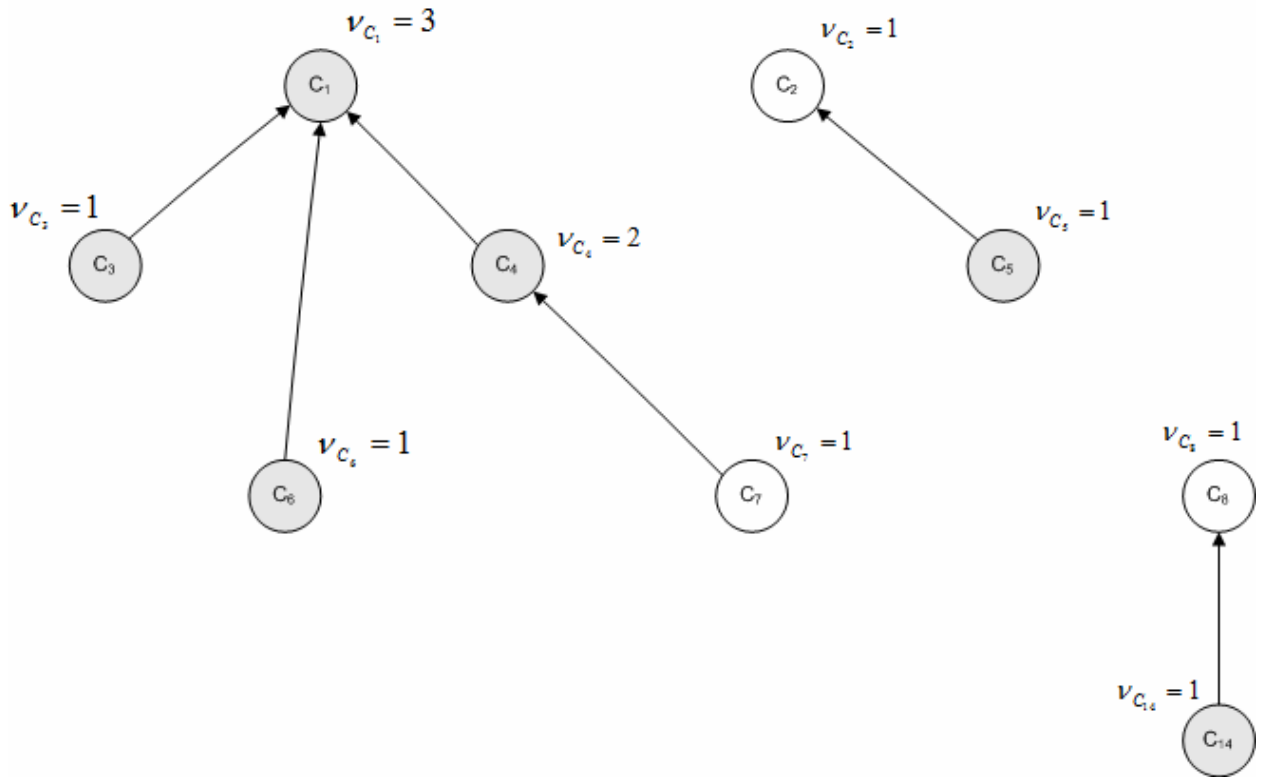


Рис 2.6. Подграф графа проектного запроса с типом отношений «isA»

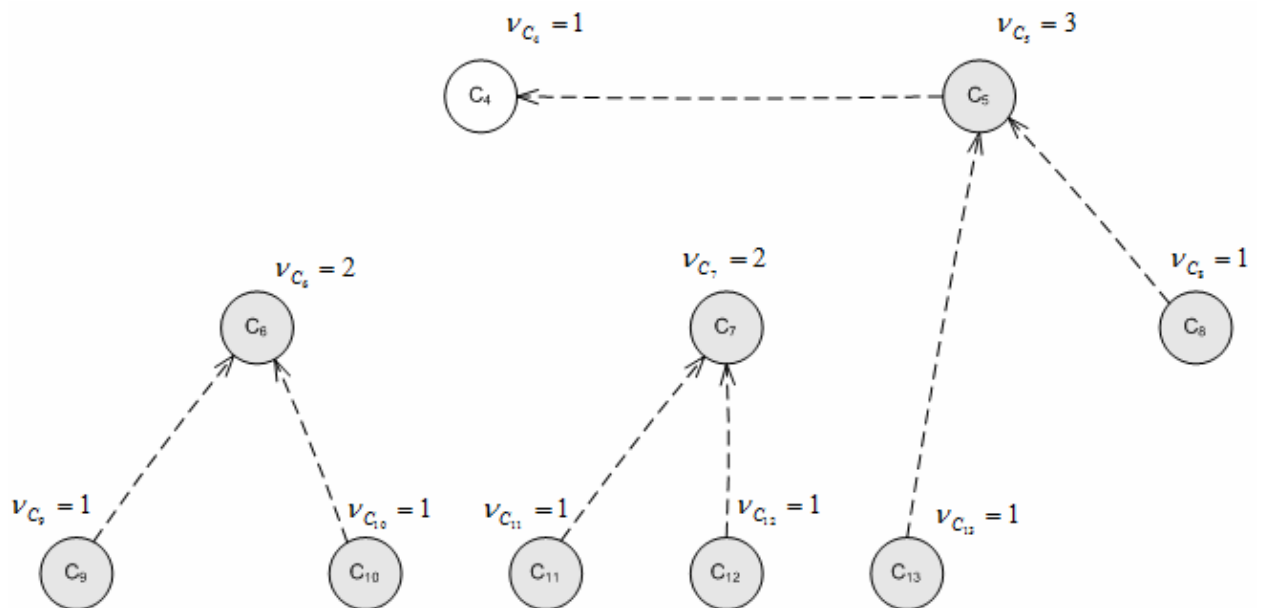


Рис 2.7. Подграф графа проектного запроса с типом отношений «part_of»

Шаг 2. Для каждой вершины подграфов определяется ее степень v_{C_j} (количество входящих и исходящих дуг). Как видно из рисунков наибольшие степени имеют понятия C_1 в подграфе «isA» и понятия C_5 , C_6 и C_7 в подграфе «part_of».

Шаг 3. В редуцированное множество понятий отдельно взятого подграфа проектного запроса включаются понятия согласно следующим правилам.

Правило 1: Если в подграфе существует вершина с максимальной степенью, то в результирующее множество соответствующего подграфа включается данная вершина и связанные с ней вершины, дуги от которых направлены к вершине с максимальной степенью.

Правило 2: Если подграф содержит две вершины, соединенные дугой, то в результирующее множество включается вершина с исходящей дугой.

Правило 3: Если подграф состоит из одной изолированной вершины, то данная вершина включается в результирующее множество понятий.

На рисунках 2.6 и 2.7 темным фоном отмечены те вершины подграфов, которые включены в результирующие множества понятий C_{isA}^{q+} и C_p^{q+} с использованием вышеприведенных правил.

Шаг 4. В редуцированное множество понятий проектного запроса включаются понятия, которые включены как во множество C_{isA}^{q+} , так и во множество C_p^{q+} : $C^{q+} = C_{isA}^{q+} \cap C_p^{q+}$.

Таким образом, после применения рассмотренных выше правил граф проектного запроса будет иметь вид, представленный на рисунке 2.8.

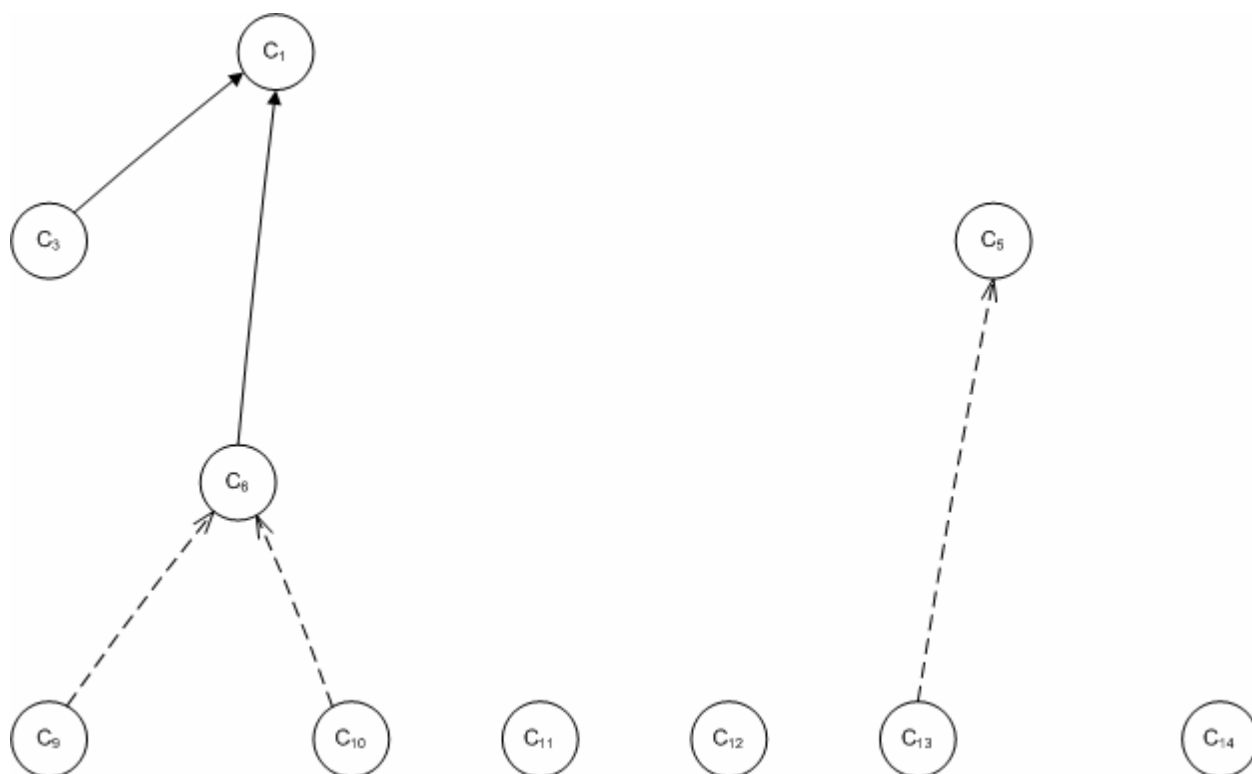


Рис 2.8. Результирующий граф проектного запроса после редукции вершин-понятий

В результате будет получен запрос, который содержит множество концептов, которые конкретизируют предметную область и имеют большое количество входящих или/и исходящих дуг.

2.3.3. Алгоритм формирования нечетких контекстно-ориентированных запросов к электронному архиву

Задачей информационной поддержки в электронном архиве является поиск множества технических документов $D = \{d_1, d_2, \dots, d_k\}$ с учетом текущей стадии проектирования АС и профиля проектировщика. Соответственно, объект поиска можно представить следующим выражением:

$$M_d = \{d_1, d_2, d_3, \dots, d_k \mid d_i = \{(c, \mu) \mid c \in C, \mu \in R\}\}. \quad (2.9)$$

В табличном представлении выражение (2.9) будет иметь следующий вид:

Таблица 2.1.

Онтологическое представление множества документов

Концепты Документы	c_1	c_2	c_3	$\dots$	c_n
d_1	μ_{11}	μ_{12}	μ_{13}	$\dots$	μ_{1n}
d_2	μ_{21}	μ_{22}	μ_{23}	$\dots$	μ_{2n}
d_3	μ_{31}	μ_{32}	μ_{33}	$\dots$	μ_{3n}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
d_k	μ_{k1}	μ_{k2}	μ_{k3}	$\dots$	μ_{kn}

где d_i – документ в электронном архиве $D = \{d_1, d_2, \dots, d_k\}$; k – количество документов в электронном архиве; c_i – концепты онтологии предметной области; n – количество концептов в онтологии предметной области; μ_{ij} – величина, характеризующая степень выраженности концепта в документе, где $0 \leq \mu_{ij} \leq 1$.

Для нахождения меры сходства между поисковым запросом и документом применяется следующее выражение:

$$similarity = \cos(\theta) = \frac{\sum_{i=1}^n \mu(c_i)_{I_q} \times \mu(c_i)_{I_d}}{\sqrt{\sum_{i=1}^n \mu(c_i)_{I_q} \times \sum_{i=1}^n \mu(c_i)_{I_d}}}, \quad (2.10)$$

где c_i – понятие из словаря концептов предметно-ориентированной онтологии; $\mu_{I_q}(c_i), \mu_{I_d}(c_i)$ – степень выраженности концепта в поисковом запросе и документе соответственно.

Таким образом, мера сходства $\cos(\theta)$ – является величиной, с помощью которой определяется релевантность документа запросу.

Алгоритм нечеткого онтологического информационного поиска состоит из ряда этапов [43], [45], [57], [59], [62], [67]:

Шаг 1. Инициализация

1. Формирование пользовательского запроса в виде онтологии (2.6).
2. Нормализация вектора запроса следующим образом:

$$Q = \{c_i, \mu_i / \max(\mu)\},$$

где c_i – концепт в онтологии предметной области; μ_i – степень выраженности концепта в информационном запросе; $\max(\mu)$ – максимальное значение степени выраженности концепта в онтологическом представлении запроса.

Шаг 2. Вычисление меры включения запроса $\tilde{I}_q$ в документ $\tilde{I}_d$

Вычисляется степень включения онтологического представления запроса (2.6) в онтологическое представление документа из множества проектных документов (2.9). Данная операция выполняется для каждого документа из множества. Степень включения запроса в документ вычисляется с помощью выражения (2.10).

Шаг 3. Ранжирование документов

Ранжирование множества документов, полученных на втором этапе в порядке убывания значения $\cos(\theta)$.

Алгоритм формирования нечетких контекстно-ориентированных запросов к электронному архиву представлен на рисунке 2.9.

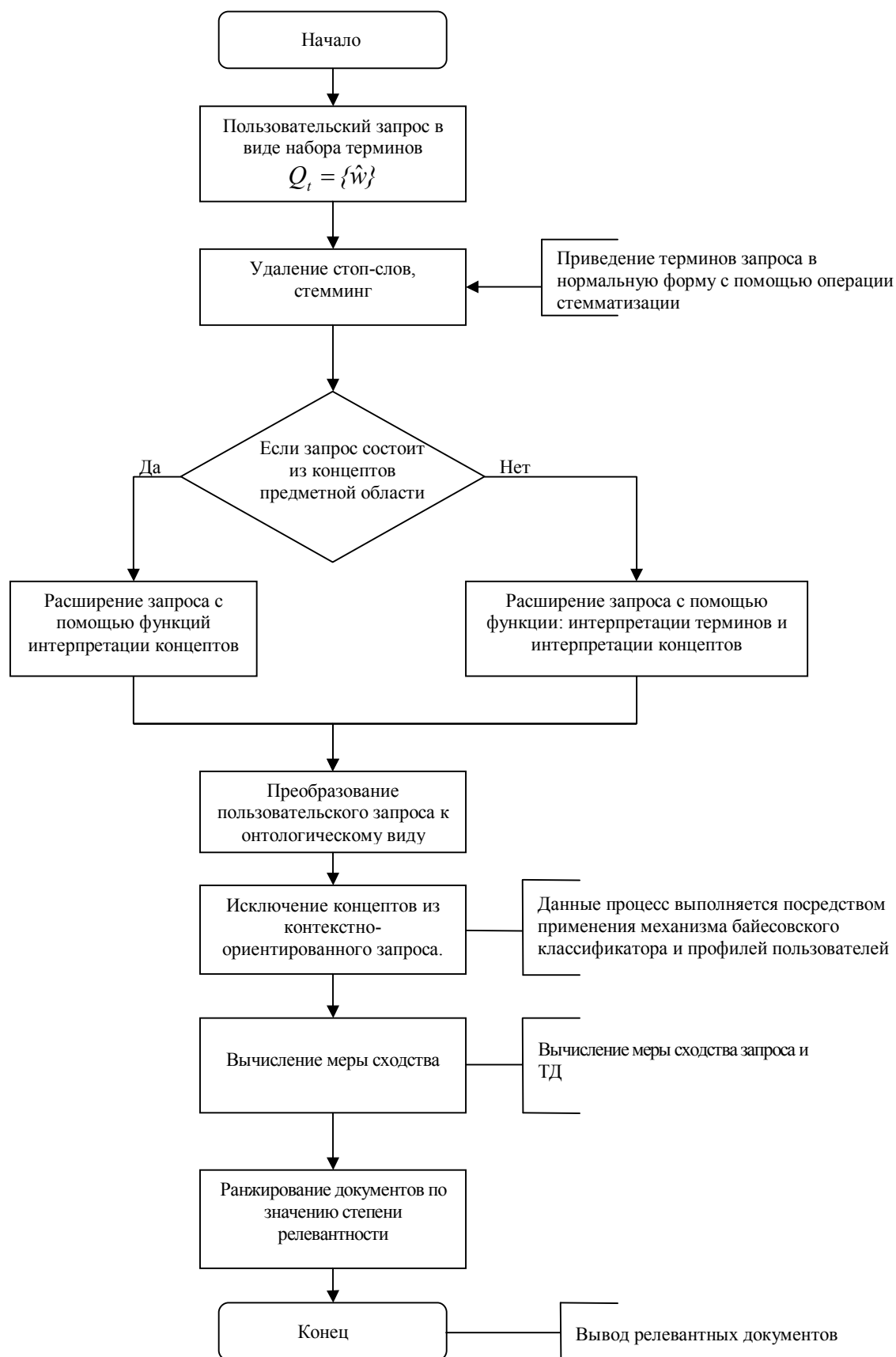


Рис. 2.9 Онтологическая модель информационного поиска

2.4. Методика интеллектуальной информационной поддержки проектирования АС

Реализация интеллектуальной информационной поддержки процесса проектирования АС невозможна без четкого плана мероприятий, который является частью соответствующей методики. Современное предприятие (проектная организация) имеет в составе корпоративной системы электронный архив технической документации с развитой системой «человеко-машинного» взаимодействия. Включение в электронный архив интеллектуальной компоненты информационной поддержки должно выполняться поэтапно, не нарушая общепринятых производственных процессов и реализованных архитектурных решений.

В таблице 2.2 представлены основные процедуры процесса интеллектуализации информационной поддержки проектирования АС, включающие в себя формирование онтологии, разработку индивидуальных профилей проектировщиков и интеграцию разработанных онтологических ресурсов в систему электронного архива проектной организации.

Таблица 2.2.

Процедуры интеллектуализации информационной поддержки

№	Вход	Процедура	Выход
1		Формирование онтологического ресурса проектной организации	
1.1	Объекты проектирования, проектные процедуры, нормативно-справочная документация	Определение онтологического базиса формируемого онтологического ресурса (онтологий)	Онтологический базис (классы, отношения, функции интерпретации)
1.2	ГОСТы, документы электронного архива	Разработка онтологий	Онтология методологии проектирования, Онтология предметной области
1.3	Терминологические словари организации, распределенные wiki-ресурсы	Разработка концептуальных сетей (организации, проектов)	Множество концептуальных сетей

2		Разработка системы индивидуальных профилей проектировщиков	
2.1	Электронный архив, методология проектирования организации	Определение тестового множества технических документов (релевантных и нерелевантных) для каждого отдельного проектировщика с учетом текущей проектной стадии	Множество релевантных и нерелевантных документов
2.2	Онтология предметной области, множество релевантных и нерелевантных документов	Выполнение онтологического отображения терминов документов на концепты онтологии предметной области	Положительные и отрицательные подмножества понятий онтологии предметной области
3		Интеграция системы профилей, онтологического ресурса с электронным архивом технической документации	
3.1	Онтология методологии проектирования, электронный архив,	Настройка системы определения текущего состояния реализуемого проекта в электронном архиве	Модифицированная онтология методологии проектирования
3.2	Онтологические ресурсы, электронный архив, система профилей проектировщиков	Включение системы профилей проектировщиков как медиатора для существующего электронного архива	Интеллектуальный электронный архив технических документов

Формирование онтологического ресурса проектной организации начинается с определения онтологического базиса, который остается неизменным до формирования новой онтологии. Задача разработки онтологии нацелена на создание онтологии методологии проектирования и онтологии предметной области. Данные ресурсы являются разделяемыми для большого количества проектировщиков, возможно, включающихся в сетевые распределенные команды. Разработка концептуальных сетей проектной организации и реализуемых проектов является более узкой задачей и привязана к деятельности конкретной организации и (или) конкретных проектов.

Целью разработки системы индивидуальных профилей проектировщиков является персонализация выполнения поисковых проектных запросов к электронному архиву технических документов для повышения точности и полноты результатов. Кроме того, такой подход к организации информационного взаимодействия специалистов и электронного архива позволяет накапливать опыт проектирования на концептуальном уровне.

Интеграция системы профилей, онтологического ресурса с электронным архивом технической документации предполагает комплекс работ по включению реализованных онтологических ресурсов в действующую автоматизированную систему управления базой технических документов.

2.5. Выводы по главе

1. Представленная в главе структурно-функциональная модель онтологии информационной поддержки проектирования АС отличается от известных моделей наличием многоуровневой структуры, в которой присутствуют знания о методологии проектирования АС, знания предметной области проектной организации и знания о применяемых терминосистемах.
2. Описанная система индивидуальных профилей проектировщиков позволяет выполнять контекстно-ориентированные проектные запросы к электронному архиву технических документов, основываясь на особенностях предыдущего опыта выполнения запросов. Модель формирования индивидуальных профилей основывается на байесовском классификаторе и прикладной онтологии.
3. Нечеткая модель контекстно-ориентированных запросов использует меру сходства контекста запроса в техническом документе. Указанная модель предполагает, что отдельно взятый технический документ имеет онтологическое представление в виде нечеткого множества,

базовое множество которого образуется системой понятий онтологии предметной области, а функция принадлежности – степенью выраженности понятий в техническом документе.

4. Представленная в главе методика интеллектуальной информационной поддержки проектирования АС предполагает интеграцию системы профилей, онтологического ресурса с электронным архивом технической документации. Предлагается такую интеграцию выполнять на онтологическом уровне с привлечением дополнительных знаний, извлеченных из распределенных wiki-ресурсов.

Глава 3

Разработка интеллектуальной системы информационной поддержки проектирования АС

3.1. Структурно-функциональное решение программной системы

Интеллектуальная система информационной поддержки проектирования АС – это программная система, предназначенная для систематизации и автоматизации работы с электронным архивом ТД, учитывающая состояние онтологии предметной области.

Основными функциями данной системы являются:

- Онтологически-ориентированное индексирование технических документов. Формирование онтологических представлений ТД.
- Хранение онтологии предметной области и онтологических представлений ТД.
- Формирование и хранение профилей пользователей, учитывающих информационную потребность субъекта проектирования на разных стадиях проектирования АС.
- Формирование контекстно-ориентированных проектных запросов к электронному архиву.
- Извлечение технических документов из электронного архива с применением онтологических профилей пользователей.

Интеллектуальная система информационной поддержки проектирования включена в состав электронного архива ФНПЦ АО «НПО

«Марс», структурная схема которого представлена на рисунке 3.1. В данном электронном архиве существует несколько ролей:

- Архивариус – задачами пользователя является заполнение архива электронными копиями технических документов;
- Аналитик – пользователь ведет учет статистической информации о работе электронного архива;
- Пользователь – конечный потребитель содержимого электронного архива.

Система электронного архива включает в себя систему разграничения прав доступа, которая определяет группы и/или отдельные документы, которые выдаются системой информационной поддержки.

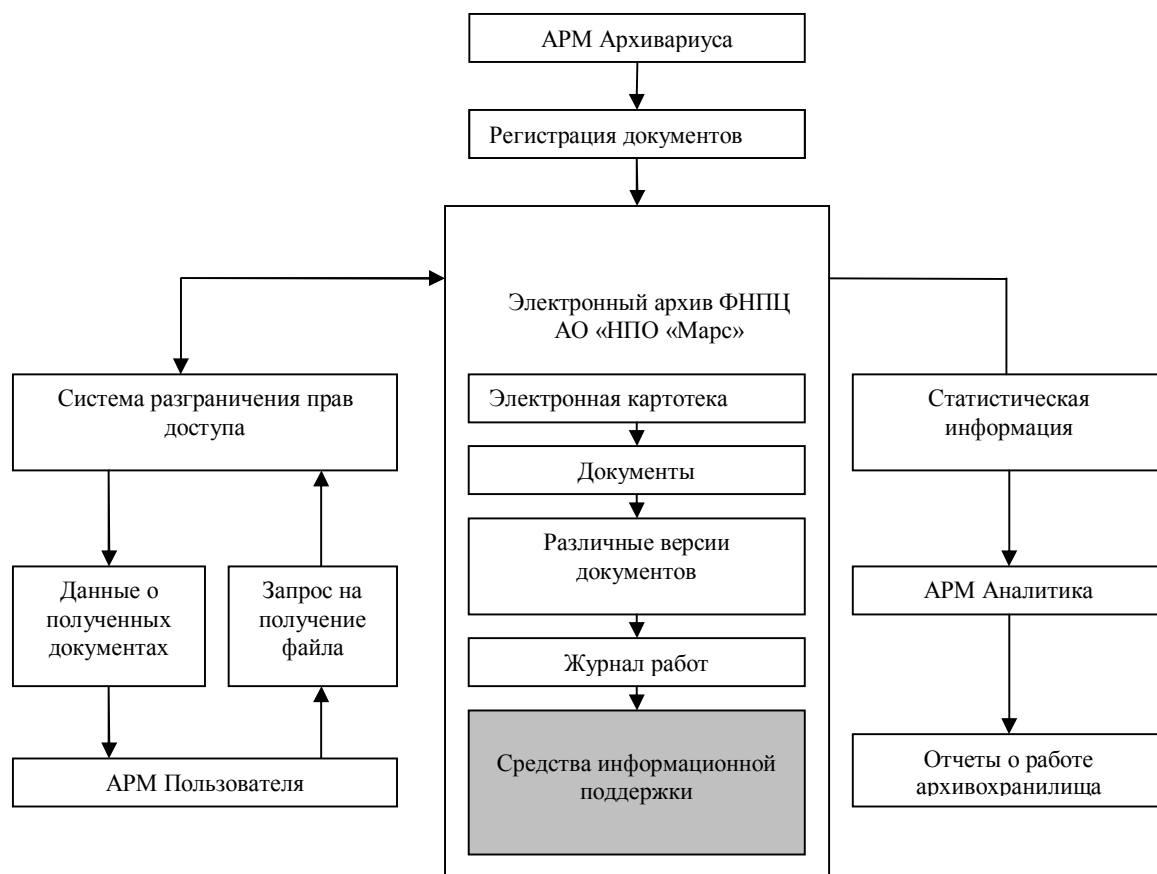


Рис. 3.1. Структура электронного архива ФНПЦ АО «НПО «Марс»

Электронный архив ФНПЦ АО «НПО «Марс» содержит множество ТД, которые используются подсистемой информационной поддержки в качестве входных данных. Технические документы используются в процессе индексирования [65], [66], полученные наборы индексов хранятся в

который является контейнером для хранения RDF-троек, которые могут быть простыми объектами Java, либо реляционная база данных.

RDF – это разработанная консорциумом W3C модель для представления данных, в особенности – метаданных [50]. Основной структурной единицей RDF является коллекция троек (или триплетов), состоящих из субъекта, предиката и объекта (S,P,O). Набор триплетов называется RDF-графом. В качестве вершин графа выступают субъекты и объекты, в качестве дуг – предикаты (или свойства). Направление дуги, соответствующей предикату в данной тройке (S,P,O), всегда выбирается так, чтобы дуга была направлена от субъекта к объекту.

Для описания атрибутов и отношений в RDF используется RDF Schema (RDFS), которая определяет классы, свойства и другие ресурсы. RDFS является семантическим расширением RDF. Все определения RDFS выражены на RDF. Новые термины, такие как «домен» и «диапазон» свойства, являются ресурсами RDF. Ресурсы объединяются в группы – классы. Члены класса называются экземплярами класса. Для указания того, что ресурс является экземпляром, используется соответствующее свойство. Определения классов и свойств отделены от множества экземпляров класса и значения свойств.

Фрагменты онтологии предметной области в формате RDF:

Фрагмент онтологий методологии проектирования.

```
<St rdf:ID="Начало">
<ConnectStToTp rdf:resource="#Требования_к_АС" />
<ConnectStToTp rdf:resource="#Техническое_задание" />
<ConnectStToTp rdf:resource="#Заявка_на_разработку_АС" />
</ St >
< St rdf:ID="Проектирование">
<Before rdf:resource="#Начало" />
<ConnectStToTp rdf:resource="#Эскизный_проект" />
<ConnectStToTp rdf:resource="#Технический_проект" />
</St >
...
```

```

<St rdf:ID="Внедрение">
  <Before rdf:resource="#Построение" />
  <ConnectStToTr
rdf:resource="#Рабочая_документация_пользователю_АС" />
</St>

```

Фрагмент терминологической сети проекта

```

<InstanceTerm rdf:ID="CIndex22092">
  <ConnectToInstance rdf:resource="#Техническое_задание"
/>
  <ConnectToTerm rdf:resource="#глобализова" />
  <ConnectToFreq rdf:datatype="float">0.41</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex22094">
  <ConnectToInstance rdf:resource="#Техническое_задание"
/>
  <ConnectToTerm rdf:resource="#техническ" />
  <ConnectToFreq rdf:datatype="float">0.41</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex22095">
  <ConnectToInstance rdf:resource="#Техническое_задание"
/>
  <ConnectToTerm rdf:resource="#задан" />
  <ConnectToFreq rdf:datatype="float">0.41</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex22096">
  <ConnectToInstance rdf:resource="#Техническое_задание"
/>
  <ConnectToTerm rdf:resource="#техзадан" />
  <ConnectToFreq rdf:datatype="float">0.41</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex22097">
  <ConnectToInstance rdf:resource="#Техническое_задание"
/>
  <ConnectToTerm rdf:resource="#исходн" />
  <ConnectToFreq rdf:datatype="float">0.54</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex22098">
  <ConnectToInstance rdf:resource="#Техническое_задание"
/>

```

```

    <ConnectToTerm rdf:resource="#документ" />
    <ConnectToFreq rdf:datatype="float">0.55</ConnectToFreq>
  </InstanceTerm>

```

На рисунках 3.3. и 3.4. изображены структурные представления RDF-схемы, а также фрагменты схем концептуальной сети проекта и методологий проектирования.

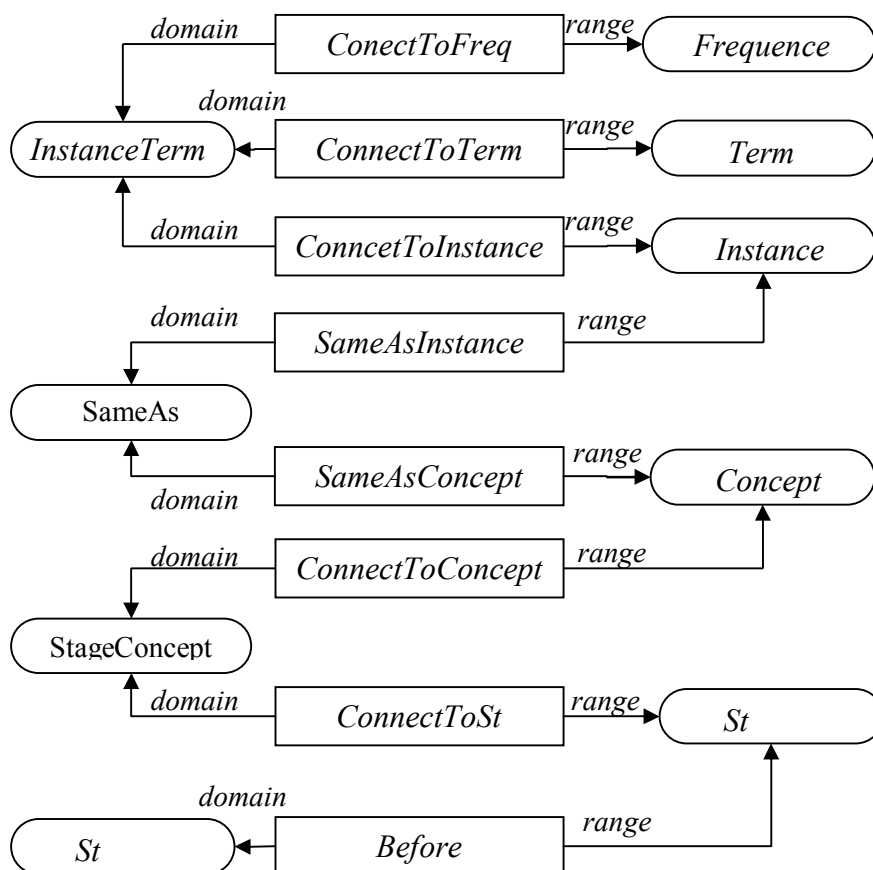


Рис. 3.3. Графическое представление RDF-схемы методологии проектирования

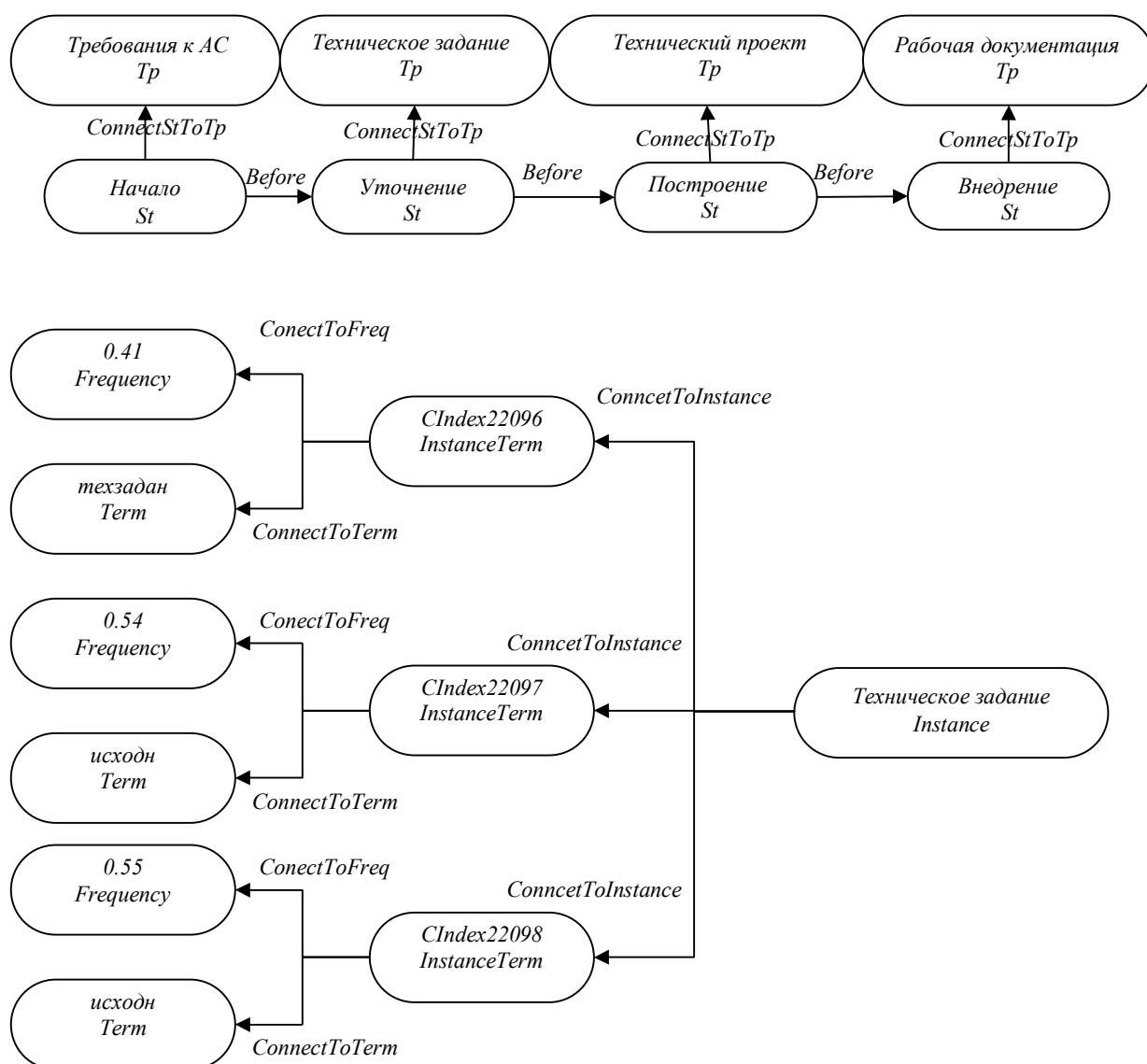


Рис. 3.4. Графическое представление фрагмента онтологии

3.2.2. Описание электронной библиотеки Wiki-ресурсов

«Википедия» — свободная общедоступная мультязычная универсальная интернет-энциклопедия, реализованная на принципах Wiki, характеризуется такими признаками:

- Возможность многократно править текст посредством самой вики-среды (сайта), без применения особых приспособлений на стороне редактора.

- Особый язык разметки — так называемая вики-разметка, которая позволяет легко и быстро размечать в тексте структурные элементы и гиперссылки, форматировать и оформлять отдельные элементы.
- Учёт изменений (версий) страниц: возможность сравнения редакций и восстановления ранних.
- Проявление изменений сразу после их внесения.
- Разделение содержимого на именованные страницы.
 - Гипертекстовость: связь страниц и подразделов сайта через контекстные гиперссылки.
- Множество авторов. Некоторые вики могут править все посетители сайта.

Язык вики поддерживает механизм гиперссылок для создания связей между вики-страницами и является более наглядным, чем HTML, и более безопасным, поскольку использование JavaScript и каскадных таблиц стилей ограничено.

3.3. Описание функциональных возможностей подсистемы информационной поддержки

Подсистема онтологически-ориентированного информационного поиска технических документов

Процесс онтологического поиска состоит из ряда этапов:

1. обработка пользовательского запроса;
 - 1.1. удаление стоп-слов;
 - 1.2. стемминг (выделение основы слова);
 - 1.3. расширение пользовательского запроса путем использования онтологии предметной области;
 - 1.4. преобразование запроса к онтологическому виду, посредством вычисления степени выраженности концептов в запросе;

- 1.5. исключение концептов, которые имеют отрицательную потребность для пользователя электронного архива.
2. загрузка индексов документов;
3. вычисление степени схожести между контекстно-ориентированным запросом и техническим документом, данная величина характеризует релевантность документа пользовательскому запросу;
4. сортировка документов по убыванию степени релевантности.

В качестве исходных документов используются онтологические представления индексов документов [60], [61]. Индексы ТД хранятся в ЭА и загружаются в процессе выполнения проектных запросов. При приведении пользовательского запроса к онтологическому виду и исключения концептов, которые оказывают негативное влияние на удовлетворение информационной потребности пользователя, из контекстно-ориентированных запросов используется онтология предметной области, которая хранится на сервере онтологии Sesame.

На рисунке 3.5. показано основное окно модуля запросов системы информационной поддержки проектировщика. Поисковый запрос вводится на естественном языке, далее данный запрос расширяется либо с помощью средств автоматического расширения запроса, либо пользователь сам расширяет запрос с помощью добавления термов, которые представлены в интерактивных окнах.

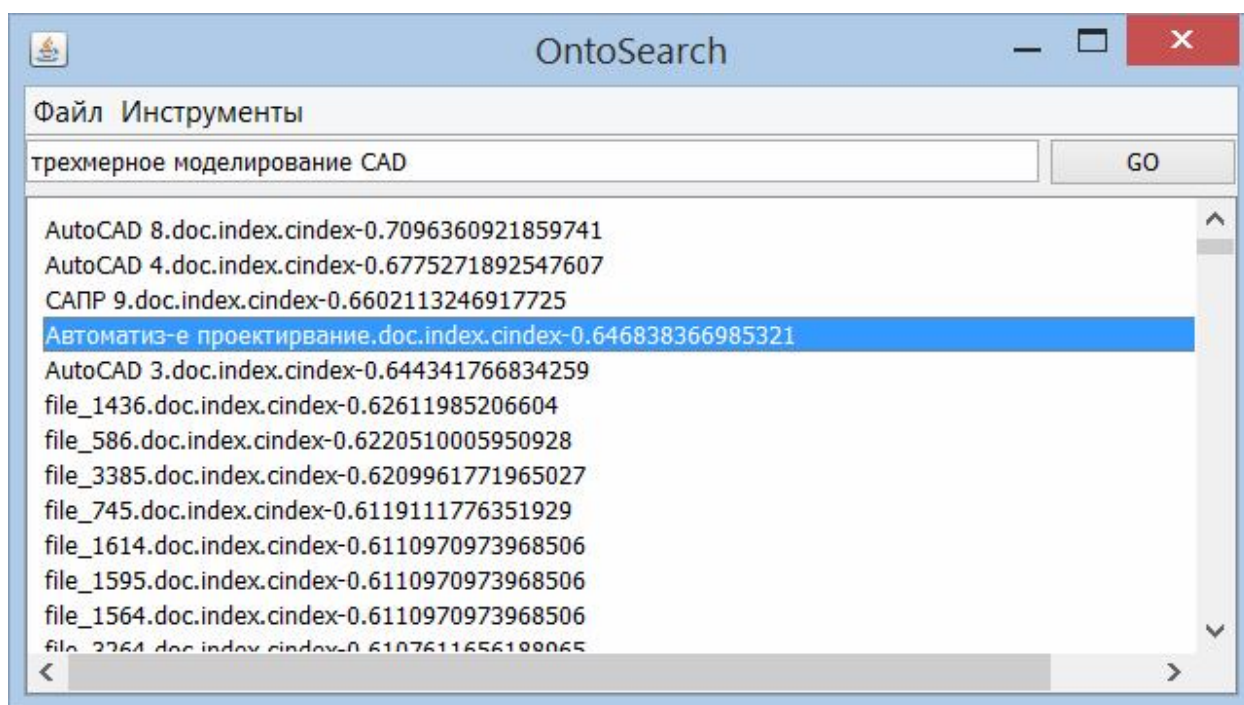


Рис. 3.5. Основное окно модуля запросов

На рисунке 3.6. представлено интерактивное окно, которое выводит концепты из предметной онтологии, имеющие высокую степень выраженности в пользовательском запросе, с терминологическим окружением.

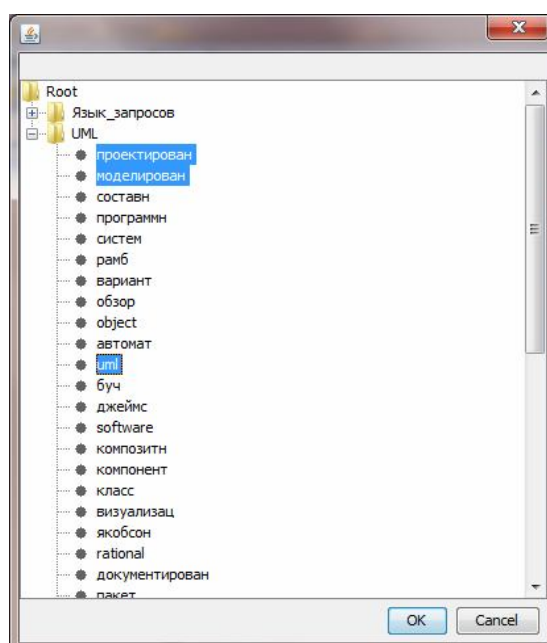


Рис. 3.6. Интерактивное окно подсказки для расширения пользовательского запроса

Представленное на рисунке 3.6. интерактивное окно позволяет отобразить пользователю термины и тем самым расширить свой запрос.

"Техническая документация".

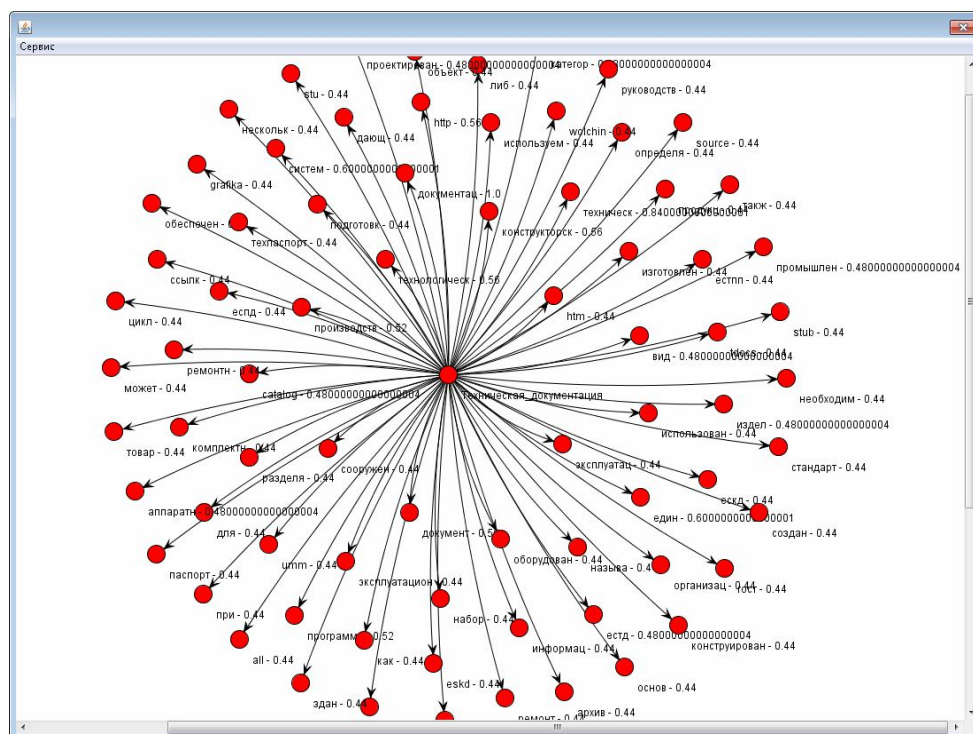


Рис 3.9 Концепт с терминологической сетью.

На рис. 3.10. представлена матрица смежности концептуальной сети.

Концепты...	Распредел...	Data Manip...	Секцион...	Объектны...	Иерархич...	База данных	Проектир...	База знаний	Клиентска...	Data Defini...	Веб-портал
Распределё...	-	0	0	0	0	1.0	0	0	0	0	0
Data Manipu...	0	-	0	0	0	1.0	0	0	0	0	0
Секциониро...	0	0	-	0	0	1.0	0	0	0	0	0
Объектные...	0	0	0	-	0	1.0	0	0	0	0	0
Иерархичес...	0	0	0	0	-	1.0	0	0	0	0	0
База данных	1.0	1.0	1.0	1.0	1.0	-	1.0	1.0	1.0	1.0	1.0
Проектиров...	0	0	0	0	0	1.0	-	0	0	0	0
База знаний	0	0	0	0	0	1.0	0	-	0	0	0
Клиентская...	0	0	0	0	0	1.0	0	0	-	0	0
Data Defini...	0	0	0	0	0	1.0	0	0	0	-	0
Веб-портал	0	0	0	0	0	1.0	0	0	0	0	-

Рис. 3.10. Матрица смежности.

3.4. Описание проекта программной системы

Программная система реализована на языке программирования Java в среде разработки Eclipse. Преимущество программы, разработанный на Java, является его кроссплатформенность, т.е. возможность исполнения программы на различных операционных системах, если существует реализация Java-машины.

В процессе разработки программной системы были использованы следующие компоненты:

- Swing – библиотека для создания графического интерфейса для программ Java.
- SAX (Simple API for XML) – набор классов для работы с файлам XML.
- Sesame API – набор библиотек и классов для работы с сервером онтологий SESAME.
- Apache POI – набор библиотек и классов для работы с файлами Microsoft Office.
- Snowball – набор классов, в которых реализован алгоритм стемматизации Портера для русского языка.
- Jung – набор классов, в которых реализованы средства визуализации графов.
- Bliki – библиотека, в которой реализованы функции для доступа к электронной библиотеке «Википедия».

3.4.1. Иерархия классов

На рисунке 3.11. представлены основные классы программной системы, в которых реализованы функционалы следующих подсистем: подсистема онтологического поиска ТД, подсистема формирования контекстно-ориентированных проектных запросов, подсистема формирования информационной потребности пользователя ЭА. Помимо основных функций также разработаны классы, которые содержат методы и

функции: доступа к хранилищу онтологий Sesame, формирование терминологической сети проекта с использованием Wiki-ресурсов, визуализация концептуальной сети средствами графической библиотеки Jung.

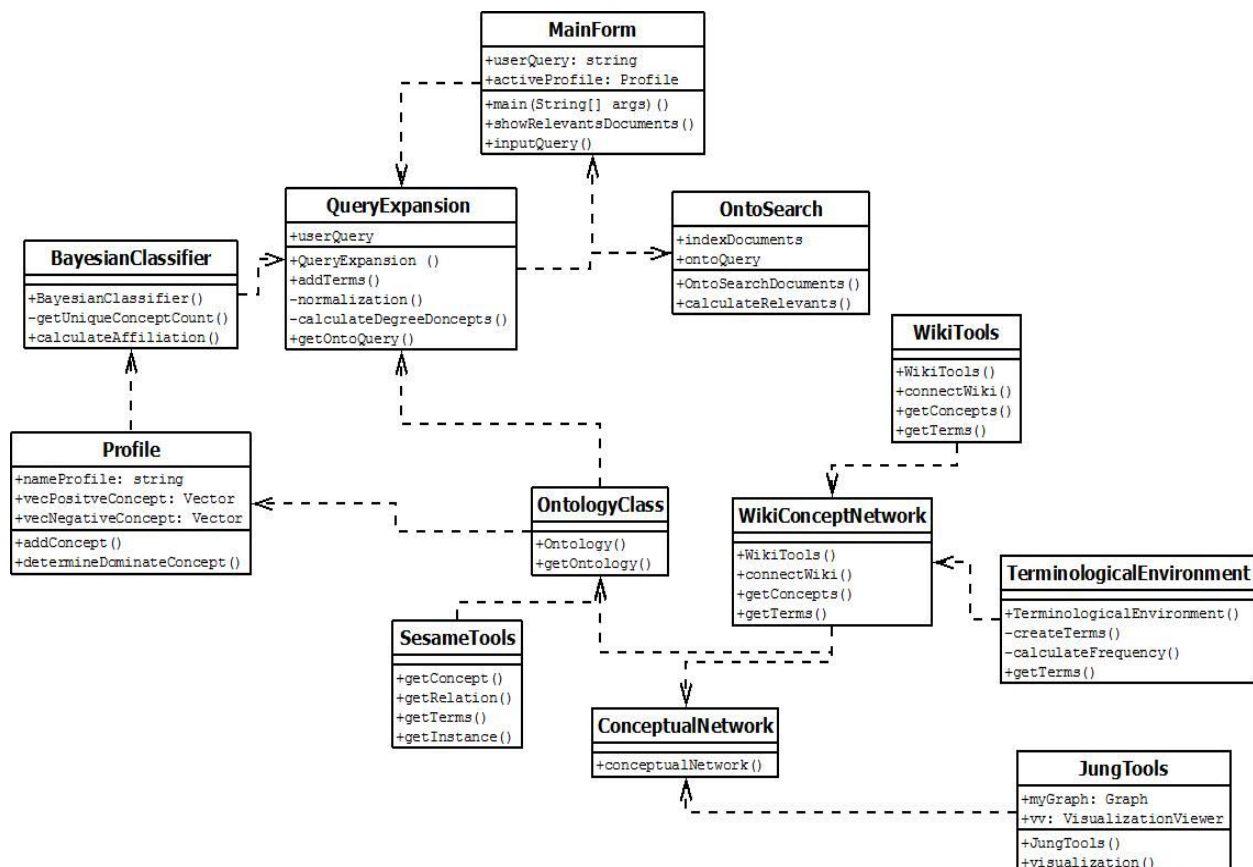


Рис. 3.11. Основные классы программной системы

Описание основных классов программной системы:

1. MainForm – класс, реализующий основное интерфейсное окно программы, в котором вводится поисковый запрос на естественном языке и выводится список релевантных документов.
2. QueryExpansion – класс, осуществляющий преобразование пользовательского запроса к контекстно-ориентированному виду. В процессе преобразования используется результат работы класса байесовского классификатора, вычисляется степень выраженности каждого концепта из предметной онтологии в контекстно-ориентированном запросе. Результат работы данного класса поступает на вход класса OntoSearch.

3. BayesianClassifier – класс, реализующий функционал классификации концепта. Определяет степень принадлежности концепта пользовательскому профилю.
4. Profile – класс, осуществляющий формирование пользовательского профиля. В данном классе реализованы функции: определение доминирующего концепта в ТД, хранение документов и доминирующих концептов в векторах.
5. OntoSearch – класс, реализующий вычисления меры сходства между проектным запросом и множеством ТД из ЭА. Результатом работы данного класса является формирование списка релевантных документов, отсортированных в порядке убывания значения меры сходства.
6. SesameTools – класс, предоставляющий функционал для работы с сервером онтологий Sesame.
7. OntologyClass – класс, который содержит поля и методы для работы с онтологией предметной области. Данный класс активно взаимодействует в процессе работы программы с классами Profile и QueryExpansion.
8. WikiConceptNetwork – класс, формирующий концептуальную сеть проекта, и устанавливающий семантические отношения между концептами на основе анализа коллекций документов предметной области.
9. WikiTools – класс, предоставляющий функционал для работы с wiki-ресурсом. С помощью механизма гиперссылок извлекает концепты и содержимое страницы из электронной библиотеки.
10. TerminologicalEnvironment – класс, который формирует для каждого концепта, извлеченного из электронной библиотеки «Википедия», терминологическое окружение.
11. ConceptualNetwork – класс, реализующий вывод концептуальной сети на пользовательский экран.

12. JungTools – класс, содержащий методы и функции для графической визуализации концептуальной сети.

3.4.2. Основные алгоритмы программной системы

В данном разделе рассматриваются блок-схемы основных алгоритмов. На рисунке 3.12. представлен алгоритм выполнения онтологически-ориентированного информационного поиска. Данный алгоритм состоит из ряда последовательных этапов, выполняющих предобработку пользовательского запроса и вычисления меры сходства запроса в документе. Предобработка запроса выполняет преобразование запроса, введенного пользователем на естественном языке к контекстно-ориентированному виду, и включает в себя следующие действия:

1. удаление стоп-слов;
2. стемматизация слов (выделение словарных основ);
3. расширение запроса, которое достигается либо за счет интерактивных действий пользователя, либо за счет алгоритма в зависимости от выбора настройки;
4. вычисление степени выраженности концептов предметной области в расширенном запросе и представление запроса в онтологическом виде.
5. исключение концептов, которые имеют оценку отрицательной информационной потребности для пользователя.

Между контекстно-ориентированным запросом и индексами ТД вычисляется мера сходства. На основании данной величины происходит ранжирование документов по убыванию и вывод пользователю результатов поиска.

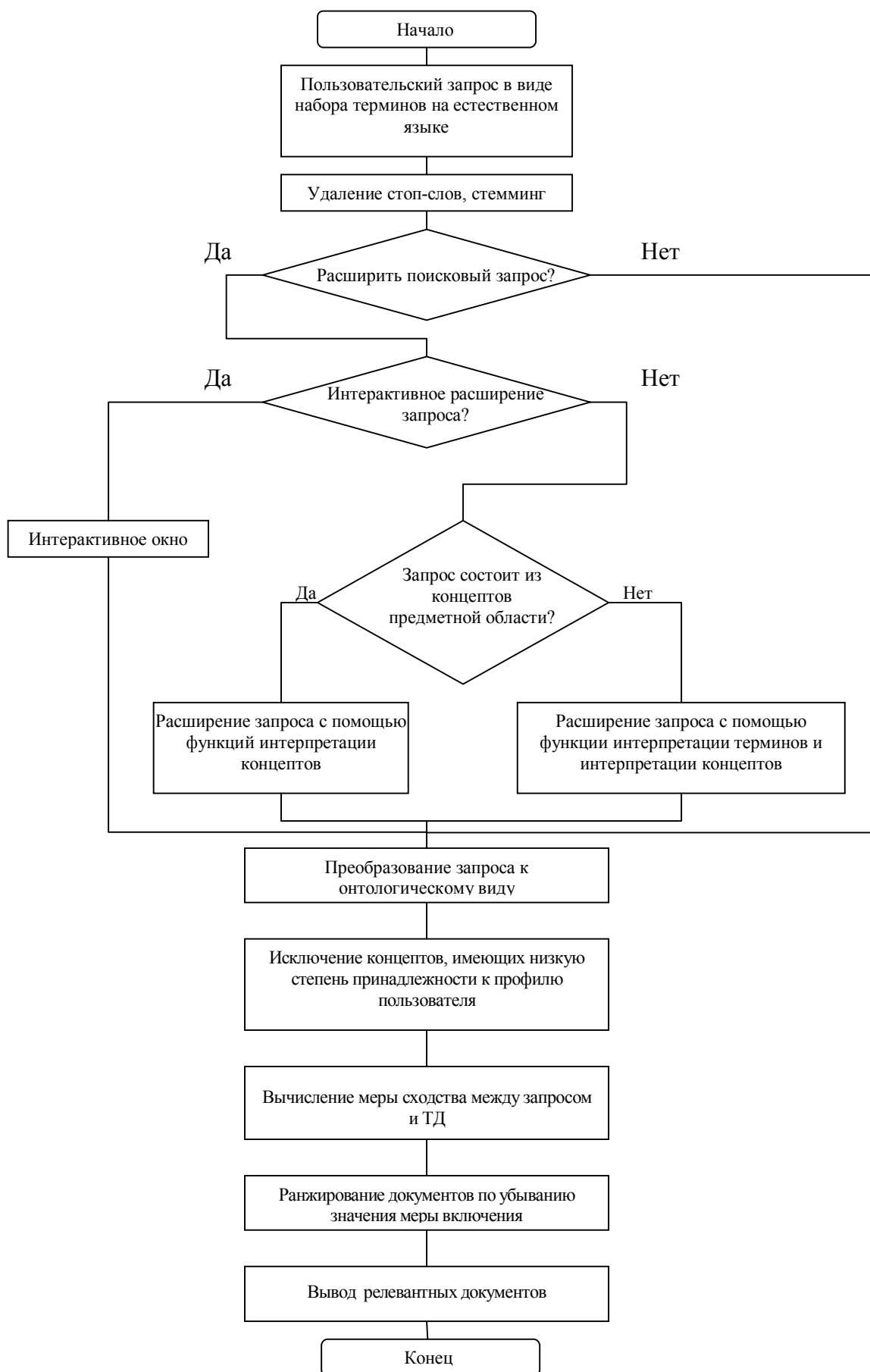


Рис. 3.12. Блок-схема онтологически-ориентированного поиска ТД

На рисунке 3.13. представлен алгоритм автоматизированного формирования онтологии предметной области с помощью электронного ресурса «Википедия». В начале алгоритма выполняется загрузка термов технических документов электронного архива и корпуса текстов предметной области, подготовленные экспертом. Далее вводятся первичные концепты, которые отбираются экспертом предметной области из стандартов и онтологий предметной области электронного архива. Данные концепты представляют собой входные параметры для алгоритма формирования концептуальной сети.

Данный алгоритм представляет собой цикл, в котором выполняется анализ страниц «Википедии». В процессе анализа извлекаются понятия и формируется список гиперссылок. Полученный список гиперссылок понятий добавляется в онтологию, если выполняется условие: существует обратная гиперссылка на страницу, на которой найдена данная страница. Условием выхода из цикла является нахождение минимального расстояния между первичными понятиями, которые ввел эксперт. Так как полученная концептуальная сеть может содержать концепты, которые не относятся к предметной области, то выполняется проверка и удаление тех концептов, которые отсутствуют среди термов документов электронного архива. Затем между оставшимися концептами определяются семантические отношения на основе логических правил и анализа корпуса текстов предметной области, которые были подготовлены экспертом, после чего онтология визуализируется в виде графов.

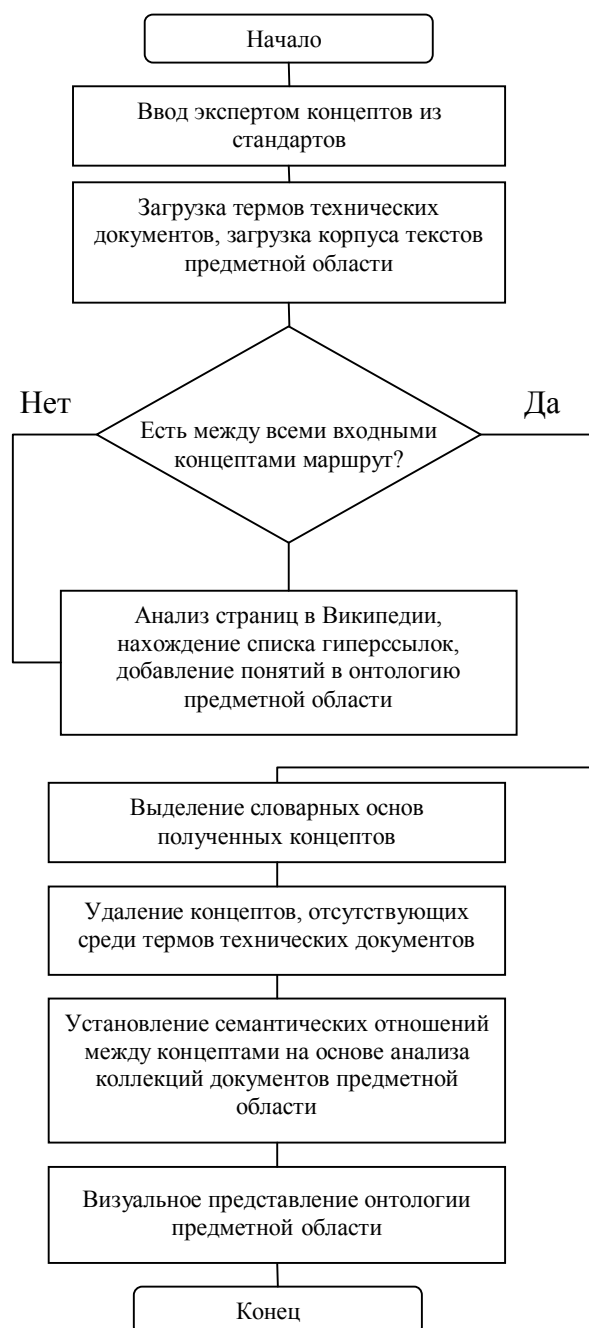


Рис 3.13. Блок-схема автоматизированного формирования онтологии предметной области

3.5. Выводы по главе

В процессе диссертационного исследования была разработана интеллектуальная программная система, состоящая из следующих подсистем:

- Подсистема интеллектуального расширения запроса и преобразование его к контекстно-ориентированному виду;

- Подсистема онтологически-ориентированного информационного поиска, учитывающая информационную потребность субъекта проектирования и стадию проектирования АС;
- Подсистема формирования индивидуального профиля проектировщика;
- Подсистема автоматизированного формирования онтологии предметной области с использованием внешних профессиональных ресурсов;
- Подсистема формирования терминологического окружения концептов предметной онтологии;
- Подсистема установления семантических отношений между концептами предметной онтологии.

Глава 4

Анализ адекватности разработанных моделей и методов на основе вычислительных экспериментов

План вычислительных экспериментов

Для оценки адекватности разработанных моделей и методов проводились вычислительные эксперименты с ТД электронного архива ФНПЦ АО «НПО «Марс» (табл. 4.1).

Таблица 4.1. План экспериментов

1.	Формирование концептуальной сети проектов
2.	Создание и наполнение профилей пользователей
3.	Выполнение поиска документов с применением профилей пользователя
4.	Выполнение поиска документов без применения пользовательских профилей
5.	Выполнение поиска документов с экспертной онтологией
6.	Построение концептуальной сети рабочего проекта ФНПЦ АО «НПО «Марс» и применение ее в процессе информационной поддержки пользователей

Рассмотрим этапы вычислительных экспериментов более подробно:

1. На данном этапе выполнялось построение концептуальной сети проектов с использованием концептов предметной онтологии электронного архива и wiki-ресурса.

2. На данном этапе создавались профили пользователей системы информационной поддержки электронного архива. Каждый профиль соответствовал конкретному этапу разработки АС.

3. С применением профилей, созданных на втором этапе, производились эксперименты поискового характера. Сформирован набор пользовательских запросов, состоящих из терминов, относящиеся к предметной области разработки АС. Сформированные запросы были представлены в двух видах:

- Первый тип относился к запросам, которые семантически явно или неявно определяют информационную потребность пользователя в конкретной предметной области;
- Второй тип запросов определяется количеством терминов в запросе: короткие запросы, состоящие из одного или двух терминов; сложносоставные запросы, которые содержат три и более термина.

4. На данном этапе производились аналогичные эксперименты, как и на третьем этапе, но их отличительной особенностью являлось то, что не использовались пользовательские профили в процессе формирования контекстно-ориентированных проектных запросов к ЭА.

5. Данный этап предполагал выполнение вычислительных экспериментов с концептуальной сетью проектов, разработанной экспертом предметной области.

6. Данный этап представляет собой выполнение вычислительных экспериментов с 1-го по 4-й этап с конкретным рабочим проектом, выполняемым коллективом работников ФНПЦ АО «НПО «Марс». Данный этап включал построение концептуальной сети рабочего проекта с использованием извлеченных концептов из технического задания проекта, и

расширенный с помощью механизма извлечение концептов из wiki-ресурса. Далее выполнялся поиск ТД с проектными запросами различных видов, определенных на третьем этапе экспериментов и с применением профилей проектировщиков, сформированных на втором этапе вычислительных экспериментов.

Оценка качества разработанных моделей и алгоритмов определялась рядом характеристик, которые будут рассмотрены ниже и производилось сравнение с аналогичными результатами, как поисковых систем, так и с системами документооборота, которые имели механизмы информационной поддержки пользователей, а именно встроенные механизмы информационного поиска. В данной работе в качестве аналогов выступали следующие программные продукты:

1) Яндекс. Персональный поиск (ЯПП) – позволяет осуществлять быстрый поиск по файлам и письмам на персональном компьютере и включает код, применяемый в поисковой системе www.yandex.ru (<http://www.softportal.com/software-5261-personalnij-poisk-ndeksa.html>).

2) 1С:Документооборот 8 (1С) – программный продукт российской фирмы «1С» на технологической платформе «1С:Предприятие 8», предназначенный для автоматизации документооборота (<http://v8.1c.ru/doc8/>).

3) Архивариус 3000 (A300) – программа поиска документов и почтовых сообщений на персональном компьютере, в локальной сети и на съёмных дисках. (<http://www.likasoft.com/ru/document-search/>).

4) Copernic Desktop Search (CDS) – программный продукт канадской фирмы Copernic Inc, предоставляющие возможности поиска документов, файлов на персональном компьютере (<http://www.copernic.com/en/products/desktop-search/>).

Далее в работе используются сокращенные наименования рассмотренных выше программных продуктов.

4.1. Формирование концептуальной сети проектов на основе wiki-ресурсов

Процесс формирования концептуальной сети проекта предполагает использование концептов из онтологий предметной области. Данные концепты поступают на вход подсистемы автоматического формирования концептуальной сети. На следующем этапе подсистема извлекает концепты и соответствующие терминологические окружения из страниц wiki-ресурсов. В процессе выполнения экспериментов использовалось большое количество различных понятий из онтологии предметной области проектирования АС, в том числе: "САПР", "базы данных", "UML", "проектирование", "изделие", "техническая документация", "языки программирования", "онтология", "база знаний", "RUP" и другие. Полученная из wiki-ресурса концептуальная сеть содержит 205 концептов и 22 000 уникальных термина, составляющих терминологические окружения понятий на уровне проекта. На рис. 4.1. представлен фрагмент концептуальной сети, извлеченный из wiki-ресурса.

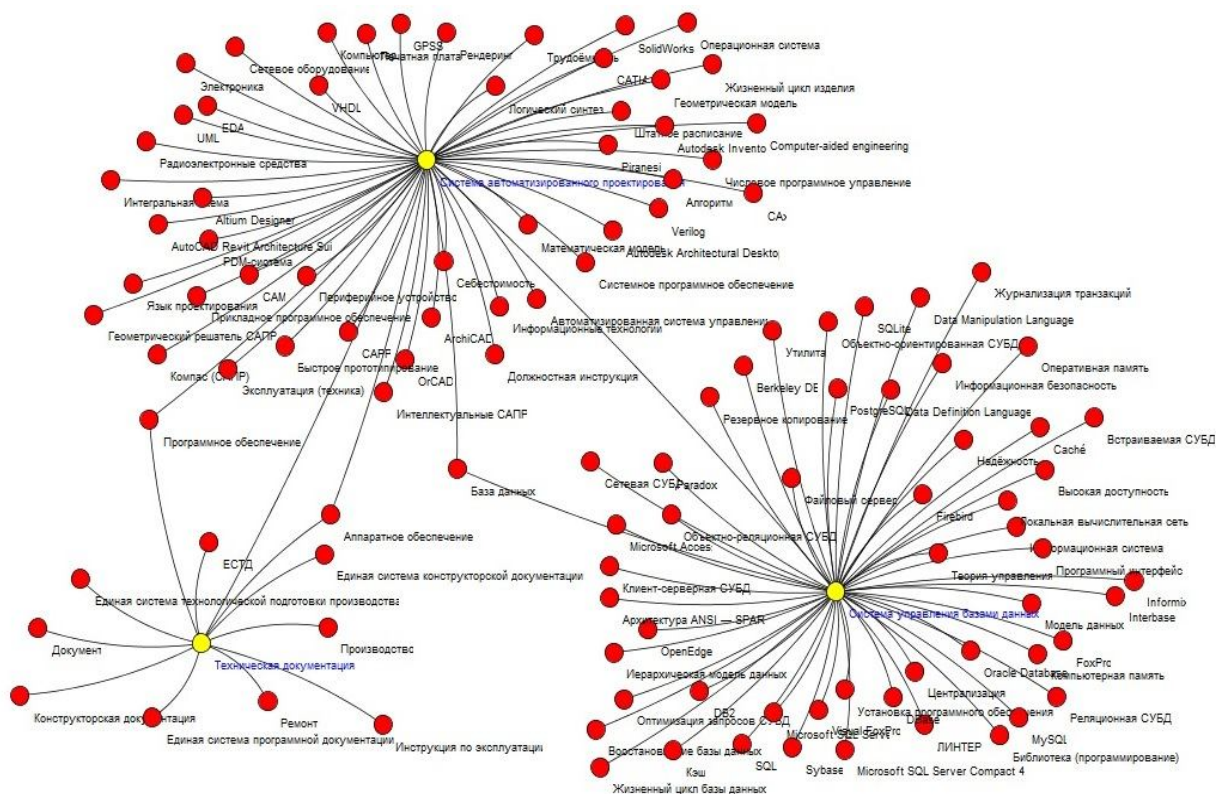


Рис. 4.1. Фрагмент концептуальной сети

На рис. 4.2. представлен концепт "Техническая документация" с соответствующим терминологическим окружением.

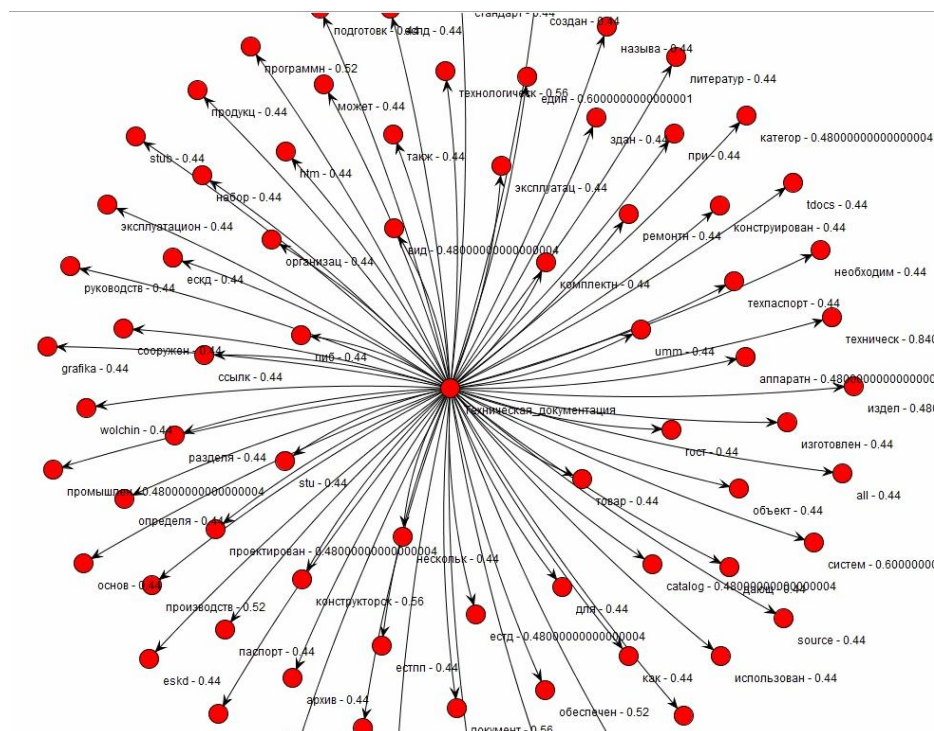


Рис. 4.2. Структура концепта "Техническая документация" с терминологическим окружением

4.2. Формирование пользовательских профилей системы информационной поддержки

В ходе вычислительных экспериментов были созданы несколько пользовательских профилей: "Проектировщик", "Программист" и "Инженер". Каждый профиль отражает информационную потребность определенной категории пользователей, которые используют архив ТД на разных стадиях процесса проектирования и разработки АС.

На рис. 4.3. представлен фрагмент профиля "Инженер", который содержит в себе концепты онтологии как положительно оцененных ТД, так и документы, не соответствующие информационной потребности пользователя.

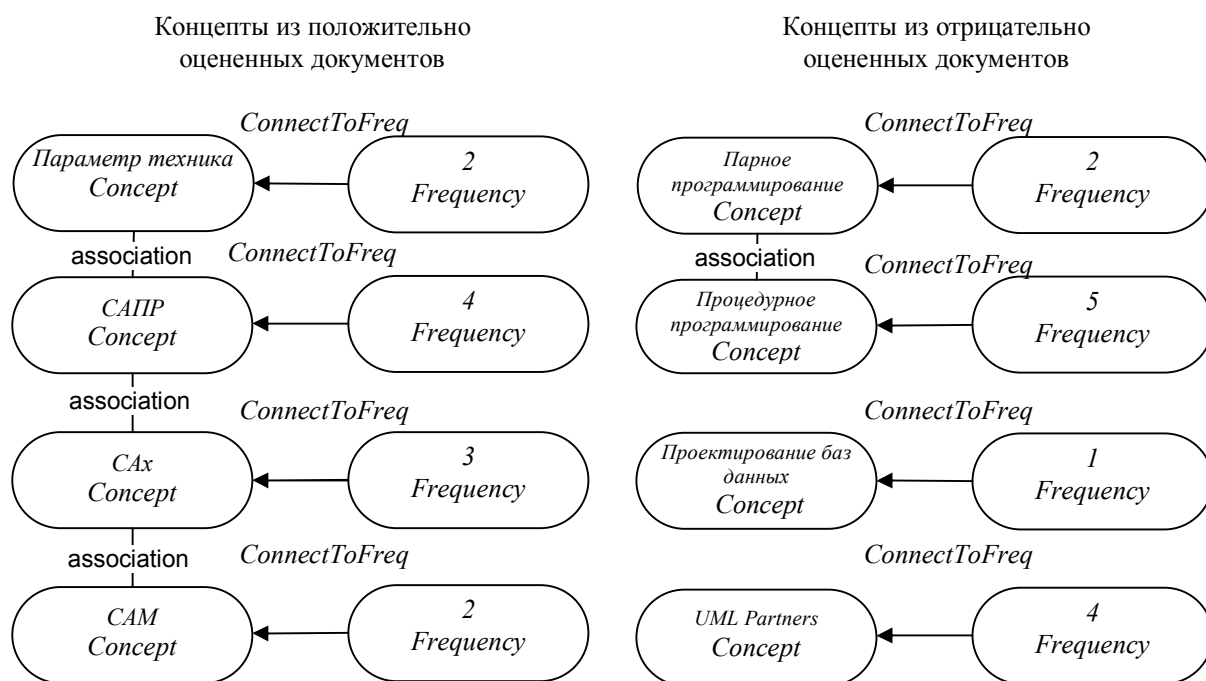


Рис. 4.3. Фрагмент пользовательского профиля "Инженер"

В данном примере представлены концепты с частотой встречаемости в классе положительно и отрицательно оцененных документов. При этом концепты из класса положительно оцененных документов относятся к одной предметной области и связаны ассоциативными отношениями между собой. Это говорит о том, что профиль содержит элементы конкретной предметной области, которая интересует пользователей, взаимодействующих с ЭА. Кроме того, профиль содержит категорию концептов, которые не соответствуют информационной потребности пользователя и относятся к иным подобластям предметной области проектирования АС.

В следующих таблицах представлены фрагменты сформированных профилей подсистемы информационной поддержки.

Таблица 4.2. Фрагмент профиля "Программист"

Класс положительно оцененных документов		Класс отрицательно оцененных документов	
Динамическая типизация	1	Rational Unified Process	2
Синтаксис	1	Схема базы данных	4

программирование			
Динамический язык программирования	2	Техническая документация	3
Высокоуровневый язык программирования	2	Диаграмма классов	3
Интерпретируемый язык программирования	4	Диаграмма развёртывания	2
Парное программирование	3	Жизненный цикл изделия	1
Objective C	1	Система управления базами данных	2

Таблица 4.3. Фрагмент профиля "Инженер"

Класс положительно оцененных документов		Класс отрицательно оцененных документов	
Параметр техника	1	Парное программирование	2
Жизненный цикл изделия	1	Процедурное программирование	5
САПР	4	Проектирование баз данных	2
CAx	3	UML Partners	4
CAM	2	Язык проектирования	4
Техническая документация	3	Диаграмма пакетов	3
ArchiCAD	2	Диаграмма прецедентов	5

Таблица 4.4. Фрагмент профиля "Проектировщик"

Класс положительно оцененных документов		Класс отрицательно оцененных документов	
Диаграмма	2	Парное	2

последовательности		программирование	
Диаграмма_прецедентов	3	Процедурное программирование	5
Диаграмма последовательности	1	Java	4
<i>UML</i>	2	САПР	4
Диаграмма коммуникации	1	САМ	3
Rational Unified Process	2	Базы данных	1
Жизненный цикл программного обеспечения	2	C++	5

4.3. Математическая модель оценки качества формирования поисковых запросов

Для оценки качества предметно-ориентированных поисковых запросов к ЭА ТД использовались следующие характеристики [29]:

Полнота (Р) – доля релевантных документов в выборке, по отношению ко всем релевантным документам коллекции:

$$P = \frac{\text{кол-во релевантных найденных документов}}{\text{кол-во релевантных документов}}.$$

Точность (Т) – доля релевантных документов выборки, по отношению ко всем документам выборки:

$$T = \frac{\text{кол-во релевантных найденных документов}}{\text{кол-во найденных документов}}.$$

Для удовлетворения баланса между двумя этими параметрами использовался параметр F-мера (F measure) [29]. Данный параметр представляет собой среднее гармоническое взвешенное:

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R},$$

где $\beta^2 = \frac{1-\alpha}{\alpha}$, $\alpha \in [0,1]$, т.е. $\beta^2 \in [0,\infty]$. По умолчанию сбалансированная F-мера присваивает точности и полноте одинаковые веса, т.е. $\alpha = 1/2$, или $\beta = 1$. Если $\beta < 1$ предпочтение отдают точности поиска, при $\beta > 1$ – полноте поиска. При $\beta = 1$ формула принимает вид:

$$F_{\beta} = \frac{2PR}{P + R}.$$

В данной работе при анализе результатов выполняемых экспериментов использовалась сбалансированная F-мера.

4.4. Сравнительный анализ результатов вычислительных экспериментов на множестве документов электронного архива ФНПЦ АО «НПО «Марс»

В ходе исследования были проведены вычислительные эксперименты, которые заключались в формировании контекстно-ориентированных проектных запросов и использовании их в системе информационной поддержки. Производилось вычисление качественных характеристик и сравнение с аналогичными характеристиками существующих систем информационного поиска.

Вычислительные эксперименты с использованием онтологии предметной области и профилей

Первоначально рассматривался вид запроса, в котором явно/не явно определялась предметная область. Сравнительные результаты по каждому профилю представлены в виде гистограмм. В качестве итоговых величин точности, полноты и F-меры использовались значения, которые наиболее часто встречались в ходе экспериментов.

В результате экспериментов с применением профиля "Программист" были получены следующие результаты для запросов, которые семантически явно определяют предметную область (рис. 4.4) и для запросов, которые не

явно идентифицируют предметную область (рис. 4.5).

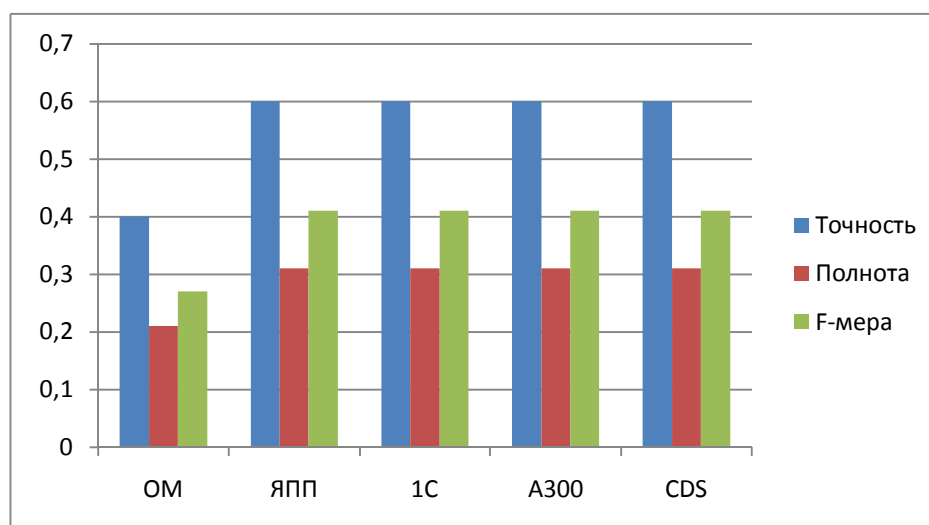


Рис.4.4. Сравнение результатов экспериментов для профиля программиста с запросами, семантически явно определяющих предметную область.

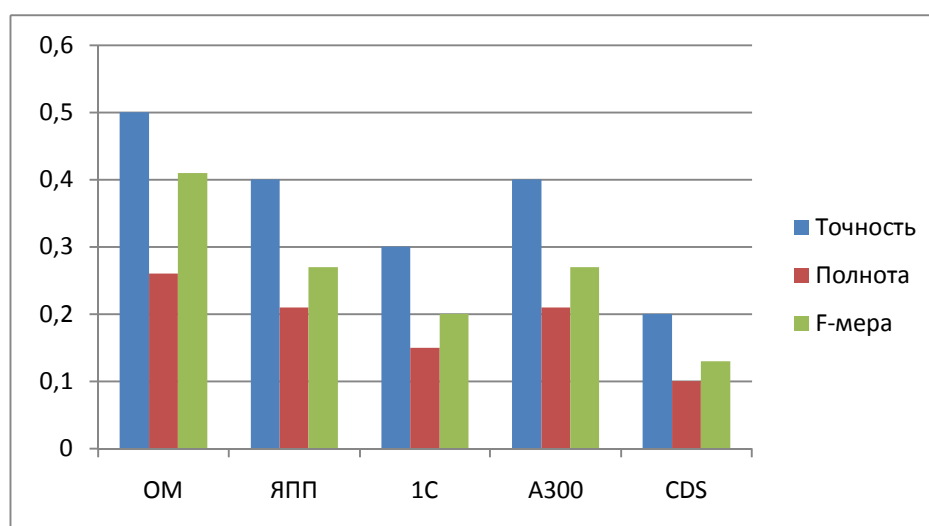


Рис.4.5. Сравнение результатов экспериментов для профиля программиста с запросами, семантически неявно определяющих предметную область.

Следующий набор запросов определялся количеством терминов. На рис. 4.6 и рис. 4.7 представлены гистограммы, в которых отражена оценка влияния размера запроса на качество поиска ТД.

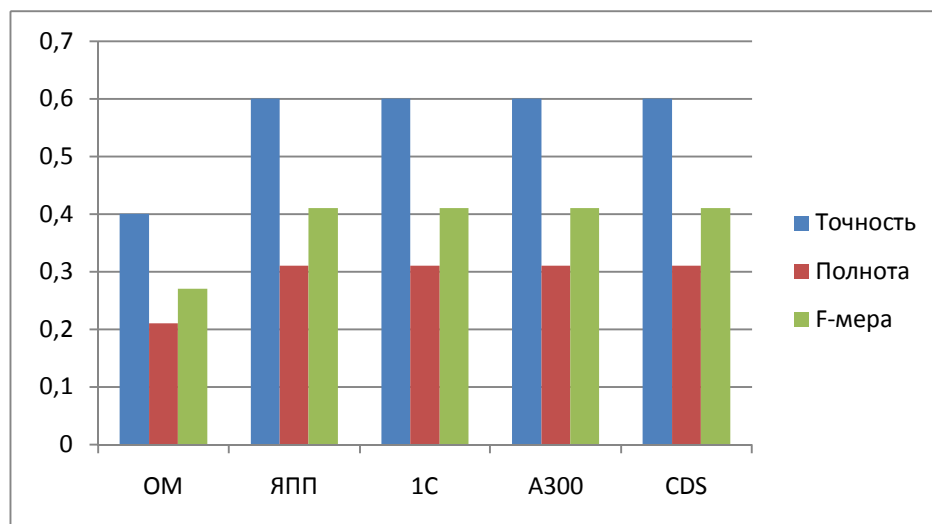


Рис. 4.6. Сравнение результатов экспериментов для профиля программиста с короткими запросами

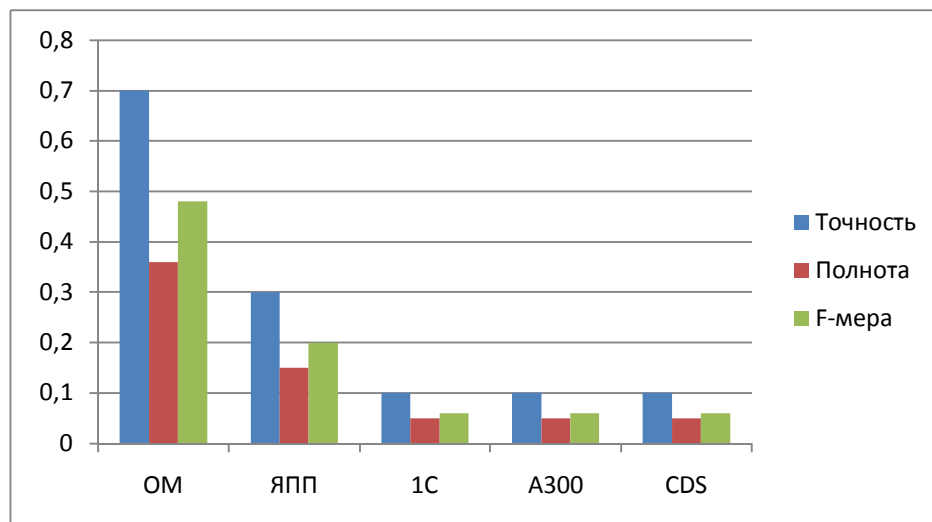


Рис. 4.7. Сравнение результатов экспериментов для профиля программиста с длинными запросами

В результате экспериментов с применением профиля "Инженер" получены следующие результаты для запросов, которые семантически явно определяют предметную область (рис. 4.8) и для запросов, которые неявно определяют предметную область (рис. 4.9).

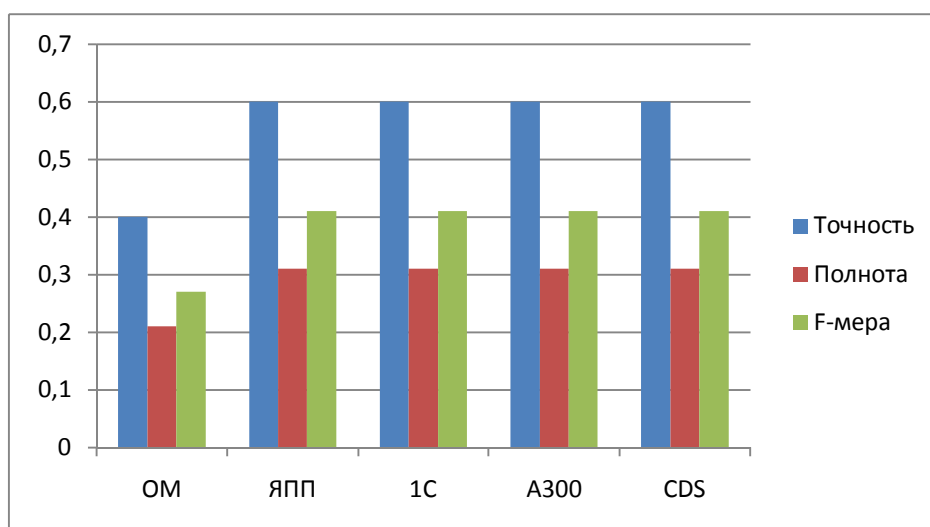


Рис. 4.8. Сравнение результатов экспериментов для профиля инженера с запросами, семантически явно определяющих предметную область

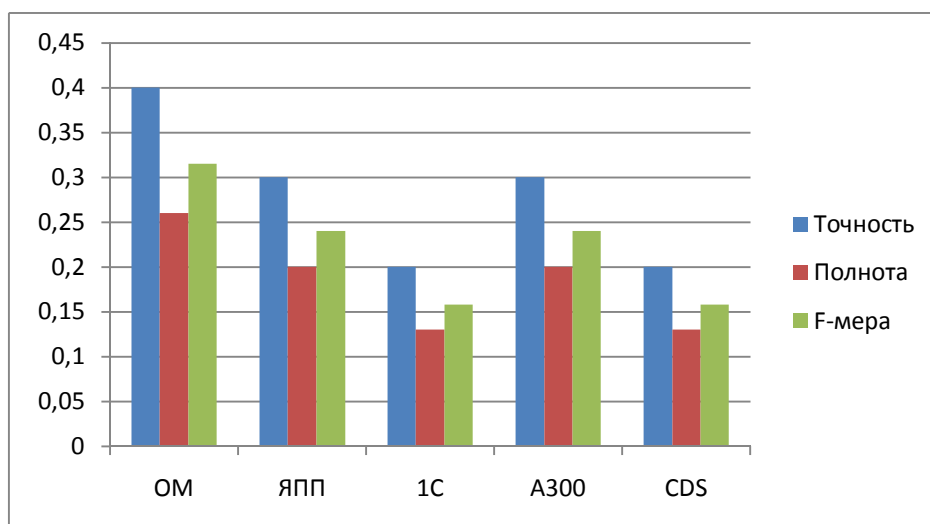


Рис. 4.9. Сравнение результатов экспериментов для профиля инженера с запросами, семантически неявно определяющих предметную область

Следующий набор запросов определялся количеством терминов. На рис. 4.10 и рис 4.11 представлены гистограммы, на которых отражается влияние размера запроса на поиск ТД для профиля инженера.

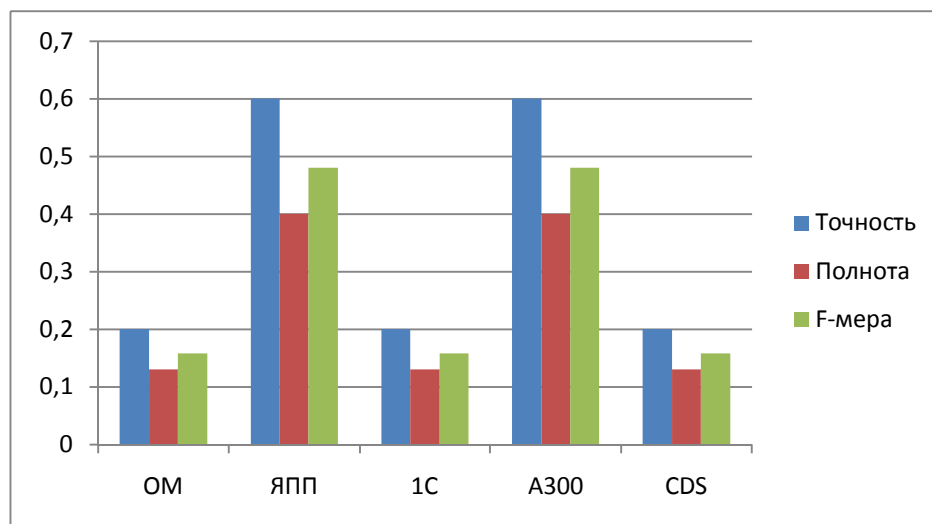


Рис. 4.10. Сравнение результатов экспериментов для профиля инженера с короткими запросами

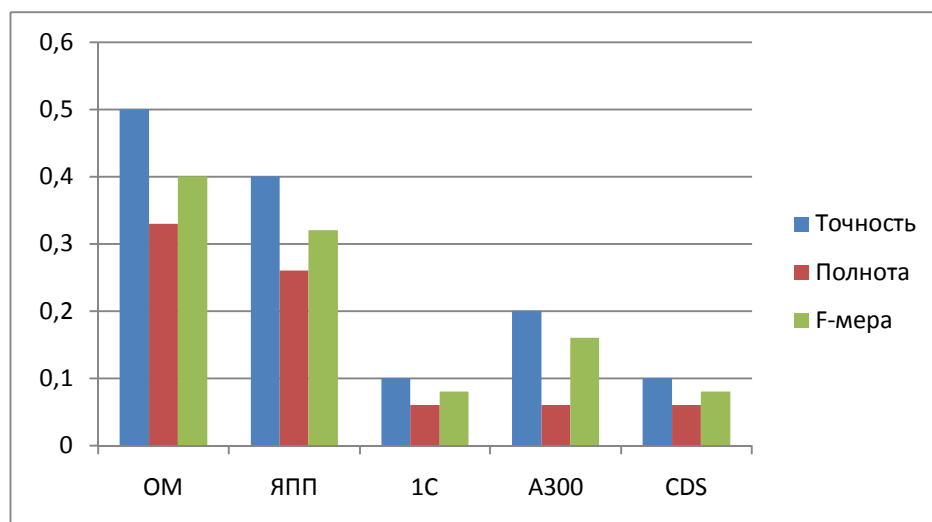


Рис. 4.11. Сравнение результатов экспериментов для профиля инженера с длинными запросами

В результате экспериментов с применением профиля "Проектировщик" получены следующие результаты для запросов, которые семантически явно определяют предметную область (рис. 4.12) и для запросов, которые неявно идентифицируют предметную область (рис. 4.13).

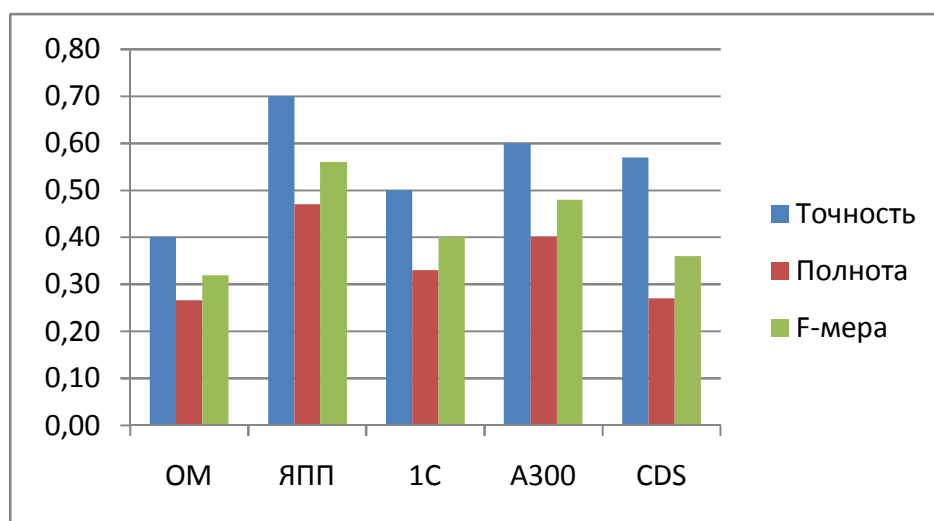


Рис. 4.12. Сравнение результатов экспериментов для профиля проектировщика с запросами, семантически явно определяющих предметную область

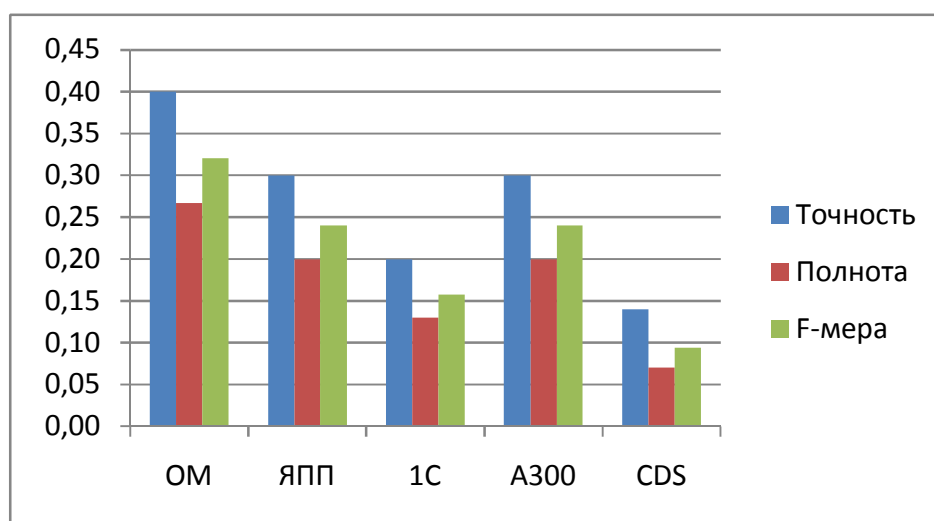


Рис. 4.13. Сравнение результатов экспериментов для профиля проектировщика с запросами, семантически неявно определяющих предметную область

Следующий набор запросов определялся количеством терминов. На рис. 4.14 и рис. 4.15 представлены гистограммы, которые иллюстрируют влияние размера запроса на качество поиска ТД для профиля проектировщика.

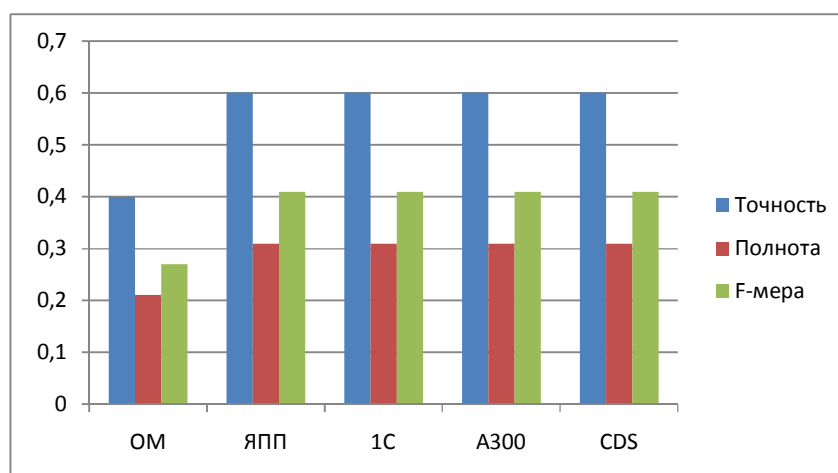


Рис. 4.14. Сравнение результатов экспериментов для профиля проектировщика с короткими запросами

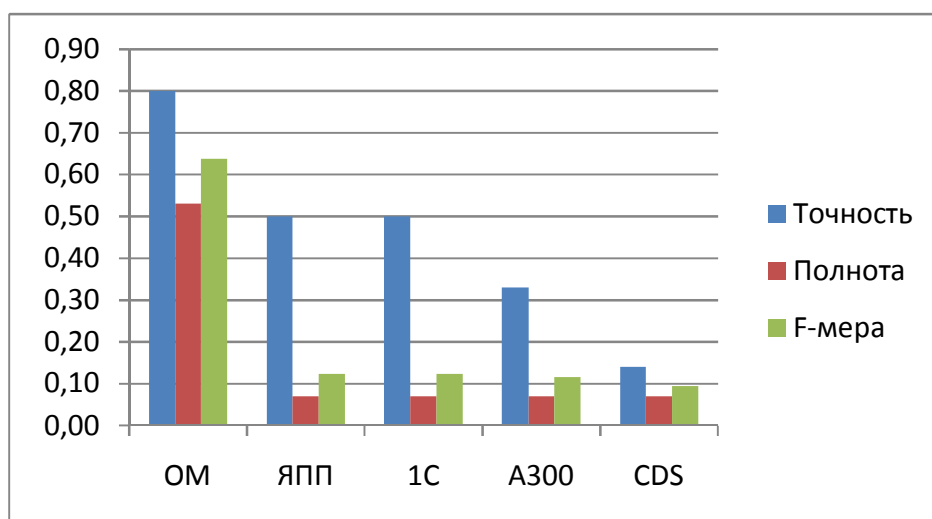


Рис. 4.15. Сравнение результатов экспериментов для профиля проектировщика с длинными запросами

На рис. 4.16 представлен сравнительный анализ двух способов формирования контекстно-ориентированного поиска: с использованием профилей пользователей и с онтологическим поиском, в котором не используется информация о содержимом профилей пользователей.

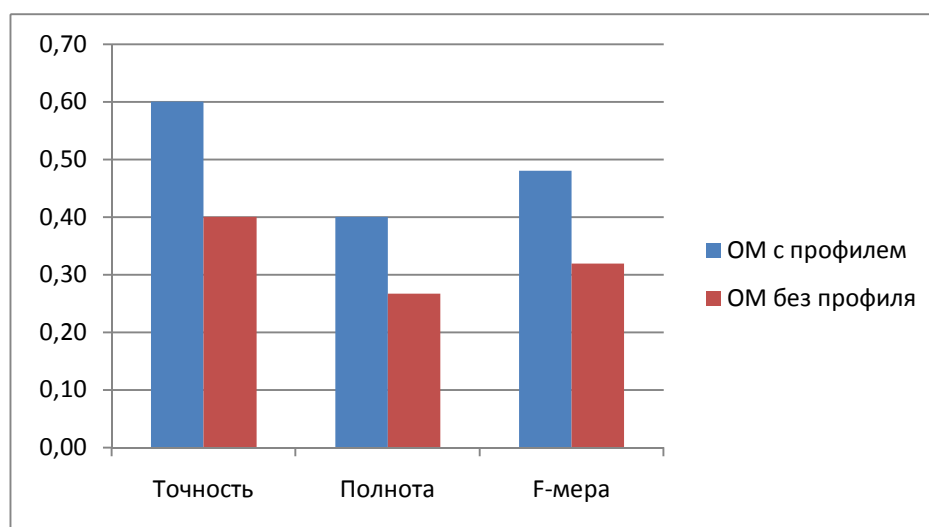


Рис. 4.16. Сравнение результатов экспериментов онтологических моделей поиска

В ходе проведенных экспериментов было определено, что онтологическая модель, использующая семантические профили пользователей, показывает более качественный результат, чем поиск, который не учитывает информационную потребность конкретной группы специалистов. Применение профилей позволяет достигнуть показателей качества, представленные в таблице 4.5.

Таблица 4.5.

Характеристики	Оценка улучшения поиска в интервальной форме
Точность	от 30% до 60%
Полнота	от 30% до 50%
F-мера	от 30% до 60%

Следующим этапом экспериментов стало сравнение качества результатов контекстно-ориентированного поиска ТД в ЭА, которые используют различные онтологии, отличающиеся способом формирования. Первая онтология содержит концептуальную сеть, концепты которой автоматизированным способом были извлечены из электронной библиотеки. Вторая онтология содержит концептуальную сеть, составленную экспертом предметной области. На рис. 4.17. представлен результат сравнения данного

этапа вычислительных экспериментов.

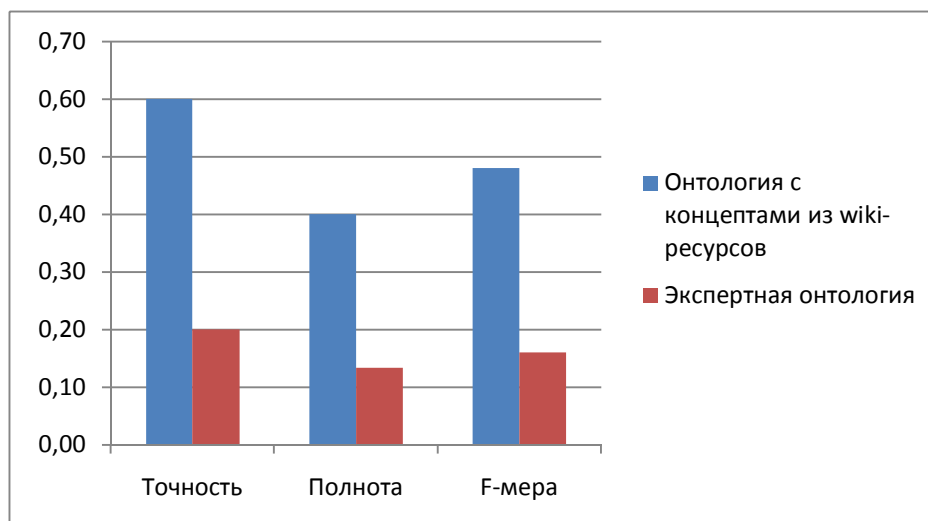


Рис. 4.17. Сравнение результатов экспериментов онтологических моделей поиска, использующие различные онтологии

Результаты вычислительных экспериментов показывают, что построение концептуальной сети автоматизированным способом и применение ее в процессах информационной поддержки позволяет улучшить качество поиска, по сравнению с экспертной онтологией. В таблице 4.6 представлены сравнительные характеристики качества поиска.

Таблица 4.6.

Характеристики	Оценка улучшения поиска в интервальной форме
Точность	от 30% до 70%
Полнота	от 40% до 70%
F-мера	от 35% до 70%

Отдельный этап вычислительных экспериментов выполнялся с рабочим проектом, реализуемым коллективом работников ФНПЦ АО «НПО «Марс». Процесс проектирования сопровождается многочисленными проектными запросами к ЭА ТД и применением опыта предыдущих разработок.

Использование интеллектуальной информационной поддержки в ЭА не нарушает общепринятых этапов проектирования, но способствует

повышению скорости реализации проекта. Интеллектуальная компонента информационной поддержки включает в себя формирование онтологии, разработку индивидуальных профилей проектировщиков и интеграцию разработанных онтологических ресурсов в систему электронного архива проектной организации. Применение данной компоненты на первоначальном этапе минимально, особенно активно интеграция происходит на стадии анализа технического задания (рис. 4.18).

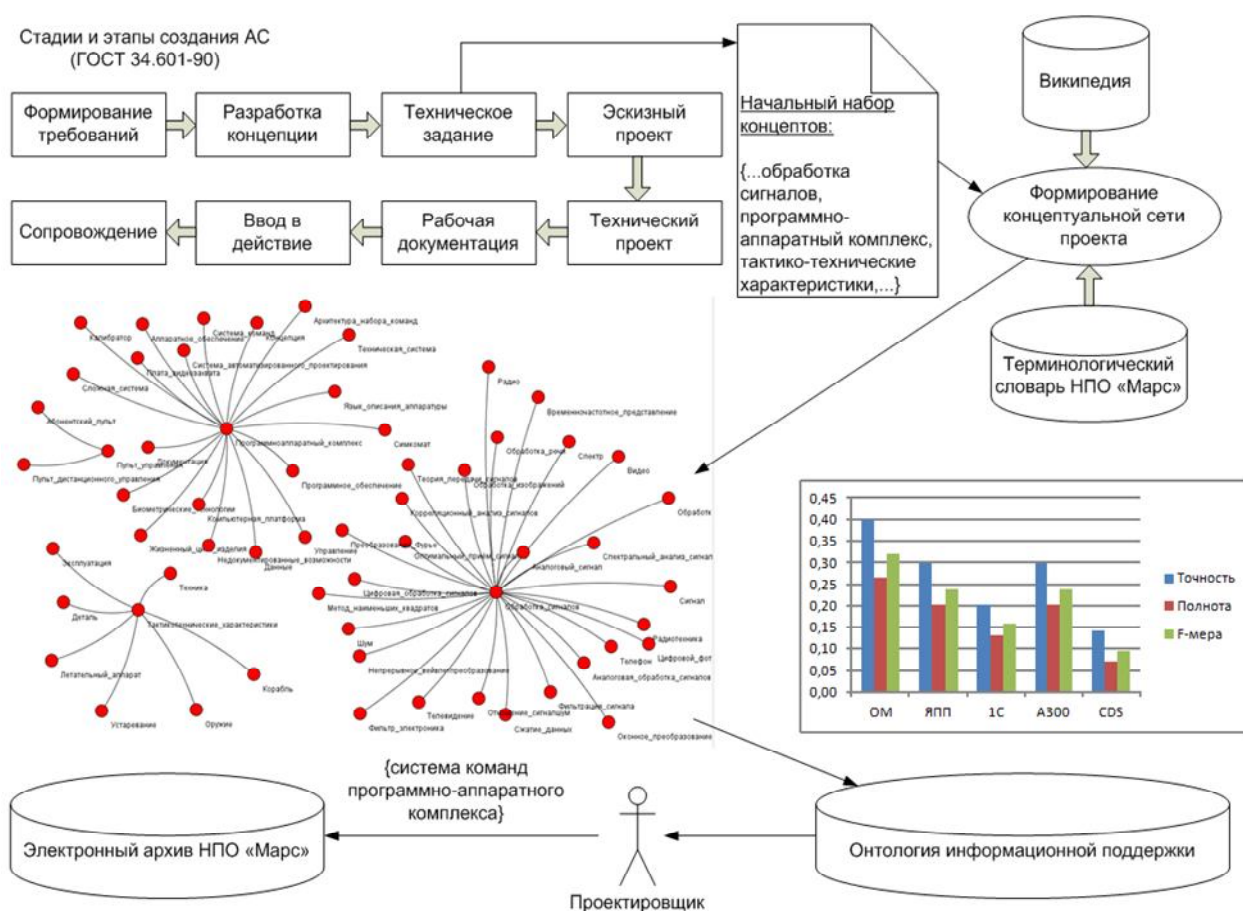


Рис 4.18. Интеграция онтологической модели информационной поддержки в ЭА проектных документов

На рисунке 4.18 представлены этапы проектирования с применением интеллектуальной компоненты информационной поддержки. Как видно из рисунка активная фаза использования возникает в процессе анализа технического задания и сопровождается извлечением понятий, которые используются как начальный набор концептов для построения

коллективом.

Для оценки качества информационной поддержки проводились аналогичные опыты, которые рассматривались выше, а именно сравнение онтологической модели поиска с существующими системами информационного поиска. На рис. 4.20 и 4.21 представлены гистограммы, которые показывают, что с запросами, которые слабо выражают семантику предметной области и состоящие из большого количества терминов, онтологическая модель показала более качественные результаты.

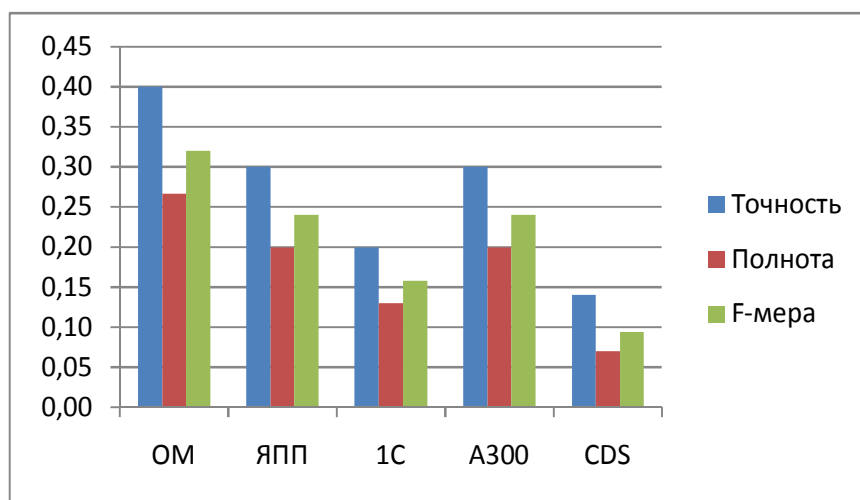


Рис.4.20. Сравнение результатов экспериментов с запросами, слабо выражающими семантику предметной области.

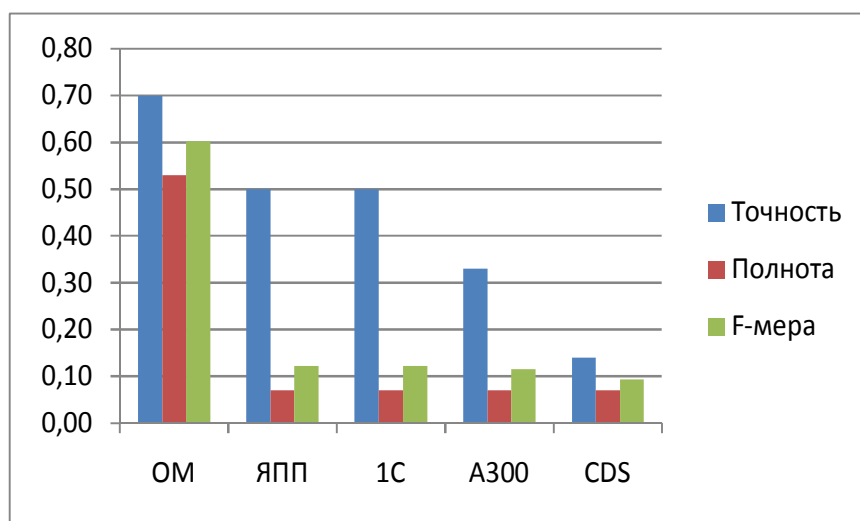


Рис. 4.21. Сравнение результатов экспериментов с длинными запросами

В таблице 4.7 представлены сравнительные характеристики качества

поиска.

Таблица 4.7.

Характеристики	Оценка улучшения поиска в интервальном значений
Точность	от 20% до 60%
Полнота	от 30% до 65%
F-мера	от 25% до 60%

Полученные результаты эксперимента с рабочим проектом показали, что онтологическая модель поиска ТД, показывает более качественный результат, чем традиционные модели поиска.

Время поиска с применением онтологической модели сравнивалось с существующей подсистемой информационного поиска электронного архива ТД НПО «Марс». В данном эксперименте сравнивались временные показатели, затрачиваемые на поиск технических документов в рамках эксперимента с рабочим проектом. Сравнительные характеристики представлены в таблице 4.8.

Таблица 4.8.

Виды поиска	Существующая подсистема поиска ЭА	Онтологическая модель поиска
Запросы, семантически явно определяющие предметную область	< 1 мин	< 3 мин
Запросы, слабо выражающие семантику предметной области	до 15 мин.	< 5 мин

На основании результатов выведем следующее заключение:

1. Онтологическая модель подсистемы информационной поддержки позволяет получать более качественные результаты, по сравнению с

другими аналогичными системами, если пользователь использует запросы, которые семантически слабо выражают конкретную предметную область.

2. Онтологическая модель показывает худший результат в случае, если используются короткие запросы по причине того, что чем короче запрос, тем слабее идентифицируются концепты, в которых термины формируют терминологическое окружение.

3. Онтологическая модель показывает качественный результат, если пользователь использует запросы, содержащие более трех терминов. Чем больше терминов используется в запросе, тем лучше идентифицируются концепты из предметной онтологии.

4. Wiki-ресурсы описывают различные предметные области с разной степенью детализации. Так, пользователи профиля "Программист", получали более качественные результаты поиска, чем пользователи профиля "Инженер" и "Проектировщик" по причине более детального описания соответствующей предметной области в электронной библиотеке.

5. Применение профилей пользователей электронного архива улучшает качество результатов проектных запросов к ЭА ТД с применением онтологической модели по сравнению с поиском документов без учета информационных потребностей пользователей.

6. Вычислительные эксперименты показали, что концептуальная сеть проекта, сформированная автоматизированным способом посредством извлечения знаний из wiki-ресурсов, является предпочтительной в процессе информационной поддержки дополнительно к онтологии, разработанной экспертом предметной области.

4.5 Выводы по главе

1. Онтологическая модель формирования контекстно-ориентированных запросов к ЭА, учитывающая информационную потребность пользователя, является эффективным средством при решении задачи информационной поддержки пользователей электронного архива ТД проектной организации. Точность выполнения проектных запросов, которые семантически нечетко выражают предметную область, улучшилась примерно на 30% по сравнению с запросами аналогичных систем.
2. Автоматизированное построение концептуальной сети с использованием профессиональных wiki-ресурсов позволяет уменьшить время создания онтологии и улучшить качество проектных поисковых запросов в среднем до 50%.
3. Разработанная программная система может применяться в проектных организациях для более эффективной организации взаимодействия пользователей с электронным архивом технической документации, позволяя сократить среднее время поиска технического документа в одном сеансе работы с электронным архивом примерно на 50% (с 14 минут до 6-7 минут).

Заключение

В ходе диссертационного исследования получены следующие результаты:

1. Предложена новая структурно-функциональная модель онтологии информационной поддержки проектирования АС, отличающаяся многоуровневой структурой, учитывающая этапы жизненного цикла проектируемого изделия и текущий контекст проекта.

2. Предложена онтологическая модель профиля проектировщика, учитывающая опыт взаимодействия субъекта проектирования с электронным архивом на концептуальном уровне.

3. Предложен алгоритм формирования контекстно-ориентированных проектных запросов к электронному архиву технических документов с целью повышения качества информационной поддержки проектировщика.

4. Разработан комплекс программ как подсистема информационной поддержки пользователя электронного архива технической документации, которая в полной мере реализует все описанные теоретические положения и позволяет сократить время поиска технических документов примерно на 40-50% с увеличением точности поиска на 30%.

5. Проведены вычислительные эксперименты, основными результатами которых являются:

- Онтологическая модель поиска, учитывающая информационную потребность проектировщика, является эффективным средством при решении задачи информационной поддержки проектирования АС. Онтологическая модель по сравнению с традиционными моделями формирования контекстно-ориентированных запросов показала более качественный результат с запросами, которые семантически нечетко

выражают предметную область и с запросами, которые содержат большое количество терминов.

- Автоматизированное построение концептуальной сети с использованием внешних профессиональных wiki-ресурсов позволяет уменьшить время создания онтологии и улучшить качество поиска технических документов в ЭА проектной организации на 30%.

Библиографический список

1. Агеев М.С. Методы автоматической рубрикации текстов, основанные на машинном обучении и знаниях экспертов.: автореф. дис. на соиск. учен. степ. канд. физико-математических наук (05.13.11); МГУ – М, 2004 – 16 с.
2. Андреев А.М., Березкин Д.В., Рымарь В.С., Симаков К.В. Использование технологии Semantic Web в системе поиска несоответствий в текстах документов. – URL: http://www.inteltec.ru/publish/articles/textan/rimar_RCDL_2006.shtml. (Дата обращения: 14.06.2011)
3. Баргесян А.А. Анализ данных и процессов: учеб. пособие. – СПб.: БХВ-Петербург, 2009.
4. Бениаминов. Некоторые проблемы широкого внедрения онтологий в IT и направления их решений. URL: <http://www.beniaminov.rsuh.ru/BeniaminovOntoNew.pdf>. (Дата обращения: 25.09.2012)
5. Бонгард М.М. Проблема узнавания. – М.: Наука, 1967.
6. Вайнцвайг М.Н. Алгоритм обучения распознаванию образов "Кора" // Алгоритмы обучения распознаванию образов / Под ред. В.Н. Вапника. – М.: Сов. радио, 1973.
7. Вятчин Д.А. Нечеткие методы автоматической классификации. – Минск: Технопринт, 2004.
8. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000.
9. Гаврилова Т.А., Червинская К.Р. Извлечение и структурирование знаний для экспертных систем. – М.: Радио и связь, 1992.
10. Гнеденко Б.В. Курс теории вероятностей. – М.: Наука, 1988.

11. Городецкий В.И., Тушканова О.Н. Онтологии и персонификация профиля пользователя в рекомендующих системах третьего поколения. // Онтология проектирования №3 (13), 2014. – С. 7–32.
12. Губин М.В. Модели и методы представления текстового документа в системах информационного поиска. – URL: <http://maxgubin.com/articles/thesis.pdf>. (Дата обращения: 12.06.2011)
13. Добров Б.В., Лукашевич Н. В., Сеницин М. Н., Шапкин В. Н. Разработка лингвистической онтологии по естественным наукам для решения задач информационного поиска. // Труды 7-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции». – Ярославль, 2005.
14. Добров Б.В., Лукашевич Н.В. Автоматическая интеллектуальная обработка текстов на основе тезаурусно-организованных знаний // Труды шестой национальной конференции по ИИ КИИ-98: Труды конференции. 1998. Т. 2. – С. 486 – 491.
15. Добров Б.В., Лукашевич Н.В. Автоматическая рубрикация полнотекстовых документов по классификаторам сложной структуры // Восьмая национальная конференция по искусственному интеллекту – Коломна, 2002.
16. Добров Б.В., Лукашевич Н.В. Тезаурус и автоматическое концептуальное индексирование в университетской информационной системе РОССИЯ. // Третья Всероссийская конференция по Электронным Библиотекам «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – Петрозаводск, 2001. – С.78 – 82.
17. Дюран Б., Оделл П. Кластерный анализ. – М.: Статистика, 1977.
18. Епрев А.С. Исследование влияния разрешения лексической многозначности с помощью контекстных векторов на эффективность категоризации текстовых документов.: автореф. дис. на соиск. учен. степ. канд. физико-математических наук (05.13.11); Институт математики им. С.Л. Соболева Сибирского отделения РАН – Новосибирск, 2011 – 19 с.

- 19.Ермаков А.Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста. // Материалы международной конференции «Диалог 2008». – Москва, 2008. – С. 154–159.
- 20.Загоруйко Н.Г. и др. Система «Ontogrid» для построения онтологий // Компьютерная лингвистика и интеллектуальные технологии. Тр. междунар. конференции Диалог'2005. – М.: 2005. – С. 146–152.
- 21.Загорулько Ю.А. Автоматизация сбора онтологической информации об интернет-ресурсах для портала научных знаний // Известия Томского политехнического университета №5. – 2008.
- 22.Загорулько Ю.А., Кононенко И.С., Сидорова Е.А. Семантический подход к анализу документов на основе онтологии предметной области. – URL: <http://www.dialog-21.ru/digests/dialog2006/materials/html/SidorovaE.html>. (Дата обращения: 12.09.2012)
- 23.Захарова И. В. Математическая модель семантического поиска с использованием онтологического подхода.: автореф. дис. на соиск. учен. степ. канд. физико-математических наук (05.13.18); ЧГУ – Челябинск, 2010 – 21 с.
- 24.Касумов В. А., Касумова К. П. Методы определения тематики и содержания индивидуальных и коллективных информационных потребностей // Научно-техническая информация Серия 2. – М.: ВИНТИ РАН, 2012. – С. 1–9.
- 25.Коваль С.А. Автоматическая переработка текста на базе объектнопредикатной системы // Структурная и прикладная лингвистика. Вып. 5. – СПб.: 1998. – С. 199–207.
- 26.Ландэ Д.В., Снарский А.А., Безсудов И.В. Интернетика: навигация в сложных сетях: модели и алгоритмы. – М.: Книжный дом «ЛИБРОКОМ», 2009.
- 27.Ландэ Д.В. Добыча знаний. – URL: <http://visti.net/dwl/art/dz/>. (Дата обращения: 08.09.2011)
- 28.Мандель И.Д. Кластерный анализ. – М.: Финансы и статистика. – 1988.

- 29.Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011.
- 30.Мао Ж., Джейн Э. Введение в искусственные нейронные сети № 4. 1997. – URL: <http://www.osp.ru/text/302/179189/> (Дата обращения: 12.01.2013)
- 31.Маркарова Т.С. Конструкции с предикатами, выражающими отношения "Часть-целое" в современном русском языке.: автореф. дис. на соиск. учен. степ.канд.филол. наук (10.02.01); МГУ – М., 1996 – 23 с.
- 32.Найханова Л.В., Технология создания методов автоматического построения онтологий с применением генетического и автоматного программирования. – Улан-Удэ.:БНЦ СО РАН, 2008.
- 33.Найханова Л.В. Основные аспекты построения онтологий верхнего уровня и предметной области. – URL: <http://window.edu.ru/resource/717/36717>. (Дата обращения: 08.02.2013)
- 34.Наместников А.М. Интеллектуальные проектные репозитории. – Ульяновск: УлГТУ, 2009.
- 35.Наместников А.М., Субхангулов Р.А. Онтологический подход к формированию проектных запросов интеллектуального агента // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар. научн.техн. конф. (Минск, 19-21 февраля 2015 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР, 2015. – С. 407–412.
- 36.Наместников А.М., Субхангулов Р.А. Онтологически-ориентированная модель классификаций текстовых документов // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2014): материалы IV Междунар. научн.техн. конф. (Минск, 20-22 февраля 2014 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2014. – С. 385–389.
- 37.Наместников А.М., Субхангулов Р.А. Оптимизация поисковых запросов на основе байесовских классификаторов // Гибридные и синергетические интеллектуальные системы: теория и практика: материалы 2-го

- международного симпозиума / под ред. проф. А.В. Колесникова. – Калининград : Изд-во БФУ им. И. Канта, 2014. – С.246–251.
- 38.Наместников А.М., Субхангулов Р.А. Предметно-ориентированный инструментарий разработки онтологии // Гибридные и синергетические интеллектуальные системы: теория и практика: материалы 1-го международного симпозиума / под ред. проф. А.В. Колесникова. – Калининград: Изд-во БФУ им. И. Канта, 2012. – С. 197 – 204.
- 39.Наместников А.М., Субхангулов Р.А. Нечеткий онтологический поиск на основе интеллектуального агента // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VIII-й Международной научно-практической конференции (Коломна, 18-20 мая 2015 г.). В 2-х томах. Т1. – М. : Физматлит, 2015. – С. 277–286.
- 40.Наместников А.М., Субхангулов Р.А. Разработка инструмента инженерии онтологии в интеллектуальном проектном репозитории // Автоматизация процессов управления №2 (28) – 2012. – С.38 – 43.
- 41.Наместников А.М., Субхангулов Р.А. Формирование информационных запросов к электронному архиву на основе концептуального индекса // Радиотехника №7 – 2014. – С. 126-129.
- 42.Наместников А.М., Субхангулов Р.А., Филиппов А.А. Онтологически-ориентированная система кластеризации и полнотекстового поиска проектных документов // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III Междунар. научн.техн. конф. (Минск, 21-23 февраля 2013г.) / редкол.: В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2013. С. 219–224.
- 43.Наместников А.М., Субхангулов Р.А., Филиппов А.А. Применение нечетких моделей в задачах кластеризации и информационного поиска текстовых проектных документов // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VII-й Международной научно-практической конференции (Коломна, 20-22 мая 2013 г.). В 3-х томах. Т3. – М.: Физматлит, 2013. – С. 1278–1289.

- 44.Наместников А.М., Субхангулов Р.А., Филиппов А.А. Разработка инструментария для интеллектуального анализа технической документации // Известия Самарского научного центра Российской академии наук № 4, Том 13. – 2011. – С. 984–990.
- 45.Наместников А.М., Субхангулов Р.А., Филиппов А.А. Система кластеризации и полнотекстового поиска проектных документов на основе прикладной онтологии // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012 (16-20 октября 2012 г., г. Белгород, Россия): Труды конференции. Т.2.– Белгород: Изд-во БГТУ, 2012. – С. 104–111.
- 46.Наместников А.М., Филиппов А.А. Метод генетической оптимизации онтологических представлений проектных документов в задаче индексирования // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012 (16-20 октября 2012 г., г. Белгород, Россия): Труды конференции. Т.4. – Белгород: Изд-во БГТУ, 2012. – С. 84–91.
- 47.Наместников А.М., Филиппов А.А., Субхангулов Р.А. Онтологический подход к кластеризации и классификации технических документов // Нечеткие системы и мягкие вычисления: VI-я Всероссийская научно-практическая конференция НСМВ-2014 (г. Санкт-Петербург, 27-29 июня, 2014 г.): В 2-х томах. Т.1 – СПб.:Политехника-сервис, 2014. – С. 144–153.
- 48.Нгуен Ба Нгок, Тузовский А.Ф. Модель информационного поиска на основе семантических метаописаний / Управление большими системами. Выпуск 41. М. : ИПУ РАН, 2013. – С.51–92.
- 49.Норенков И. П. Основы автоматизированного проектирования: учеб. для вузов. – 4-е изд., перераб. и доп. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2009.
- 50.Норенков И.П. Интеллектуальные технологии на основе онтологий // Информационные технологии № 1, 2010. – С. 17–23.

- 51.Олдендерфер М. С., Блэшфилд Р. К. Кластерный анализ / Факторный, дискриминантный и кластерный анализ: пер. с англ.; Под. ред. И. С. Енюкова. – М.: Финансы и статистика, 1989.
- 52.Поспелов Д.А. Становление информатики в России. / В кн. «Очерки истории информатики в России». - Редакторы-составители Д. А. Поспелов и Я. И. Фет. - Новосибирск: Научно-издательский центр ИГГМ СО РАН, 1998.
- 53.Росеева О.И., Загорулько Ю.А. Организация эффективного поиска на основе онтологий. // Труды международного семинара Диалог 2001 по компьютерной лингвистике и ее приложениям, Т.2 . – Аксаково: 2001. – С. 333–342.
- 54.Рубашкин В.Ш. Представление и анализ смысла в интеллектуальных информационных системах. – М.: Наука, 1989.
- 55.Соколов А.В. Философия информации: профессионально-мировоззренческое пособие. СПб.: СПбГУКИ, 2010.
- 56.Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В. Онтологии и тезаурусы. Казань: КГУ, 2006.
- 57.Субхангулов Р.А. Онтологический поиск технических документов на основе модели интеллектуального агента // Автоматизация процессов управления №4 (38) – 2014. – С. 85–91
- 58.Субхангулов Р.А. Онтологически-ориентированная модель поиска текстовых документов // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. Н. Н. Войта. – Ульяновск: УлГТУ, 2013. – С. 357–363.
- 59.Субхангулов Р.А. Онтологически-ориентированный метод поиска проектных документов // Автоматизация процессов управления №4 (30) – 2012 – С.83 – 89.
- 60.Субхангулов Р.А. Разработка инструментария онтологии в интеллектуальном проектном репозитории // Автоматизация процессов управления: сборник докладов Молодежной научно-технической

- конференции, Ульяновск, 13- 14 декабря 2011 г. / под общ. ред. А.А. Емельянова. – Ульяновск : ФНПЦ ОАО «НПО «Марс», 2011. – С. 93–100.
- 61.Субхангулов Р.А. Создание редактора онтологии для анализа проектных документов // Информатика и вычислительная техника: сборник научных трудов 4-й Всероссийской научно-технической конференции аспирантов, студентов и молодых ученых ИВТ-2012. В 2 т. / под ред. Н. Н. Войта. – Ульяновск : УлГТУ, 2012. – С. 283–291.
- 62.Субхангулов Р.А., Филиппов А.А. Применение навигационной структуры электронного архива проектной организации в задачах интеллектуального анализа технических документов // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар. научн.техн. конф. (Минск, 19-21 февраля 2015 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2015. – С. 389–394.
- 63.Титов Ю. А. САПР технологических процессов. Ульяновск: УлГТУ, 2009.
- 64.Филиппов А.А. Индексирование и кластеризация проектных документов на основе графовой модели онтологии // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под. ред. Н. Н. Войта. – Ульяновск : УлГТУ, 2011. – С. 367–372.
- 65.Филиппов А.А. Концептуальный индексатор проектных документов // Тезисы докладов 45-й научно-технической конференции УлГТУ «Вузовская наука в современных условиях» (24-29 января 2011 года). – Ульяновск: УлГТУ, 2011. – С. 181–188.
- 66.Филиппов А.А. Нечеткая кластеризация концептуальных индексов проектных документов // Автоматизация процессов управления: сборник докладов Молодежной научно-технической конференции, Ульяновск, 13-14 декабря 2011 г. / под общ. ред. А.А. Емельянова. – Ульяновск: ФНПЦ ОАО «НПО «Марс», 2011. – С. 116–122.
- 67.Филиппов А.А., Субхангулов Р.А. Онтологически-ориентированная система интеллектуального анализа данных технических документов // Четырнадцатая национальная конференция по искусственному интеллекту с

международным участием КИИ-2014 (24-27 сентября 2014 г., г. Казань, Россия): Труды конференции. Т.3 – Казань: Изд-во РИЦ "Школа", 2014. – С. 228–236.

68.Хант Э. Искусственный интеллект. – М.: Мир, 1978.

69.Чекина Генетическая кластеризация технической документации в проектных репозиториях сапр: автореф. дис. на соиск. учен. степ.канд. технических наук (05.13.12); УлГТУ – Ульяновск., 2012 – 35 с.

70.Чесноков С.В. Детерминационный анализ социально-экономических данных. – М.: Наука, 1982

71.Ярушкина Н.Г. Основы теории нечетких и гибридных систем: Учеб. пособие. – М.: Финансы и статистика, 2004.

72.Bizer, C. Linked Data – The Story So Far / C. Bizer, T. Heath, T. Berners-Lee // International Journal on Semantic Web and Information Systems (IJSWIS), 2009.

73.Burges C. A Tutorial on Support Vector Machines for Pattern Recognition. URL: <http://research.microsoft.com/pubs/67119/svmtutorial.pdf>. (Дата обращения: 03.07.2015)

74.Cantador, I. Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations / I. Cantador, M. Szomszor, H. Alani, M. Fernández, P. Castells // In Proc of 1st Intern. Workshop on Collective Semantics: Collective Intelligence & the Semantic Web (CISWeb 2008), Tenerife, Spain. 2008.

75.Castells, P. An Adaptation of the Vector-Space Model for Ontology-based Information Retrieval / P. Castells, M. Fernández, D. Valle // IEEE Transactions on Knowledge and Data Engineering. – 2007. – 19 (2). – P. 261–272.

76.Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, An Intriduction to Information Retrieval. – URL: <http://nlp.stanford.edu/IR-book/> . Дата обращения: 15.03.2012.

- 77.Ciorascu C., Ciorascu I. and Stoffel K. knOWLer – Ontological Support for Information Retrieval Systems // Porceeding of 26th Annual International ACM SIGIR Conference. Workshop on Semantic Web, Toronto, Canada, August 2003.
- 78.Costa, A.C. Cores: Context-aware, ontology-based recommender system for service recommendation / A.C. Costa, R.S.S. Guizzardi, J.G.P. Filho // In Proc. 19-th Intern. Conf. on Advanced Information Systems Engineering (CAISE07). 2007.
- 79.Ellen M. Voorhees. Natural language processing and information retrieval. In Information Extraction: Towards Scalable, Adaptable Systems, P. 32–48, 1999.
- 80.Fei Song and W. Bruce Croft. A general language model for information retrieval (poster abstract). Research and Development in Information Retrieval, P. 279–280, 1999.
- 81.Gauch, S. Ontology-Based Personalized Search and Browsing / S. Gauch, J. Chaffee, A. Pretschner // ACM Web Intelligence and Agent System. – 2003. – Vol. 1. – No. 3/4. – P. 219–234.
- 82.Gonzalo J., Verdejo F., Chugur I., Cigarran J. Indexing with WordNet synsets can improve text retrieval. In: Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP.
- 83.Gruber Th. What is an Ontology // URL: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>. (Дата обращения: 04.07.2014)
- 84.Guarino N. Formal Ontology and Information Systems – URL: <http://www.mif.vu.lt/~donatas/Vadovavimas/Temos/OntologiskaiTeisingasKonceptinisModeliavimas/papildoma/Guarino98-Formal%20Ontology%20and%20Information%20Systems.pdf>. (Дата обращения: 03.05.2013)
- 85.Hassanpour S., DasA.K. Ontology-based text mining of concept definitions in biomedical literature. Proceedings of the Third Canadian Semantic Web Symposium (CSWS), Vancouver, Canada, 2011.
- 86.Haykin, S. Neural Networks: A Comprehensive Foundation. – New York: Macmillan College Publishing, 1994
- 87.Hotho, A., Maedche, A., Staab, S.: Ontology-based Text Clustering, Workshop «Text Learning: Beyond Supervision», IJCAI 2001.

- 88.Hovy E Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses – URL: http://www.researchgate.net/profile/Eduard_Hovy2/publication/228796674_Combining_and_standardizing_large-scale_practical_ontologies_for_machine_translation_and_other_uses/links/0046352557a2e83125000000.pdf (Дата обращения: 01.02.2014)
- 89.Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. // Proceedings of ICML-97, 14th International Conference on Machine Learning. – 1996
- 90.Joachims T. Estimating the Generalization Performance of a SVM Efficiently. // Proceedings of the International Conference on Machine Learning, – Morgan Kaufman, 2000
- 91.Joachims T. Making Large-Scale SVM Learning Practical. Advances in Kernel Methods / Support Vector Learning, Schölkopf B., Burges C., Smola A. (ed.), – MIT-Press, 1999.
- 92.Khaled Khelif, Rose Dieng-Kuntz and Pascal Barbry, An ontologybased approach to support text mining and information retrieval in the biological domain , Special Issue on Ontologies and their Applications of the Journal of Universal Computer Science (JUCS), Vol. 13, No. 12, P. 1881–1907, 2007.
- 93.Leung, K.W.T. Deriving Concept-Based User Profiles from Search Engine Logs / K.W.T. Leung, D.L. Lee // IEEE Transaction on Data and Knowledge Engineering. – 2010. – Vol. 22. – No. 7. – P. 969–982.
- 94.Liu, F. Personalized Web Search by Mapping User Queries to Categories / F. Liu, C. Yu, W. Meng // In Proc. of Intern. Conf. on Information and Knowledge Management (CIKM), 2002.
- 95.Maedche A., Zacharias V. Clustering ontology-based metadata in the semantic web. 13th European Conference on Machine Learning ECML'02 6th European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD'02, Helsinki, Finland, 2002.

96. Middleton, S.E. Ontological user profiling in recommender systems / S.E. Middleton, N.R. Shadbolt, D.C. de Roure // ACM Transaction on Information Systems. – 2004. – 22(1). – P. 54–88.
97. Milne D., Witten I. H. Learning to link with Wikipedia. In Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08). ACM, New York, NY, USA.
98. Muller, H.M., Kenny, E.E., Sternberg, P.W.: Textpresso: An OntologyBased Information Retrieval and Extraction System for Biological Literature. PLoS Biol. 2(11):e309.doi:10.1371/journal.pbio.0020309, 2004.
99. 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. (Дата обращения: 12.11.2013)
100. Pellet reasoning server: <http://clarkparsia.com/pellet/> (Дата обращения 18.07.2014).
101. Peña, P. Collective Knowledge Ontology User Profiling for Twitter / P. Peña, R. del Hoyo, J. Vea-Murguía, C. González, S. Mayo // 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT), 2013.
102. Pereira, R., Riarte, I., Gomide, F. Relational Ontology in Information Retrieval Systems. In: Fuzzy Databases and Data Mining, Pro. IFSA2005, Tsinghua University Press, 2005, 509–514.
103. Popov B., Kiryakov A., Ognyanoff D., Manov D. and Kirilov A., KIM – Semantic annotation platform for information extraction and retrieval. Natural Language Engineering, 10, Issues 3–4, P. 375–392, 2004.
104. Ratnov L., Roth D., Downey D., Anderson M. Local and Global Algorithms for Disambiguation to Wikipedia. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies -

Volume 1 (HLT '11), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 1375–1384.

105. Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval, Addison Wesley; 1st edition, 1999.

106. Roberts S.E. The probability ranking principle in IR Journal of Documentation, 33, 294–304.

107. Rocchio J. Relevance feedback in information retrieval – URL: <http://sigir.org/files/museum/pub-08/XXIII-1.pdf>. (Дата обращения: 15.03.2014)

108. Salton G, Allan J., and Buckley C. Approaches to Passage Retrieval in Full Text Information Systems. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, P. 49–58, 1993.

109. Salton G, Wong A, Yang C. A Vector Space Model for Automatic Indexing // Communications of the ACM, 18(11), 1975 P. 613–620

110. Salton G., Fox E., Wu H. Extended Boolean information retrieval. Communication of the ACM. 2001. Vol. 26. № 4. P. 35–43.

111. Salton, G., Automatic Text Processing. Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

112. Sanderson M. (1994). Word Sense Disambiguation and information retrieval. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

113. Stumme G., Hotho F., Berendt B. Semanti Web Mining. State of the art and future directions/ Web Semantis: Siene, Servies and Agents on the World Wide Web, 4, 2006, p. 124–143.

114. Su, Z.G. Research on Personalized Recommendation Algorithm Based on Ontological User Interest Model / Z.G. Su, J. Yan, H.F. Ling, H.S. Chen // J. of Computational Information Systems. – 2012. – Vol. 8. – No 1. – P. 169–181. Available also at <http://www.Jofcis.com/> (Актуально на 10.07.2014).

115. Trajkova, J. Improving Ontology-Based User Profile / J. Trajkova, S. Gauch // RIAO, 2004. – P. 380–390.

116. Turdakov D., Velikhov P. Semantic Relatedness Metric for Wikipedia Concepts Based on Link Analysis and its Application to Word Sense Disambiguation. In proceedings of the Fifth Spring Young Researchers Colloquium on Databases and Information Systems, SYRCoDIS'2008.
117. Vallet, D. Personalized Content Retrieval in Context Using Ontological Knowledge / D. Vallet, P. Castells, M. Fernández, P. Mylonas, Y. Avrithis //IEEE Trans. on Circuits and Systems for Video Technology. – 2007. – 17(3). – P. 336–346.
118. Vapnik V. N. Statistical Learning Theory. NY: John Wiley, 1998.
119. Vapnik V. The Nature of Statistical Learning Theory. New York, Springer-Verlag, 1995.
120. Xu, Y. Privacy-Enhancing Personalized Web Search / Y. Xu, K. Wang, B. Zhang, Z. Chen // Proceedings of World Wide Web (WWW) Conference, 2007. – P. 591–600.
121. Zeng, Y. User Interests: Definition, Vocabulary, and Utilization in Unifying Search and Reasoning / Y. Zeng, Y. Wang, Z.S. Huang, D. Damjanovic, Zh. Ning, C. Wang // In An A. et al. (Eds.): Active Media Technology 2010, Lecture Notes in Computer Science, vol. 6335, Springer, 2010. – P. 98–107.

Приложение А

УТВЕРЖДАЮ

Генеральный директор,

председатель НТС ФНПЦ

АО «НПО «Марс», к.т.н.

В.А. Маклаев

_____ 2015 г.



А К Т

об использовании результатов кандидатской диссертации Р.А. Субхангулова
“Онтологическая информационная поддержка проектирования в электронных
архивах технической документации”

Научно-техническая комиссия в составе:

председателя комиссии: первый заместитель генерального директора по
науке, к.т.н. Э.Д. Павлыгин,
членов комиссии: главный научный сотрудник, д.т.н.
Г.П. Токмаков,
начальник отдела ИАСУП, к.т.н. А.А. Перцев,
начальник отдела технической документации,
А.Е. Ефремов

настоящим актом подтверждает использование следующих научных и
практических результатов диссертационной работы Р.А. Субхангулова
“Онтологическая информационная поддержка проектирования в электронных
архивах технической документации” для анализа технических документов
электронного архива предприятия:

- алгоритм формирования контекстно-зависимых запросов к электронному архиву технических документов на основе байесовского классификатора и онтологической модели профиля проектировщика с целью информационной поддержки деятельности проектировщика;
- программная система информационной поддержки процесса проектирования автоматизированных систем на основе онтологии.

Комплекс программ онтологической информационной поддержки как

подсистема электронного архива предприятия использован при проектировании автоматизированных систем.

Эффективность использования научно-технических результатов подтверждена экспериментальными исследованиями, целью которых являлось определение количественной оценки качества выполнения проектных поисковых запросов к электронному архиву в сравнении с традиционными методами поиска электронных технических документов на основе набора ключевых слов.

Для реализации информационной поддержки проектирования автоматизированных систем в электронном архиве на ФНПЦ АО «НПО «Марс» была разработана прикладная онтология, содержащая в своем составе около 500 понятий и более 15000 уникальных терминов. Точность выполнения проектных поисковых запросов к электронному архиву с использованием онтологических моделей примерно на 30% лучше по сравнению с системами Яндекс.Персональный поиск и Архивариус 3000. Среднее время поиска технического документа в одном сеансе работы с электронным архивом сократилось примерно на 50% (с 14 минут до 6-7 минут).

Результаты получены в ходе выполнения и внедрения х/д НИР № 230/2005 «Интеллектуальный сетевой архив электронных информационных ресурсов», выполняемого Ульяновским государственным техническим университетом по заказу ФНПЦ АО «НПО «Марс».

Председатель комиссии:

Первый заместитель генерального директора
по науке, к.т.н.



Э.Д. Павлыгин

Члены комиссии:

Главный научный сотрудник, д.т.н.



Г.П. Токмаков

Начальник отдела ИАСУП, к.т.н.



А.А. Перцев

Начальник отдела технической документации



А.Е. Ефремов

Приложение Б

Онтология предметной области

Б.1 Фрагмент схемы онтологии

```
<!-- Life Circle -->
  <rdfs:Class rdf:ID="Stage"/>
  <rdfs:Class rdf:ID="StageConcept"/>
  <rdf:Property rdf:ID="Before">
    <rdfs:domain rdf:resource="#Stage" />
    <rdfs:range rdf:resource="#Stage" />
  </rdf:Property>
  <rdf:Property rdf:ID="PartOfStage">
    <rdfs:domain rdf:resource="#Stage" />
    <rdfs:range rdf:resource="#Stage" />
  </rdf:Property>
  <rdf:Property rdf:ID="ConnectToConcept">
    <rdfs:domain rdf:resource="#StageConcept" />
    <rdfs:range rdf:resource="#Concept" />
  </rdf:Property>
  <rdf:Property rdf:ID="ConnectToStage">
    <rdfs:domain rdf:resource="#StageConcept" />
    <rdfs:range rdf:resource="#Stage" />
  </rdf:Property>
<!-- Life Circle End -->

<!-- Concepts -->
  <rdfs:Class rdf:ID="Concept"/>
  <rdfs:Class rdf:ID="Term"/>
  <rdfs:Class rdf:ID="ConceptTerm"/>
  <rdfs:Class rdf:ID="ConceptInstance"/>
  <rdf:Property rdf:ID="PartOf">
    <rdfs:domain rdf:resource="#Concept" />
    <rdfs:range rdf:resource="#Concept" />
  </rdf:Property>
  <rdf:Property rdf:ID="SubclassOf">
    <rdfs:domain rdf:resource="#Concept" />
    <rdfs:range rdf:resource="#Concept" />
  </rdf:Property>
  <rdf:Property rdf:ID="ConnectToCTConcept">
    <rdfs:domain rdf:resource="#ConceptTerm" />
    <rdfs:range rdf:resource="#Concept" />
  </rdf:Property>
  <rdf:Property rdf:ID="ConnectToCTTerm">
```

```

        <rdfs:domain rdf:resource="#ConceptTerm" />
        <rdfs:range rdf:resource="#Term" />
    </rdf:Property>
    <rdf:Property rdf:ID="ConnectToCTFreq">
        <rdfs:domain rdf:resource="#ConceptTerm" />
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float" />
    </rdf:Property>
    <rdf:Property rdf:ID="ConnectToCIConcept">
        <rdfs:domain rdf:resource="#ConceptInstance" />
        <rdfs:range rdf:resource="#Concept" />
    </rdf:Property>
    <rdf:Property rdf:ID="ConnectToCIInstance">
        <rdfs:domain rdf:resource="#ConceptInstance" />
        <rdfs:range rdf:resource="#Instance" />
    </rdf:Property>
<!-- Concepts End -->

<!-- Instances -->
    <rdfs:Class rdf:ID="Instance"/>
    <rdfs:Class rdf:ID="InstanceTerm"/>
    <rdf:Property rdf:ID="ConnectToInstance">
        <rdfs:domain rdf:resource="#InstanceTerm" />
        <rdfs:range rdf:resource="#Instance" />
    </rdf:Property>
    <rdf:Property rdf:ID="ConnectToTerm">
        <rdfs:domain rdf:resource="#InstanceTerm" />
        <rdfs:range rdf:resource="#Term" />
    </rdf:Property>
    <rdf:Property rdf:ID="ConnectToFreq">
        <rdfs:domain rdf:resource="#InstanceTerm" />
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float" />
    </rdf:Property>
<!-- Instances End-->

```

Б.2 Фрагмент онтологии предметной области

```

<rdf:RDF
<!-- Ontology of design methodology---->
<Stage rdf:ID="Начало">
    <Before rdf:resource="# Проектирование " />
</Stage>
<Stage rdf:ID="Проектирование">
    <Before rdf:resource="# Построение" />
</Stage>
<Stage rdf:ID="Построение">
    <Before rdf:resource="# Внедрение " />
</Stage>
<Stage rdf:ID="Внедрение"/>
<Stage rdf:ID="Формирование_требований_к_АС" />
<Stage rdf:ID="Обследование_объекта_и_обоснование_необходимости_создания_АС">
    <PartOfStage rdf:resource="#Формирование_требований_к_АС" />
</Stage>
<Stage rdf:ID="Формирование_требований_пользователя_к_АС">
    <PartOfStage rdf:resource="#Формирование_требований_к_АС" />
</Stage>
<Stage rdf:ID="Оформление_отчёта_о_выполненной_работе_и_заявки_на_разработку_АС">
    <PartOfStage rdf:resource="#Формирование_требований_к_АС" />
</Stage>
...

```

```

<Stage rdf:ID="Стадия_сопровождения_АС" />
  <Stage rdf:ID="Выполнение_работ_в_соответствии_с_гарантийными_обязательствами">
<PartOfStage rdf:resource="#Стадия_сопровождения_АС" />
</Stage>
<!-- Ontology of design methodology End---->
<!-- Domain Ontology -->
<Concept rdf:ID="Система">
  <PartOf rdf:resource="#Общетеchnические_термины" />
</Concept>
<Concept rdf:ID="Информационная_модель">
  <SubclassOf rdf:resource="#Система" />
</Concept>
<Concept rdf:ID="Автоматизированный_производственный_комплекс">
  <SubclassOf rdf:resource="#Система" />
</Concept>
<Concept rdf:ID="Критерий_эффективности_деятельности">
  <PartOf rdf:resource="#Общетеchnические_термины" />
</Concept>
<Concept rdf:ID="Цель_деятельности">
  <SubclassOf rdf:resource="#Критерий_эффективности_деятельности" />
</Concept>
<Concept rdf:ID="Объект_деятельности">
  <SubclassOf rdf:resource="#Критерий_эффективности_деятельности" />
</Concept>
<Concept rdf:ID="Управление">
  <PartOf rdf:resource="#Общетеchnические_термины" />
</Concept>
<Concept rdf:ID="Автоматический_процесс">
  <SubclassOf rdf:resource="#Управление" />
</Concept>
...
<Concept rdf:ID="Диалоговый_ввод_заданий">
  <PartOf rdf:resource="#Сеанс_работы" />
</Concept>
<!-- Domain Ontology End -->
<!-- Terminological network of projects -->
<Instance rdf:ID="Microsoft_Solutions_Framework"/>
<Instance rdf:ID="Жизненный_цикл_программного_обеспечения"/>
<Instance rdf:ID="MSF"/>
<Instance rdf:ID="Visual_Studio_Team_System"/>
<Instance rdf:ID="UML"/>
<Instance rdf:ID="Диаграмма_последовательности"/>
<Instance rdf:ID="Диаграмма_классов"/>
<Instance rdf:ID="Диаграмма_пакетов"/>
<Instance rdf:ID="DFD"/>
<Instance rdf:ID="Диаграмма_прецедентов"/>
<Instance rdf:ID="Блоксхема"/>
<Instance rdf:ID="Разработка_программного_обеспечения"/>
...
<Term rdf:ID="разработк"/>
<Term rdf:ID="программн"/>
<Term rdf:ID="обеспечен"/>
<Term rdf:ID="microsoft"/>
<Term rdf:ID="solutions"/>
<Term rdf:ID="framework"/>
<Term rdf:ID="msf"/>
<Term rdf:ID="методолог"/>
<Term rdf:ID="предложен"/>
<Term rdf:ID="корпорац"/>
<Term rdf:ID="опира"/>
<Term rdf:ID="практическ"/>
<Term rdf:ID="описыва"/>
<Term rdf:ID="управлен"/>

```



```

<Term rdf:ID="людьм"/>
<Term rdf:ID="рабоч"/>
<Term rdf:ID="процесс"/>
<Term rdf:ID="решен"/>
<Term rdf:ID="представля"/>
<Term rdf:ID="соб"/>
<Term rdf:ID="согласова"/>
<Term rdf:ID="набор"/>
<Term rdf:ID="выпуст"/>
<Term rdf:ID="свет"/>
<Term rdf:ID="пакет"/>
<Term rdf:ID="руководств"/>
...
<InstanceTerm rdf:ID="CIndex1">
<ConnectToInstance rdf:resource="#Microsoft_Solutions_Framework" />
<ConnectToTerm rdf:resource="#разработк" />
<ConnectToFreq
rdf:datatype="http://www.w3.org/2001/XMLSchema#ConnectToFreq">0.5692307692307692</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex2">
<ConnectToInstance rdf:resource="#Microsoft_Solutions_Framework" />
<ConnectToTerm rdf:resource="#программн" />
<ConnectToFreq
rdf:datatype="http://www.w3.org/2001/XMLSchema#ConnectToFreq">0.4461538461538462</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex3">
<ConnectToInstance rdf:resource="#Microsoft_Solutions_Framework" />
<ConnectToTerm rdf:resource="#обеспечен" />
<ConnectToFreq
rdf:datatype="http://www.w3.org/2001/XMLSchema#ConnectToFreq">0.43846153846153846</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex4">
<ConnectToInstance rdf:resource="#Microsoft_Solutions_Framework" />
<ConnectToTerm rdf:resource="#microsoft" />
<ConnectToFreq
rdf:datatype="http://www.w3.org/2001/XMLSchema#ConnectToFreq">0.6076923076923078</ConnectToFreq>
</InstanceTerm>
<InstanceTerm rdf:ID="CIndex5">
<ConnectToInstance rdf:resource="#Microsoft_Solutions_Framework" />
<ConnectToTerm rdf:resource="#solutions" />
<ConnectToFreq
rdf:datatype="http://www.w3.org/2001/XMLSchema#ConnectToFreq">0.43846153846153846</ConnectToFreq>
</InstanceTerm>
</rdf:RDF>
<!-- Terminological network of projects End -->

```

Приложение С

Исходные коды основных алгоритмов программной системы

С.1 Алгоритм формирования концептуальной сети из внешних профессиональных wiki-ресурсов

```
public WikiResours(String[] listOfTitleStrings, int threshold, boolean backNetParam) {

    arrEdgeConcept.clear();
    arrConceptVertex.clear();
    WikiModel wikiModel = new WikiModel("http://ru.wikipedia.org/wiki/${image}",
    "http://ru.wikipedia.org/wiki/${title}");
    wikiModel.setUp();
    int porog = 1;
    this.threshold = threshold;
    this.backNet = backNetParam;
    this.count = this.count + 1;
    user.login();
    List<Page> listOfPages = user.queryContent(listOfTitleStrings);
    for (Page page : listOfPages) {
        String htmlContent = wikiModel.render(new PlainTextConverter(),
        page.toString());
        ConceptVertex myConceptVertex = new ConceptVertex(page.getPageid(),
        createConceptForSesame(page.getTitle(), page.getCurrentContent()));
        arrConceptVertex.add(myConceptVertex);
        HashMap<String, String> hMapLocal = new HashMap();
        Set<String> strLinks = wikiModel.getLinks();
        System.out.println(strLinks);
        Iterator<String> itr = strLinks.iterator();
        while (itr.hasNext()) {
            String it = itr.next();
            List<Page> listOfPagesLoc = user.queryContent(getArray(it));
            for (Page pageLoc : listOfPagesLoc) {
                String htmlContentLoc = wikiModel.render(pageLoc.toString());
                Set<String> strLinksLoc = wikiModel.getLinks();
                if ((pageLoc.getTitle().contains("год")) ||
                (pageLoc.getTitle().contains("век")) ||
                (pageLoc.getTitle().contains("Категория")) ||
```

```

        (pageLoc.getTitle().contains("Файл:")) ||
        (pageLoc.getTitle().contains("px")) ||
        checkMonth(pageLoc.getTitle()) ||
        pageLoc.getTitle().contains("Россия") ||
        pageLoc.getTitle().contains("РФ") ||
        pageLoc.getTitle().contains("Российская Федерация") ||
        (pageLoc.getTitle().contains("Thumb"))){
            continue;
        }
        if (checkConcept(createConceptForSesame(pageLoc.getTitle()))){
            continue;
        }
        if (this.backNet){
            if (!checkLink(strLinksLoc, page.getTitle())){
                continue;
            }else{
                EdgeConcept myEdgeConceptBack = new
                    EdgeConcept(createConceptForSesame(pageLoc.getTitle()),
                        createConceptForSesame(page.getTitle()));
                arrEdgeConcept.add(myEdgeConceptBack);
            }
        }else{
            EdgeConcept myEdgeConceptBack = new
                EdgeConcept(createConceptForSesame(pageLoc.getTitle()),
                    createConceptForSesame(page.getTitle()));
            arrEdgeConcept.add(myEdgeConceptBack);
        }
        ConceptVertex myConceptVertex1 = new ConceptVertex(pageLoc.getPageid(),
            createConceptForSesame(pageLoc.getTitle()), pageLoc.getCurrentContent());
        arrConceptVertex.add(myConceptVertex1);
        EdgeConcept myEdgeConceptTo = new
            EdgeConcept(createConceptForSesame(page.getTitle()),
                createConceptForSesame(pageLoc.getTitle()));
        arrEdgeConcept.add(myEdgeConceptTo);
        System.out.println(page.getTitle() + " - " + pageLoc.getTitle());
        RecursiveFunction(pageLoc, porog, wikiModel);
        String context = pageLoc.getCurrentContent();
        hMapLocal.put(pageLoc.getTitle(), context);
    }
}
}

```

С.2 Алгоритм формирования контекстно-ориентированных проектных запросов

```

private void Podgotovka(){
    hMapConceptMu.clear();
    hMapConceptMuNorm.clear();
    String strTerm = stopwords.deleteStopWords(this.Quest.getText());
    String[] arrayQuest = strTerm.split(" ");
    hMapQuest.clear();
    int n = arrayQuest.length;
    for (int i = 0; i < n; i++){
        this.hMapQuest.put(arrayQuest[i], (float) 1);
    }
    for (Entry exEntry : this.hMapOntologyProject.entrySet()){
        float SumDifference = 0;

```

```

Vector max = new Vector();
OntologyProject myOntologyProject =
hMapOntologyProject.get(exEntry.getKey());
for (Entry inEntry : myOntologyProject.hMapTermFreq.entrySet()) {
    boolean flagOnce = true;
    String termOntology = inEntry.getKey().toString();
    for (Entry questEntry : hMapQuest.entrySet()) {
        String termQuest = questEntry.getKey().toString();
        if (termOntology.equals(termQuest)) { // если терм запроса равен терму концепта из
            онтологии
            float Difference = Float.parseFloat(inEntry.getValue().toString()) -
            Float.parseFloat(questEntry.getValue().toString());
            SumDifference = SumDifference + Math.abs(Difference);
            if (Float.parseFloat(inEntry.getValue().toString()) >
            Float.parseFloat(questEntry.getValue().toString())) {
                max.add(Float.parseFloat(inEntry.getValue().toString()));
            }
            else {
                max.add(Float.parseFloat(questEntry.getValue().toString()));
            }
            flagOnce = false;
        }
    }
    if (flagOnce) {
        SumDifference = SumDifference +
        Float.parseFloat(inEntry.getValue().toString());
        max.add(Float.parseFloat(inEntry.getValue().toString()));
    }
}
float SumMax = 0;
for (int index = 0; index < max.size(); index++) {
    SumMax = SumMax + Float.parseFloat(max.elementAt(index).toString());
}
float mu = 1 - SumDifference/SumMax;
this.hMapConceptMu.put(exEntry.getKey().toString(), mu);
}
float max = 0;
int Count = 0;
for (Entry entry : this.hMapConceptMu.entrySet()) {
    if (Float.parseFloat(entry.getValue().toString()) != 0) {
        Count = Count + 1;
    }
    if (Float.parseFloat(entry.getValue().toString()) > max) {
        max = Float.parseFloat(entry.getValue().toString());
    }
}
for (Entry entry : this.hMapConceptMu.entrySet()) {
    this.hMapConceptMuNorm.put(entry.getKey().toString(),
    Float.parseFloat(entry.getValue().toString())/max);
}
if (this.hMapConceptMuNorm.isEmpty()) {
    for (Entry exEntry : this.hMapOntologyProject.entrySet()) {
        this.hMapConceptMuNorm.put(exEntry.getKey().toString(), (float) 0.0);
    }
}
for (Entry entryProfile : hMapUsersProfiles.entrySet()) {
    if (this.ActiveProfile.equals(entryProfile.getKey().toString())) {
        UsersProfiles myUserProfile = hMapUsersProfiles.get(entryProfile.getKey());
        Vector<Vector> vecPositiv = new Vector();
        Vector<Vector> vecNegativ = new Vector();
        //Формирование вектора позитивной информации
        for (Entry entryPosConcept:myUserProfile.hMapPositiveDocument.entrySet()) {
            Vector vec = (Vector) entryPosConcept.getValue();
            for (int i = 0; i < vec.size(); i++) {
                String concept = vec.elementAt(i).toString();
                boolean flg = true;
                for (int j = 0; j < vecPositiv.size(); j++) {
                    if (vecPositiv.elementAt(j).elementAt(0).equals(concept)) {
                        int count = Integer.parseInt(vecPositiv.elementAt(j).elementAt(1).toString());
                        count = count + 1;
                        vecPositiv.elementAt(j).setElementAt(count, 1);
                    }
                }
            }
        }
    }
}

```

```

        flg = false;
    }
}
if (flg){
    Vector locVec = new Vector();
    locVec.addElement(concept);
    locVec.addElement(1);
    vecPositiv.addElement(locVec);
}
}
//Формирование вектора негативной информации
for (Entry entryNegConcept:myUserProfile.hMapNegativeDocument.entrySet()){
    Vector vec = (Vector) entryNegConcept.getValue();
    for (int i = 0; i < vec.size(); i++){
        String concept = vec.elementAt(i).toString();
        boolean flg = true;
        for (int j = 0; j < vecNegativ.size(); j++){
            if (vecNegativ.elementAt(j).elementAt(0).equals(concept)){
                int count =
                    Integer.parseInt(vecNegativ.elementAt(j).elementAt(1).toString());
                count = count + 1;
                vecNegativ.elementAt(j).setElementAt(count, 1);
                flg = false;
            }
        }
        if (flg){
            Vector locVec = new Vector();
            locVec.addElement(concept);
            locVec.addElement(1);
            vecNegativ.addElement(locVec);
        }
    }
}
for (Entry entry : this.hMapConceptMuNorm.entrySet()){
    int countPositivConcept = 0;
    int countNegativConcept = 0;
    for (int i = 0; i < vecPositiv.size(); i++){
        String concept = entry.getKey().toString();
        if (vecPositiv.elementAt(i).elementAt(0).equals(concept)){
            countPositivConcept =
                Integer.parseInt(vecPositiv.elementAt(i).elementAt(1).toString());
        }
    }
    for (int i = 0; i < vecNegativ.size(); i++){
        String concept = entry.getKey().toString();
        if (vecNegativ.elementAt(i).elementAt(0).equals(concept)){
            countNegativConcept =
                Integer.parseInt(vecNegativ.elementAt(i).elementAt(1).toString());
        }
    }
    if (countNegativConcept > countPositivConcept){
        entry.setValue(Float.parseFloat("0"));
    }
}
}
if (this.flConceptWiki){
    for (Entry entry : this.hMapConceptMuNorm.entrySet()){
        for (int i = 0; i < arrayQuest.length; i++){
            for (Entry entryProfile : hMapUsersProfiles.entrySet()){
                if (this.ActiveProfile.equals(entryProfile.getKey().toString())){
                    UsersProfiles myUserProfile = hMapUsersProfiles.get(entryProfile.getKey());
                    for (Entry entryDoc : myUserProfile.hMapConceptWiki.entrySet()){
                        String concept = entry.getKey().toString().toString().replace("_", " ");
                        if (entryDoc.getKey().toString().equals(concept)){
                            Vector vec = (Vector) entryDoc.getValue();
                            for (int j = 0; j < vec.size(); j++){
                                ConceptVertex myConceptVertex = (ConceptVertex) (vec.elementAt(j));
                                if (myConceptVertex.context.contains(arrayQuest[i].toString())){
                                    this.hMapConceptMuNorm.put(entry.getKey().toString(), (float)1);
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
}

```

$$\left\{ \begin{array}{l} \\ \end{array} \right\}$$

С.3 Алгоритм вычисление меры сходства между контекстно-ориентированным проектным запросом и ТД

```

public ClassTraditionMethod(HashMap<String, Float> hMapConceptMu, HashMap<String,
                                IndexDocument> hMapIndex,
                                UsersProfiles myUserProfile){

    this.hMapConceptMu = hMapConceptMu;
    this.hMapIndex      = hMapIndex;
    double firstTerm    = firstTerm();
    arrayClassResult = new ClassResult[this.hMapIndex.size()];
    int index = 0;
    for (Entry entry :this.hMapIndex.entrySet()){
        IndexDocument IndexDoc = (IndexDocument) entry.getValue();
        double numerator = 0.0;
        double denominator = 0.0;
        double secondTerm = secondTerm(IndexDoc);
        Vector<Vector> vecPositiv = new Vector();
        Vector<Vector> vecNegativ = new Vector();
        try{
            for (Entry entryPosConcept:myUserProfile.hMapPositiveDocument.entrySet()){
                Vector vec = (Vector) entryPosConcept.getValue();
                for (int i = 0; i < vec.size(); i++){
                    String concept = vec.elementAt(i).toString();
                    boolean flg = true;
                    for (int j = 0; j < vecPositiv.size(); j++){
                        if (vecPositiv.elementAt(j).elementAt(0).equals(concept)){
                            int count =
                                Integer.parseInt(vecPositiv.elementAt(j).elementAt(1).toString());
                            count = count + 1;
                            vecPositiv.elementAt(j).setElementAt(count, 1);
                            flg = false;
                        }
                    }
                }
            }
            if (flg){
                Vector locVec = new Vector();
                locVec.addElement(concept);
                locVec.addElement(1);
                vecPositiv.addElement(locVec);
            }
        }
    }
}

catch(Exception ex){
}

//Формирование вектора негативной информации
try{

```

```

    for (Entry entryNegConcept:myUserProfile.hMapNegativeDocument.entrySet()) {
        Vector vec = (Vector) entryNegConcept.getValue();
        for (int i = 0; i < vec.size(); i++){
            String concept = vec.elementAt(i).toString();
            boolean flg = true;
            for (int j = 0; j < vecNegativ.size(); j++){
                if (vecNegativ.elementAt(j).elementAt(0).equals(concept)){
                    int count =
                        Integer.parseInt(vecNegativ.elementAt(j).elementAt(1).toString());
                    count = count + 1;
                    vecNegativ.elementAt(j).setElementAt(count, 1);
                    flg = false;
                }
            }
            if (flg){
                Vector locVec = new Vector();
                locVec.addElement(concept);
                locVec.addElement(1);
                vecNegativ.addElement(locVec);
            }
        }
    }
}

catch(Exception ex){
}

for (Entry entry2 : IndexDoc.hMapConceptFloat.entrySet()) {
    int countPositivConcept = 0;
    int countNegativConcept = 0;
    for (int i = 0; i < vecPositiv.size(); i++){
        String concept = entry2.getKey().toString();
        if (vecPositiv.elementAt(i).elementAt(0).equals(concept)){
            countPositivConcept =
                Integer.parseInt(vecPositiv.elementAt(i).elementAt(1).toString());
        }
    }
    for (int i = 0; i < vecNegativ.size(); i++){
        String concept = entry2.getKey().toString();
        if (vecNegativ.elementAt(i).elementAt(0).equals(concept)){
            countNegativConcept =
                Integer.parseInt(vecNegativ.elementAt(i).elementAt(1).toString());
        }
    }
    if (countNegativConcept > countPositivConcept){
        entry2.setValue(Float.parseFloat("0"));
    }
}

for (Entry entry2: IndexDoc.hMapConceptFloat.entrySet()) {
    String tmp = this.hMapConceptMu.get(entry2.getKey().toString()).toString();
    float firstMultiplier = Float.parseFloat(tmp);
    float secondMultiplier =
        Float.parseFloat(entry2.getValue().toString().replace(",", "."));
    numerator = numerator + firstMultiplier*secondMultiplier;
}

denominator = firstTerm*secondTerm;
double result = numerator/denominator;
if (!Double.isNaN(result)){
    arrayClassResult[index] = new ClassResult(entry.getKey().toString(),
        (float) (result));
    index = index + 1;
}

```

```
}  
else{  
    arrayClassResult[index] = new ClassResult(entry.getKey().toString(),  
        (float) (0));  
    index = index + 1;  
}  
}  
    Arrays.sort(arrayClassResult);  
}
```