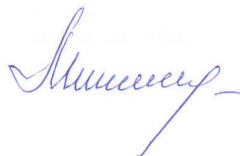


*На правах рукописи*



**Минашкин Сергей Александрович**

**МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
ИНТЕЛЛЕКТУАЛЬНЫХ ПОИСКОВЫХ СИСТЕМ НА ОСНОВЕ  
ИСПОЛЬЗОВАНИЯ МУЛЬТИАГЕНТНОЙ АРХИТЕКТУРЫ**

Специальность 05.13.11

Математическое и программное обеспечение вычислительных машин,  
комплексов и компьютерных сетей

Автореферат  
диссертации на соискание ученой степени  
кандидата технических наук

Рязань 2013

Работа выполнена на кафедре «Вычислительная и прикладная математика» ФГБОУ ВПО «Рязанский государственный радиотехнический университет» (ФГБОУ ВПО «РГРТУ»).

Научный  
руководитель:

**Каширин Игорь Юрьевич**, доктор технических наук, профессор, профессор каф. ВПМ ФГБОУ ВПО «РГРТУ»

Официальные  
оппоненты:

**Скворцов Сергей Владимирович** - доктор технических наук, профессор, профессор каф. САПР ВС ФГБОУ ВПО «РГРТУ»

**Швечков Виталий Алексеевич** - кандидат технических наук, старший научный сотрудник института инновационных образовательных проектов и проблем управления ФГБОУ ВПО «Российский государственный университет нефти и газа имени И.М. Губкина»

Ведущая  
организация:

ФГБОУ ВПО "Тамбовский государственный технический университет"

Защита состоится "25" декабря 2013 г. в 12.00 часов на заседании диссертационного совета Д212.211.01 в ФГБОУ ВПО «Рязанский государственный радиотехнический университет» по адресу: 390005, г. Рязань, ул. Гагарина, д. 59/1.

С диссертацией можно ознакомиться в библиотеке ФГБОУ ВПО «Рязанский государственный радиотехнический университет».

Автореферат разослан "18" ноября 2013 г.

Ученый секретарь

диссертационного совета

канд. техн. наук, доцент



В.Н. Пржегорлинский

## **ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ**

### **Актуальность темы**

В настоящее время глобальная сеть Интернет стала наиболее популярным средством поиска различной информации. Поисковые механизмы - ядро глобальной сети (ГС), их задача состоит в обеспечении эффективной навигации и выдаче искомой информации. Способ современного решения этой задачи - качественная система интеллектуального поиска, способная на любой запрос пользователя подбирать источники, максимально релевантные или пертинентные запросу. В этой области выполнено множество научных и технологических работ. Вместе с тем число информационных ресурсов в Интернете растет лавинообразно. При этом ресурсы в основной своей массе не имеют должного стандартного описания. Отсутствует их классификация. Так, например, при поиске информации по какой-либо тематической рубрике можно весьма долго просматривать Web-сайты, которые, на первый взгляд, описывают нужную рубрику, но используют бытовую лексику, не уделяя никакого внимания ни теоретическим основам, ни подробностям конкретных технологий. Это, как правило, коммерческие сайты, цель которых - любыми средствами привлечь посетителя в рекламных целях. Миллионы копирайтеров и рерайтеров производят большое количество некачественной информации. Таким образом, выявляется актуальная проблема - отсутствие стандартов описания и единообразной унифицированной системы классификации ресурсов ГС, а также отсутствие алгоритмов поиска с использованием систем, классифицирующих ресурсы.

Одним из способов решения данной проблемы является подход к описанию информационных ресурсов, разработанный ученым из Швейцарии Тимати Бернерсом-Ли. Им предложен семантический подход к описанию информационных ресурсов. Развитие подхода предполагает представление и запись информации в форме семантической сети с помощью онтологий. Онтология - это попытка всеобъемлющей и детальной формализации некоторой области знаний с помощью определённой концептуальной схемы. Таким образом, программа-клиент может непосредственно извлекать из ГС знания и делать из них логические заключения. Эта технология получила название Semantic Web.

Весомые результаты в работах по представлению знаний связаны с такими учеными, как В.П. Гладун, Д.А. Поспелов, Э.В. Попов, В.Ф. Хорошевский, И.Ю.Каширин, Д. Бобров, М. Минский, Р. Шенк, Т. Виноград.

Для реализации концепции Semantic Web консорциумом W3C, технологически развивающим Интернет, был предложен стандарт в форме языка описания онтологии OWL (Ontology Web Language). Язык OWL последовательно расширяется от простых его подмножеств к более сложным по схеме  $OWL-Lite \subseteq OWL-DL \subseteq OWL-Full$ . В настоящее время существует также OWL-S - язык описания Интернет-сервисов.

Семантическое описание требует разработки новых поисковых средств, повышающих релевантность и пертинентность на основе автоматического

построения программ небольшого объема – интеллектуальных поисковых агентов, которые получают информацию от ресурсов ГС и работают с ними для улучшения результирующей информационной выборки. Преимущества такого подхода в том, что агенты настроены на узкоспециализированную область, для которой они выдают наиболее точный результат в соответствии с предварительным семантическим описанием.

### **Цель работы**

Целью диссертационной работы является улучшение качества информационного поиска в ГС на основе создания новой архитектуры вспомогательных интеллектуальных средств за счет применения и построения поисковых агентов. Для построения интеллектуальных агентов необходимо разработать формализм представления знаний на основе современных языков семантического описания Интернет-ресурсов. Для повышения оперативности, релевантности и пертинентности поиска необходимо разработать математический аппарат, дающий возможность производить операции композиции и оптимизации на множестве OWL- ориентированных поисковых агентов.

Для практического решения поставленной цели необходима разработка алгоритмов, методов и программных средств, обеспечивающих надежную и эффективную работу интеллектуальных поисковых агентов.

### **Основные задачи**

Для достижения цели диссертации необходимо решение следующих задач:

- выполнить расширенный анализ существующих систем информационного поиска в ГС с целью выявления их технологических преимуществ и недостатков;
- в соответствии с рядом выявленных требований определить основные критерии для построения системы интеллектуального поиска;
- произвести анализ современных средств описания Интернет-ресурсов, позволяющих представлять их в форме знаний с формализацией семантики для разработки адекватного описания поисковых агентов;
- разработать математический аппарат, позволяющий производить операции на множестве поисковых агентов для оптимизации поиска;
- разработать новую архитектуру системы интеллектуального поиска;
- разработать алгоритмы и методы для работы с семантическими структурами и построения на их основе поисковых агентов;
- разработать программный инструментарий, реализующий созданную архитектуру и алгоритмы системы интеллектуального поиска для их практического применения, а также для качественной проверки математического аппарата, созданного для оптимизации интеллектуальных поисковых агентов.

### **Объект исследования**

Объектом исследования настоящей диссертации являются системы поиска информации в ГС, а также их оптимизация с помощью точных интеллектуальных агентов.

### **Методы исследования**

Теоретические исследования в диссертации выполнены на основе теории множеств, универсальных алгебр, прикладного исчисления предикатов. Прикладная часть работы использует теорию объектно-ориентированного программирования.

### **Научная новизна**

Разработан математический формализм и введены основные понятия для работы с интеллектуальными поисковыми агентами. Показано, что предложенный формализм достаточен для эффективного анализа современных поисковых машин. Для оптимизации поиска с целью повышения его пертинентности создана алгебраическая система OWL-ориентированных поисковых агентов, позволяющая производить операции композиции на множестве поисковых агентов. Предложен оригинальный алгоритм интеллектуального поиска на основе применения обучающихся агентов, позволяющий улучшить поисковые характеристики.

### **На защиту выносятся:**

- алгебраический формализм, позволяющий адекватно описывать композиционные свойства поисковых агентов, а также производить оптимизацию таких композиций;
- новое архитектурное решение для проектирования интеллектуальных поисковых машин, позволяющее создавать, адаптировать к конкретной предметной области и применять на практике специализированные поисковые агенты;
- инструментальный программный комплекс, реализующий на практике математические и архитектурные решения диссертационной работы.

### **Теоретическая значимость**

Теоретическая значимость работы заключается в разработке математического формализма:

- введены основные определения, понятия и операции для алгебры  $A_{OWL}$ , алгебры OWL-ориентированных поисковых агентов;
- выполнен анализ адаптивных свойств композиции поисковых агентов на основе  $A_{OWL}$ ;
- разработан способ обучения поисковой системы на основе решения уравнения  $AS_{OWL}$ .

### **Практическая значимость**

На основе полученных теоретических результатов автором диссертации разработан программный инструментарий (набор классов и методов языка программирования C#), в основе которого лежит объектно-ориентированное программирование, для создания интеллектуальных поисковых агентов и работы с ними. Эффективность применения инструментария достигается за счет улучшения следующих составляющих:

- упрощается процесс разработки; как следствие, сокращается время для создания поисковых систем с применением интеллектуальных агентов;

- на основе автоматического решения уравнения  $AS_{OWL}$  выполняется обучение поисковой системы для повышения релевантности и пертинентности поиска;

- на основе оригинального алгоритма сопоставления OWL-онтологий выполняется автоматическое построение поисковых агентов.

Программный инструментарий, реализованный в рамках диссертационной работы, дал возможность разработать программный продукт Spider v2.0, представляющий собой систему интеллектуального поиска с возможностью использования поисковых агентов как основных средств получения информации. Он позволяет производить поиск информации в ГС, имеет возможность описывать модель пользователя и на практике демонстрирует работу алгоритма обучения поисковых агентов. Выполненные эксперименты показали улучшение оперативности, релевантности и пертинентности поиска.

### **Реализация и внедрение результатов диссертационной работы**

Результаты диссертационной работы внедрены в управлении образования и молодежной политики администрации города Рязани, в производственной деятельности ФГУП "ГНПРКЦ "ЦСКБ-Прогресс" - ОКБ "Спектр", а также в учебном процессе Рязанского государственного радиотехнического университета при выполнении практических занятий по дисциплине «Проектирование систем искусственного интеллекта» в студенческих группах специальности 231000 «Программная инженерия».

### **Апробация работы**

Основные научные положения диссертационной работы докладывались и обсуждались на следующих научных конференциях:

- международной научно-практической конференции "Наука и образование в XXI веке" (Тамбов, 30 сентября 2013);

- международной научно-практической конференции "Актуальные научные вопросы и современные образовательные технологии" (Тамбов, 28 июня 2013);

- международной научно-практической конференции "Вопросы образования и науки в XXI веке" (Тамбов, 29 апреля 2013);

- международной научно-технической конференции "Проблемы передачи и обработки информации в сетях и системах телекоммуникаций" (Рязань, 6-8 мая 2012);

- 56-й студенческой научно-технической конференции РГРТУ, секция вычислительной и прикладной математики (Рязань, 22 апреля 2009).

### **Публикации**

По теме диссертации опубликовано 11 печатных работ, в том числе две статьи в издании, входящем в перечень ВАК для кандидатских и докторских диссертаций; 3 статьи в научно-технических журналах и межвузовских сборниках научных трудов; 6 докладов на международных и всероссийских конференциях.

## **Структура и объем диссертации**

Диссертационная работа состоит из введения, четырех глав, списка литературы, заключения, изложенных на 150 страницах (включая 23 рисунка и 15 таблиц), и 1-го приложения на 2-х страницах. Список литературы содержит 112 наименований.

## **СОДЕРЖАНИЕ РАБОТЫ**

**Во введении** обосновывается актуальность темы, сформулированы её цель, научная новизна, приведены сведения о практическом использовании полученных научных результатов и представлены основные положения, выносимые на защиту.

**В первой главе** проводится анализ концепций, подходов и методов информационного поиска в ГС, устанавливается, что одним из основных способов поиска информации в ГС Интернет являются поисковые машины.

Поисковые системы обычно состоят из трех компонент:

- агент (паук, робот или кроулер), который перемещается по информационным ссылкам ГС и собирает информацию;
- БД, которая содержит всю информацию, собираемую агентами;
- поисковый механизм, который используется как интерфейс для взаимодействия с БД.

*Агенты* - самые "интеллектуальные" из поисковых средств. Они могут выполнять намного больше операций, чем просто поиск. Например, они могут выполнять транзакции от имени пользователя без его участия, могут искать информационные ресурсы специфической тематики и возвращать их списки, отсортированные по параметру посещаемости. Агенты могут обрабатывать содержание документов, находить и индексировать другие виды ресурсов, причем не только текстовые страницы, но даже фотографии. Они могут также быть запрограммированы для извлечения информации из уже существующих БД. Независимо от информации, которую агенты индексируют, они передают ее в БД поискового механизма.

Общий поиск информации в ГС осуществляют программы, известные как пауки. Пауки сообщают о содержании найденного документа, индексируют его и извлекают итоговую информацию. Они также просматривают заголовки, некоторые ссылки и посылают проиндексированную информацию в БД поискового механизма.

*Кроулеры* просматривают заголовки и возвращают только первую ссылку, найденную в Интернете. Это - самые простые механизмы поиска, а потому наименее эффективные.

*Роботы* могут быть запрограммированы так, чтобы переходить по различным ссылкам различной глубины вложенности, выполнять индексацию и даже проверять ссылки в документе. Вследствие конструктивных особенностей они могут продолжительное время затрачивать на обработку циклических

ссылок информационных ресурсов, поэтому при проходе по ссылкам им нужны значительные ресурсы.

В основе принципа работы поискового механизма в настоящее время лежит достаточно сложный процесс обработки информации, но в упрощенном виде этот процесс можно описать следующим образом. Агенты извлекают и индексируют различные виды информации. Некоторые из них индексируют каждую словоформу во встречающемся документе, в то время как другие индексируют только несколько наиболее важных словоформ, фиксируют размер документа и число слов в нем, индексируют название, заголовки и подзаголовки и т. п. Вид построенного индекса определяет, какая форма поиска может быть выполнена поисковым механизмом и как полученная информация будет интерпретирована.

Поисковый механизм выявляет предмет запроса, основанный на информации, указанной в заполненной форме, и выводит соответствующие документы, подготовленные и хранимые в БД. Чтобы определить порядок, в котором список документов будет показан, механизм актуализации БД применяет алгоритм ранжирования. В идеальном случае документы, наиболее релевантные пользовательскому запросу, будут помещены первыми в ранжированном списке. Различные поисковые системы используют различные алгоритмы ранжирования, однако основные общие составляющие для вычисления релевантности следующие:

- количество слов запроса в текстовом содержимом документа (т.е. в HTML-коде);
- теги, в которых эти слова располагаются;
- местоположение искомых слов в документе;
- удельный вес слов, относительно которых определяется релевантность в общем количестве слов документа.

Доверие к информационному ресурсу определяется на основе сложных алгоритмов, в которых используется множество критериев, важнейшими из них являются количество ссылок, входящих на Web-сайт, и количество исходящих ссылок. Большинство поисковых систем имеет свои алгоритмы определения доверия.

В главе рассмотрены три основные русскоязычные поисковые системы: Яндекс, Google и Bing, выделены их особенности, размеры поисковых индексных баз, описаны поисковые возможности.

На основе доступной открытой информации выполнено описание математического формализма, используемого поисковыми системами.

Основу работы Google составляет алгоритм PageRank. Он используется для вычисления ранга полезности Web-страницы или какого-либо другого информационного ресурса. В диссертации подробно описан процесс его вычисления. Выполнен математический анализ формулы алгоритма, из которого следует, что основным фактором влияния на PageRank является количество входящих ссылок, количество исходящих ссылок. На результаты почти никак не влияет семантика представления данных, а тем более формализация их описания.



В настоящее время в основе работы поисковой системы Яндекс лежит алгоритм Matrixnet, использующий машинное обучение. Яндекс оперирует большим числом разных характеристик, например такими, как признаки страниц и запросов. Для вычисления признаков поисковая машина анализирует свойства Web-страниц и поисковых запросов. У всех страниц есть какие-либо признаки. Некоторые из них статические, связанные с самой Web-страницей. Некоторые признаки – динамические. Они связаны одновременно с запросом и Web-страницей. В диссертационной работе подробно описан процесс обучения и ранжирования. Описаны достоинства и недостатки этого алгоритма.

В конце первой главы сделан анализ функциональных особенностей и недостатков существующих систем информационного поиска. Выполнено сравнение поисковых машин Google и Яндекс. В главе определены основные требования, которые должны предъявляться к современным поисковым агентам:

- необходим переход к распределенной модели вычислений с привязкой к региону, из которого поступил запрос;
- необходим переход от модели «один поиск для всех» к модели персонального поиска;
- целесообразен переход от релевантности к пертинентности поиска;
- требуются разработка и применение новых технологий для описания существующих и вновь появляющихся информационных ресурсов.

В соответствии с требованиями сформулирована задача с детализацией подзадач для проектирования интеллектуально-адаптированных поисковых агентов, использующих формализацию семантики информационных ресурсов.

**Вторая глава** посвящена вопросам построения OWL-DL формализма для анализа и проектирования интеллектуальных поисковых агентов.

В начале главы дано краткое введение в технологию Semantic Web. На конкретных примерах рассмотрены стандарты языков семантического описания ресурсов ГС, таких как RDF, RDFS и OWL с подмножеством его языков OWL-Lite, OWL-DL, OWL-Full, а также рассмотрен язык онтологического описания Интернет-сервисов OWL-S.

Основной теоретической частью второй главы является разработка алгебры OWL-ориентированных поисковых агентов, позволяющей производить математические операции с поисковыми агентами для повышения оперативности, релевантности и пертинентности поиска. Язык OWL для этого формализма не является доминирующим, однако большинство примеров рассмотрены с использованием языка OWL-DL. Разработанная алгебра пригодна для исследования языков семантического описания, представленных в начале второй главы.

В главе вводятся основные определения.

Определение 1. Java, C, C++, C# - программа небольшого объема, которая может встраиваться в код HTML и выполнять поиск информационных ресурсов, необходимых пользователю, называется *поисковым агентом*. Агенты генерируются программным инструментарием на основе OWL-DL описания или

в результате автоматизированного построения такого описания после диалога с пользователем, осуществляющим поиск.

Определение 2. Поисковый агент, не использующий в своей работе встроенный анализатор текстового содержания информационных ресурсов, называется *браузерным агентом поиска*.

Определение 3. Поисковый агент, использующий встроенный анализатор текстового содержания информационных ресурсов, называется *агентом расширенного поиска*.

Определение 4. Поисковый агент, использующий исключительно встроенный анализатор содержания информационных ресурсов, а также унифицируемые OWL-конструкции в их метаописании, называется *расширением поиска*.

Для анализа функциональных возможностей композиций поисковых агентов описана соответствующая прикладная универсальная алгебра. В дальнейшем описании используются теоретико-множественные операции и отношения " $\cup$ ", " $\cap$ ", " $\supset$ ", " $\subset$ ", означающие соответственно объединение, пересечение множеств, обратное нестрогое и прямое нестрогое включения множеств. Кроме того, используются логические связки "&", " $\vee$ ", " $\Rightarrow$ ", означающие соответственно конъюнкцию, дизъюнкцию и импликацию аргументов. Знаки " $\forall$ ", " $\exists$ " соответствуют кванторам всеобщности и существования.

Пусть  $A$  – потенциально бесконечное множество поисковых агентов, а  $D$  – также возможно бесконечное множество информационных ресурсов (документов) ГС или достаточно большого информационного хранилища. Формально это можно записать следующим образом:

$A = \{a_1, a_2, \dots, a_i, \dots\}$ , где  $a_i$  -  $i$ -й поисковый агент,

$D = \{d_1, d_2, \dots, d_j, \dots\}$ , где  $d_j$  -  $j$ -й документ.

В этом случае множество  $D$  является универсумом информационных ресурсов, на котором ведется поиск.

Применение агента  $a_i$  в качестве фильтра к универсуму  $D$  приведет к получению некоторого подмножества  $D_i \subseteq D$ , соответствующего функциональной направленности поискового агента. В частном случае множество  $D_i$  может оказаться пустым:

$$\forall a_i, \exists D_i [a_i(D) = D_i] \ \& \ [D_i \subseteq D \vee D_i = \{\emptyset\}].$$

Очевидно, что на универсуме поисковых ресурсов существует универсальная алгебра множеств, содержащая в своей сигнатуре операции дополнения, объединения и пересечения. Описание этой алгебры в силу ее известности оставлено за рамками диссертации.

Кроме того, как рассматривалось ранее, множество поисковых агентов  $A$  можно разделить на три подмножества:

$A = \bar{A} \cup \tilde{A} \cup \hat{A}$ , где  $\bar{A}$  - множество браузерных агентов,  $\tilde{A}$  - множество агентов расширенного поиска,  $\hat{A}$  - множество композиций агентов из  $\bar{A}$  и  $\tilde{A}$ . Предполагается также, что в каждом из этих подмножеств существуют нулевой

агент, обозначаемый “ $O$ ”, и единичный агент, обозначаемый “ $\lambda$ ”. Смысл первого из них в том, что он, не осуществляя поиска, выдает после своей работы пустое множество информационных ресурсов, в то время как второй выдает после своей работы все множество ресурсов, которое у него было на рассмотрении (на входе):

$$O(D_i) = \emptyset, \quad \lambda(D_i) = D_i.$$

В описанном множестве  $A$ , так же как и в его подмножествах, можно выделить базис элементарных поисковых агентов  $A_0, \bar{A}_0, \tilde{A}_0, \hat{A}_0$ , неразложимых при их декомпозиции на более простые:

$$A_0 = \bar{A}_0 \cup \tilde{A}_0 \cup \hat{A}_0.$$

Алгебра OWL-ориентированных поисковых агентов  $A_{OWL}$  описывается следующей двойкой:  $A_{OWL} = \langle A, \Omega \rangle$ ,

где  $A$  – рассмотренное ранее множество всех поисковых агентов, а  $\Omega$  – сигнатура алгебры, т.е. множество операций, замкнутых на множестве  $A$ .  $\Omega$  является конечным множеством и состоит из следующих операций:

$$\Omega = \{\bullet, \uparrow, \downarrow, \uparrow\uparrow, \downarrow\downarrow, \oplus\},$$

где “ $\bullet$ ” – операция последовательной композиции агентов,

“ $\uparrow$ ” – операция дополнения агента соответствующим ему расширением поиска,

“ $\downarrow$ ” – операция удаления расширения из агента расширенного поиска,

“ $\uparrow\uparrow$ ” – операция дополнения расширения поиска соответствующим ему нерасширенным агентом поиска,

“ $\downarrow\downarrow$ ” – операция удаления нерасширенного агента поиска из расширенного агента поиска,

“ $\oplus$ ” – операция объединения агентов, при которой получается новый агент, включающий после своей работы результаты двух агентов.

Таким образом, алгебра  $A_{OWL}$  имеет следующий тип:

$$\langle O(0), \lambda(0), \bullet(2), \uparrow(1), \downarrow(1), \uparrow\uparrow(1), \downarrow\downarrow(1), \oplus(2) \rangle,$$

где в круглых скобках указывается местность операций (число аргументов). Операции с нулевым числом аргументов являются выделенными константами.

Теоретико-множественная семантика операций дана в таблице 1.

Для оперирования со сложными выражениями алгебры  $A_{OWL}$  с целью их оптимизации, а следовательно и оптимизации поиска, приведены основные свойства ее операций, следующие из семантики операций:

1) свойства констант:

$$a_i \bullet O = O; \quad a_i \oplus O = a_i; \quad a_i \bullet \lambda = a_i; \quad a_i \oplus \lambda = \lambda,$$

$$\uparrow O = O; \quad \downarrow O = O; \quad \uparrow\uparrow O = O; \quad \downarrow\downarrow O = O; \quad \uparrow\lambda = \lambda; \quad \downarrow\lambda = \lambda; \quad \uparrow\uparrow\lambda = \lambda; \quad \downarrow\downarrow\lambda = \lambda;$$

2) ассоциативность:

$$(a_i \oplus a_j) \oplus a_k = a_i \oplus (a_j \oplus a_k);$$

в общем случае операция “ $\bullet$ ” ассоциативностью не обладает;

3) идемпотентность:

$$a_i \oplus a_i = a_i; \quad a_i \bullet a_i = a_i;$$

4) коммутативность:

$$a_i \oplus a_j = a_j \oplus a_i;$$

в общем случае операция “•” коммутативностью не обладает;

Таблица 1- Семантика операций

| № п/п | Наименование операции          | Обозначение операции      | Семантическое описание                                                                                                                                              |
|-------|--------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | Нулевая константа              | $0(0)$                    | $\forall D_t \subseteq D, 0(D_t) = \emptyset$                                                                                                                       |
| 2     | Единица алгебры                | $\lambda(0)$              | $\forall D_t \subseteq D, \lambda(D_t) = D_t$                                                                                                                       |
| 3     | Последовательная композиция    | $\bullet(2)$              | $\forall a_i \in A, \forall a_j \in A, \forall D_t \subseteq D, \exists a_{ij} \in A, [a_{ij} = a_i \bullet a_j] \Rightarrow [a_{ij}(D_t) = a_i(a_j(D_t))]$         |
| 4     | Дополнение расширением поиска  | $\uparrow(1)$             | $\forall a_i \in A, \forall D_t \subseteq D, \exists a_j \in \tilde{A}, [a_j = \uparrow a_i] \Rightarrow [a_i(D_t) \subseteq a_j(D_t)]$                             |
| 5     | Удаление расширения            | $\downarrow(1)$           | $\forall a_j \in A, \forall D_t \subseteq D, \exists a_i \in \bar{A}, [a_i = \downarrow a_j] \Rightarrow [a_i(D_t) \supseteq a_j(D_t)]$                             |
| 6     | Дополнение расширенным агентом | $\uparrow\uparrow(1)$     | $\forall a_i \in \bar{A}, \forall D_t \subseteq D, \exists a_j \in A, [a_j = \uparrow\uparrow a_i] \Rightarrow [a_i(D_t) \subseteq a_j(D_t)]$                       |
| 7     | Удаление расширенного агента   | $\downarrow\downarrow(1)$ | $\forall a_j \in \tilde{A}, \forall D_t \subseteq D, \exists a_i \in \bar{A}, [a_i = \downarrow\downarrow a_j] \Rightarrow [a_i(D_t) \supseteq a_j(D_t)]$           |
| 8     | Объединение агентов            | $\oplus(2)$               | $\forall a_i \in A, \forall a_j \in A, \forall D_t \subseteq D, \exists a_{ij} \in A, [a_{ij} = a_i \oplus a_j] \Rightarrow [a_{ij}(D_t) = a_i(D_t) \cup a_j(D_t)]$ |

5) множественное использование унарных операций:

$$\uparrow\uparrow a_i = \uparrow a_i; \uparrow\uparrow\uparrow a_i = \uparrow\uparrow a_i; \downarrow\downarrow a_i = \downarrow a_i; \downarrow\downarrow\downarrow a_i = \downarrow a_i.$$

Приведенные свойства справедливы для любых множеств поисковых агентов и не требуют рассмотрения многосортности множества-носителя  $A$ . В то же время учет сортности как принадлежности каждого из поисковых агентов к одному из множеств  $A, \tilde{A}, \bar{A}, \hat{A}_0$  дает возможность найти более сильные свойства операции последовательной композиции “•”. Далее приводятся эти основные свойства.

1. Коммутативность:

$$\forall a_i \in \tilde{A}, \forall a_j \in \tilde{A}, \forall D_t \subseteq D, \exists a_{ij} \in A, [a_i \bullet a_j = a_j \bullet a_i],$$

$$\forall a_i \in \bar{A}, \forall a_j \in \bar{A}, \forall D_t \subseteq D, \exists a_{ij} \in A, [a_i \bullet a_j = a_j \bullet a_i],$$

$$\forall a_i \in \hat{A}, \forall a_j \in \hat{A}, \forall D_t \subseteq D, \exists a_{ij} \in A, [a_i \bullet a_j = a_j \bullet a_i].$$

2. Дистрибутивность слева и справа:

$$\forall a_i \in \tilde{A}, \forall a_j \in \tilde{A}, \forall a_k \in \tilde{A}, \forall D_t \subseteq D, [(a_i \oplus a_j) \bullet a_k = a_i \bullet a_k \oplus a_j \bullet a_k],$$

$$\begin{aligned}
& \forall a_i \in \tilde{A}, \forall a_j \in \tilde{A}, \forall a_k \in \tilde{A}, \forall D_i \subseteq D, [a_k \bullet (a_i \oplus a_j) = a_k \bullet a_i \oplus a_k \bullet a_j], \\
& \forall a_i \in \bar{A}, \forall a_j \in \bar{A}, \forall a_k \in \bar{A}, \forall D_i \subseteq D, [(a_i \oplus a_j) \bullet a_k = a_i \bullet a_k \oplus a_j \bullet a_k], \\
& \forall a_i \in \bar{A}, \forall a_j \in \bar{A}, \forall a_k \in \bar{A}, \forall D_i \subseteq D, [a_k \bullet (a_i \oplus a_j) = a_k \bullet a_i \oplus a_k \bullet a_j], \\
& \forall a_i \in \hat{A}, \forall a_j \in \hat{A}, \forall a_k \in \hat{A}, \forall D_i \subseteq D, [(a_i \oplus a_j) \bullet a_k = a_i \bullet a_k \oplus a_j \bullet a_k], \\
& \forall a_i \in \hat{A}, \forall a_j \in \hat{A}, \forall a_k \in \hat{A}, \forall D_i \subseteq D, [a_k \bullet (a_i \oplus a_j) = a_k \bullet a_i \oplus a_k \bullet a_j].
\end{aligned}$$

Положительным качеством приведенных свойств является возможность оптимизации структуры поисковых агентов с точки зрения их вычислительной сложности, совместное использование коммутативности и дистрибутивности дает возможность производить обратное оптимизирующее преобразование:

$$a_k \bullet a_i \oplus a_k \bullet a_j = a_k \bullet (a_i \oplus a_j).$$

Поисковые агенты являются компьютерными программами, работающими в соответствии с какими-либо алгоритмами, для них справедливо свойство, связанное с понятием вычислительной сложности. В случае алгоритмов поиска сложность алгоритма можно вычислять как число обращений к поисковому серверу или вспомогательным программам для выполнения извлечений информационных ресурсов из ГС или информационного хранилища. Далее в главе произведен анализ адаптивных свойств композиций OWL-ориентированных поисковых агентов.

Введено понятие *коэффициента сложности поискового агента*, который вычисляется следующим образом. При определенном универсуме документов (информационных ресурсов) информационного хранилища  $D$  и для конкретного поискового агента  $a_j$  выделяется абстрактное понятие «число просматриваемых  $i$ -м агентом данных информационных ресурсов  $\forall D_i \subseteq D$ »  $|D_i|$ . Этими данными могут быть в реальности индексные массивы или деревья, метаданные в формате HTML или OWL или даже сами тексты искомых документов. Таким образом, максимальным числом просматриваемых документов является  $|D|$ .

Определение 5. Результат вычисления функции  $\Theta(a_i)$  для поискового агента  $a_i$  называется *коэффициентом сложности* поискового агента  $a_i$ .

Определение 6. Поисковые агенты, полученные из более простых поисковых агентов исключительно с помощью операций, имеющих свойства коммутативности и дистрибутивности, называются *регулярными* поисковыми агентами. Агенты, не обладающие такими свойствами, называются *нерегулярными*.

Далее приведено описание обучения поисковой системы, которое фактически сводится к построению достаточно полного и точного по своим результатам поискового агента за счет правильного подбора последовательности операций из алгебраической системы  $AS_{OWL}$  для получения пертинентного поискового агента.

Алгебраическая система  $AS_{OWL}$  -это алгебра  $A_{OWL} = \langle A, \Omega \rangle$ , расширенная

до - алгебраической системы  $AS_{OWL}$  добавлением в нее множества  $R$  следующих отношений:

$$AS_{OWL} = \langle A, \Omega, R \rangle; R = \{=_f, \neq, \leq, \geq, <, >\},$$

где “ $=_f$ ” – выделенное отношение функционального равенства поисковых агентов,

“ $=, \neq$ ” – отношения соответственно равенства и неравенства коэффициентов сложности поисковых агентов,

“ $\leq, \geq, <, >$ ” – отношения нестрогих и строгих неравенств поисковых агентов по их коэффициентам сложности.

На первых этапах обучения все информационные ресурсы  $D_{res}$ , выданные пользователю после работы агента, соответствующего подобранным и ранжированным ключевым словам и словосочетаниям, можно разделить на два непересекающихся подмножества:

$$D_{res} = \{D_+, D_-\}, D_+ \cap D_- = \emptyset,$$

где  $D_+$  - релевантные ресурсы,  $D_-$  - нерелевантные ресурсы.

Кроме того, в универсуме  $D$  информационного хранилища, на ресурсах которого осуществляется поиск, имеются документы  $D_{out}$ , которые должны были войти в множество  $D_+$ , но не вошли в результате несовершенства поискового агента.

Если рассмотреть основные свойства алгебраической системы  $AS_{OWL}$ , становится понятно, что нерелевантные ресурсы  $D_-$  можно исключить лишь добавлением последовательной композиции поискового агента. Для этого необходимо решить уравнение, в котором бы одним из членов был неизвестный поисковый агент  $X$ , работающий как фильтр для документов  $D_-$ :

$$a_i \bullet X = a_i, a_i(D) = D_{res} = D_+ \cup D_-, a_i(D) = D_+.$$

В то же время поисковый агент  $a_i(D) = D_{res}$  должен быть составлен таким образом, чтобы включать в свой результат элементы  $D_{out}$ . Для этого потребуются решение уравнения, в котором бы одним из членов был неизвестный агент  $Y$ , работающий на добавление информационных ресурсов  $D_{out}$ :

$$a_i \oplus Y = a_i, a_i(D) = D_{res} = D_+ \cup D_-, a_i(D) = D_+ \cup D_{out}.$$

Таким образом, должна быть решена система уравнений из двух только что описанных уравнений. Решение следует искать как подстановку  $\{X \rightarrow a_t, Y \rightarrow a_s\}$ , дающую композиционный поисковый агент:

$$(a_i \bullet X) \oplus Y = a_{res} = (a_i \bullet a_t) \oplus a_s, a_{res}(D) = D_+ \cup D_{out}.$$

Сложность такого решения обусловливается связанностью требований одновременного уточнения и расширения результата  $D_{res}$ . Например, можно решать и другое уравнение:

$$(a_i \oplus Y) \bullet X = a_{res} = (a_i \bullet a_t) \oplus a_s, a_{res}(D) = D_+ \cup D_{out}.$$

Эта задача упрощается, если на момент решения системы уравнений в множестве-носителе алгебры уже эксплицитно выделены (сгенерированы ранее) элементы  $a_t, a_s$ . Как правило, на практике приходится получать эти элементы заново, настраивая агенты по дополнительным ключевым словам и словосочетаниям, сообщаемым пользователем в процессе цикла обучения

поисковой системы. Алгоритм обучения может работать циклически, попеременно, то комплексируя последовательные дополнительные агенты, играющие роль фильтров, то добавляя расширяющие элементы.

Во второй главе выполнена демонстрация применимости OWL для описания информационных ресурсов в ГС и приведено подробное описание на примере раздела научных знаний, представленных на рис.1. Описание может использоваться для автоматического построения поисковых агентов.

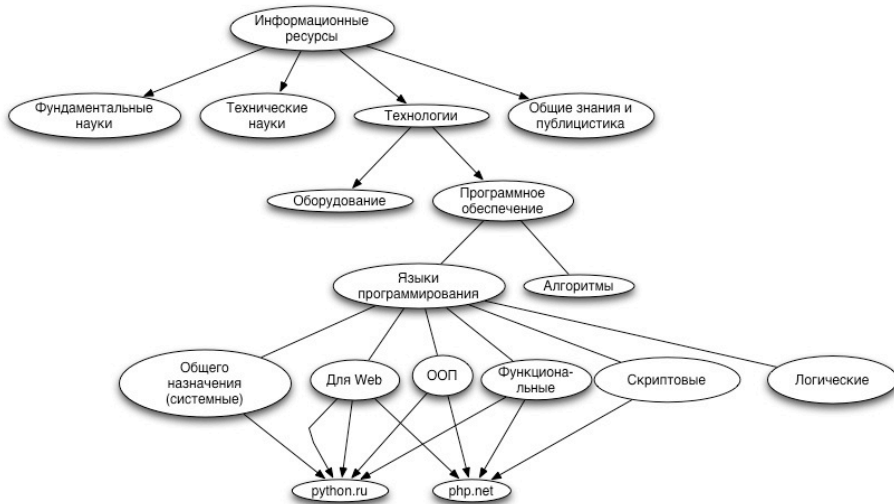


Рис. 1. Схема родовидовой таксономии

В конце второй главы приведен пример формализации интеллектуальных поисковых агентов с использованием OWL-S описания.

**В третьей главе** рассмотрены архитектурные и проектные решения при автоматизации проектирования интеллектуальных поисковых агентов. Целью главы является изложение особенностей архитектуры системы мультиагентной модульной структуры для интеллектуального поиска в ГС.

Далее представлены общие принципы архитектуры современных систем интеллектуального поиска в ГС. Архитектура в общем случае состоит из ряда модулей, которые связаны друг с другом в функциональном плане и представляют собой завершённую программную платформу. В главе рассматриваются все компоненты системы, включая БД, в которой централизованно хранится основная информация.

Далее приведены особенности архитектуры информационного поиска в поисковой машине Яндекс и выполнен анализ архитектуры поисковой системы Google. Кроме того, рассмотрены вопросы программной реализации поисковых систем, таких как Russian Context Server, Convera RetrievalWare, Yandex, Google.

На основе выполненного анализа предложена новая мультиагентная архитектура системы интеллектуального поиска (рис. 2).

Приведены основные преимущества такой архитектуры.

1. В поиске участвует модель пользователя, что является хорошим дополнением к короткому запросу пользователя и позволяет перейти от всеобщего поиска к индивидуальному, что приведет к более пертинентному поиску.

2. Поиск выполняется не одной поисковой машиной по ее индексу, а множеством поисковых агентов с их разнообразной индексной базой. Вместе с тем следствием такого поиска является большое количество документов в результате. Это может привести к ухудшению релевантности и пертинентности. Для устранения этого применяется алгебра поисковых агентов, которая позволяет производить операции оптимизации и композиции сервисов агентов и тем самым уменьшать количество документов в итоге для достижения приемлемой точности результата.

3. Архитектура предусматривает возможность обучения агентов на основе данных, полученных от пользователя при решении уравнения  $AS_{OWL}$ . Такой подход делает поиск более персональным и, как следствие, более пертинентным.

4. Архитектура системы составлена из отдельных локальных компонентов, что позволяет использовать для разработок и реализации уже готовые открытые Интернет-сервисы.

В этой же главе предложен алгоритм системы интеллектуального поиска с использованием мультиагентного подхода (рис. 3).

Далее в форме последовательности шагов описаны алгоритмы "обучения и самообучения" интеллектуальных агентов на основе решения уравнения алгебры  $AS_{OWL}$ .

Обучение поисковой системы фактически сводится к построению достаточно полного и точного по своим результатам поискового агента за счет правильного подбора последовательности операций из алгебраической системы  $AS_{OWL}$ . Для этого пользователю необходимо оценить полученный результат по двухбалльной шкале «0» и «1». Оценка «0» ставится, если документ или ресурс в выдаче не удовлетворяет запросу пользователя. Оценка «1» ставится в случае, если результат хотя бы минимально удовлетворяет запросу пользователя. Пользователь, по возможности, должен оценить как можно больше документов в выдаче. Таким образом, на первом этапе обучения получается множество информационных ресурсов, разделенных на два непересекающихся множества в соответствии с их оценками.





После выполнения оценки результатов можно уточнить поисковый запрос, добавив в него исключающие "стоп"-слова и уточняющие ключевые слова.

Далее система сопоставляет полученные результаты с используемыми поисковыми агентами, новыми поисковыми данными и автоматически расставляет оценки каждому агенту. Для этого используется оценочная функция

$$M=Y*10-N,$$

где  $M$  - оценка поискового агента,  $Y$  - число документов с оценкой «1»,  $N$  - число документов с оценкой «0». Далее, в зависимости от оценки агента, система делает один из выводов:

- нужно ли оставлять агента в композиции на своем месте, если  $M>0$ ,
- нужно ли перенести агента в начало композиции, если  $M>10$ ,
- нужно ли исключить агента из композиции, если  $M<0$ .

После того как все оценки расставлены, система выбирает агентов из БД на основе новых поисковых данных, введенных пользователем. Далее производится сопоставление рейтингов найденных агентов с рейтингами агентов, составленных на основе оценочной функции, и включение или исключение агентов из результирующей композиции.

Когда список агентов получен, системе необходимо попытаться упростить задачу применения агентов, используя свойства операций алгебры  $A_{OWL}$ . Применяя эти свойства, можно изменить получаемый результат, например изменяя порядок исполнения агентов в композиции с использованием результатов оценочной функции.

В заключение полученная цепочка агентов передается на исполнение основному алгоритму интеллектуального поиска и, таким образом, цикл замыкается.

Далее в главе представлены проектные решения для программной реализации интеллектуальных агентов (объектно-ориентированные классы).

Agent - класс для создания и работы с ключевым определением  $A_{OWL}$  - поисковым агентом.

AgentMath - статический класс для выполнения математических операций композиции, объединения, дополнения и расширения поиска алгебры  $A_{OWL}$  над агентами.

User - класс для описания модели пользователя и работы с ней.

Search - класс для выполнения четкого описания поискового запроса, который может состоять как из одного ключевого слова, так и из целого набора параметров для поиска, включающего в себя OWL-описание.

PageParser – класс, с помощью которого можно производить парсинг HTML- документов, например извлекать <title>-теги , <meta>-теги или весь документ в HTML-формате и т. д.

Analysis - класс с набором свойств и методов для синтаксического анализа HTML -документа.

Помимо представленных выше классов было разработано множество вспомогательных классов для программной реализации, получившей название

Spider v2.0. Например, класс Dictionary предоставляет набор свойств и методов для работы со словарем синонимов.

**В четвертой главе** рассматривается программный продукт Spider v2.0, разработанный в рамках диссертационной работы. Цель программы - улучшить поисковые характеристики существующих поисковых машин. Назначение программы - продемонстрировать на практике основные теоретические результаты, полученные в предыдущих главах. При этом предложена новая архитектура, получены проектные решения, связанные с разработкой и применением интеллектуальных поисковых агентов, выполняющих поиск информации в ГС Интернет.

На основе проведенного исследования конкретизированы функции, которые выполняет разработанный инструментарий.

1. Система выполняет поиск, исходя из индивидуальной модели пользователя. Для решения этой задачи приложение имеет программное описание модели пользователя и при поиске исходит из его потребностей в текущий момент времени.

2. Для описания Web-ресурсов и запросов пользователя применяется предложенная в предыдущих главах структуризация OWL-описания.

3. В инструментарии реализованы механизм работы с алгеброй OWL-ориентированных поисковых агентов и основные операции над поисковыми агентами.

4. Архитектура инструментария соответствует архитектуре интеллектуального поиска, предложенной в третьей главе диссертации.

5. В основе алгоритма работы системы лежит алгоритм работы системы интеллектуального поиска, также предложенный в третьей главе.

6. Система обладает возможностью обучения интеллектуальных поисковых агентов для получения оптимальных результатов поисковой выдачи.

Основной особенностью инструментария является алгоритм сопоставления понятий из онтологий (рис. 4).

Поисковый инструментарий Spider v2.0 может осуществлять поиск по различным видам запросов.

1. Поиск по ключевому слову на естественном языке - это стандартный поиск. Для его выполнения был разработан словарь синонимов со своим набором классов и методов.

2. Поиск на основе ввода готового OWL-описания, которое может включать в себя все конструкции языка OWL-Full. Программный инструментарий не осуществляет интерпретацию языка описания онтологий, он лишь осуществляет разбор и сопоставление конструкций.

3. Поиск информации для текущей модели пользователя без запроса осуществляется на основе информации об образовании, специальности и должности пользователя с применением словаря синонимов. Для этого метода поиска можно добавлять специальные поисковые агенты, например агенты новостных серверов, таких как РБК, mail.ru. Инструментарий самостоятельно

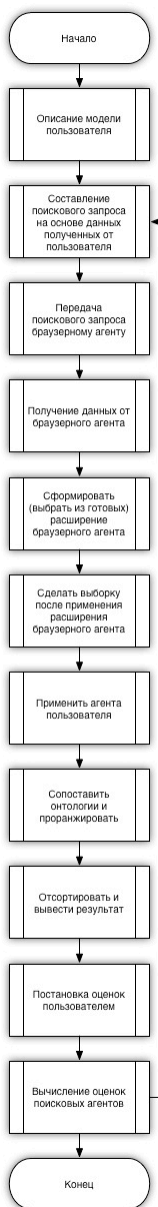


Рис. 3. Основной алгоритм системы интеллектуального поиска с использованием мультиагентной архитектуры

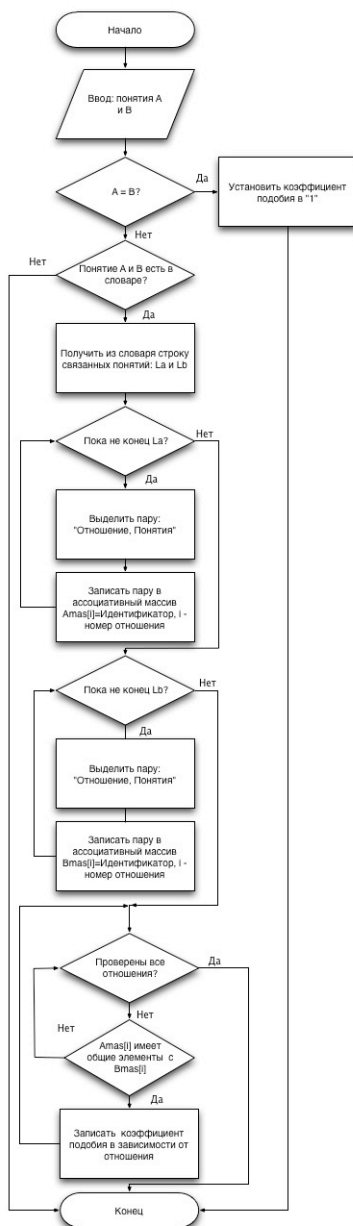


Рис. 4. Алгоритм сопоставления онтологий

пытается определить информационную потребность пользователя и выводит на экран всю найденную информацию. После этого пользователь отмечает pertinentные результаты, на основе чего агенты "переобучаются". Это означает, что средствами алгебры  $A_{OWL}$  (глава 2) подбирается удовлетворяющий пользователя вариант и система начинает выдавать пользователю pertinentную информацию как результат работы алгоритма обучения.

Для создания пользовательского интерфейса были выбраны среда разработки Microsoft Visual Studio 2010 и языки программирования C#, XML, HTML, Transact-SQL.

Все графические элементы управления выполнены в системе Microsoft Visual Studio 2010 и находятся в пространстве имен System.Windows.Forms.

Программа Spider v2.0 имеет свою собственную индексную базу, в которую помещаются все найденные системой ресурсы. При первом запуске она может быть использована как метапоисковая система с возможностью получения пространства информационных ресурсов от поисковых систем Google, Яндекс и от интеллектуальных поисковых агентов, добавленных в собственную базу либо сформированных автоматически на основе OWL - описания. После каждого поиска у пользователя есть возможность улучшить результат за счет алгоритма обучения поисковых агентов.

В главе подробно описаны результаты проводимых экспериментов.

1. Эксперимент был направлен на применение инструментария для разработки поисковых систем, созданного в ходе диссертационной работы. Результат показал, что с применением инструментария, на основе которого построено приложение Spider v2.0 трудоемкость работы понижается примерно на порядок. После получения данных от поисковых агентов в ходе эксперимента был выполнен процесс их автоматического обучения, что позволило сократить время работы программы без изменения результативности поиска (рис. 5).

2. Эксперимент был направлен на использование непосредственно самой программы Spider v2.0. с определенным набором поисковых агентов и тщательно описанной моделью пользователей для всех экспериментаторов. Сравнение производилось с поисковыми системами Google и Яндекс, в результате чего было установлено, что программа Spider v2.0 для одной группы экспериментаторов показала хорошее значение pertinentности для всех запросов в связи с полной совместимостью контекста запроса и модели пользователя. У двух других групп экспериментаторов значение pertinentности было выше при применении Spider v2.0 в среднем на 30 %. Значение этого показателя связано с тем, что pertinentность считалась только для первых десяти источников в выдаче. Показатель релевантности уменьшается с увеличением числа ключевых слов в запросе. На основе этого был сделан вывод о том, что программный инструментарий Spider v2.0, выполненный в рамках диссертации, приводит в большинстве случаев к повышению релевантности и pertinentности поиска в сравнении с существующими поисковыми машинами, но почти всегда проигрывает им по скорости

выполнения поиска, которая, кроме того, зависит от количества слов самого запроса.

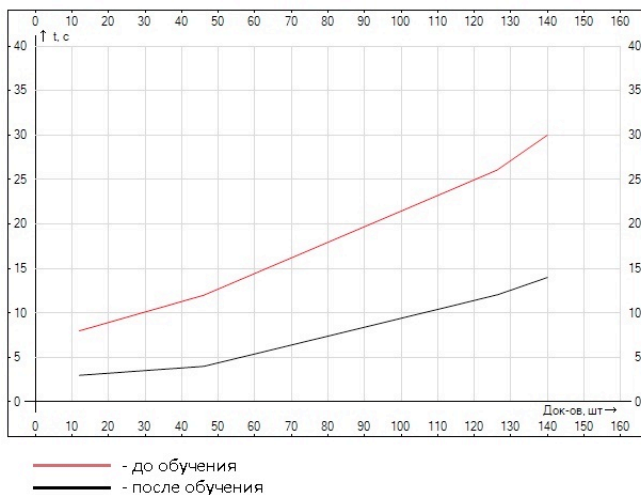


Рис. 5. График зависимости времени работы системы от количества найденных документов до и после обучения агентов

**В заключении** диссертационной работы сформулированы основные научные результаты, полученные в рамках решения поставленной научно-технической проблемы, связанной с развитием систем информационного поиска в ГС на основе применения OWL-ориентированных интеллектуальных поисковых агентов.

#### Список публикаций по теме диссертации

1. Каширин И.Ю., Минашкин С.А. Алгебра для описания композиционных свойств поисковых агентов в глобальных информационных сетях// Математическое и программное обеспечение вычислительных систем: межвуз. сб. науч. тр./под ред. А. Н. Пылькина. -Рязань: РГРТУ, 2012. – С. 33-36.
2. Каширин И.Ю., Минашкин С.А. Полиморфическое сопоставление онтологий с частично унифицированными тезаурусами //Проблемы передачи и обработки информации в сетях и системах телекоммуникаций: материалы 17-й международной науч.-техн. конф. - Рязань: РГРТУ, 2012. – С. 34-36.
3. Каширин И.Ю., Минашкин С.А. Онтологии для представления знаний в интерактивных сервисах информационных сетей // Вестник Рязанского государственного радиотехнического университета № 1 (выпуск 39).- 2012. - Часть 2. – С. 72-76.

4. Каширин И.Ю., Минашкин С. А. Обучение поисковой системы на основе решения уравнений алгебры поисковых агентов// Новые информационные технологии в научных исследованиях: сб. науч. тр. по материалам XVII Всероссийской науч.-техн. конф. студентов, молодых ученых и специалистов. - Рязань: РГРТУ, 2012. – С. 94-97.
5. Каширин И.Ю., Минашкин С. А. Тенденции развития современных поисковых систем// Информационные технологии в учебном процессе и научных исследованиях: межвуз. сб. науч. тр. /под ред. А. Н. Пылькина. - Рязань: РГРТУ, 2012. – С. 42-45.
6. Минашкин С. А. Semantic Web как современное средство описания Интернет -ресурсов // Математическое и программное обеспечение вычислительных систем: межвуз. сб. науч. тр. /под ред. А. Н. Пылькина. - Рязань: РГРТУ, 2012. – С. 121-130.
7. Минашкин С.А. Программный инструментарий для разработки поискового агента // Программное обеспечение вычислительных и информационных систем (Приложение к межвузовскому сборнику научных трудов «Математическое и программное обеспечение вычислительных систем»): тезисы докладов 56-й студенческой научно-технической конференции; секция вычислительной и прикладной математики / под ред. А.Н. Пылькина. – Рязань: РГРТУ, 2009. - 45 с.
8. Минашкин С.А. Проектирование адаптивной архитектуры системы интеллектуального поиска // Вопросы образования и науки в XXI веке: сб. науч. тр. по материалам международной научно-практической конференции 29 апреля 2013 г.: в 11 частях. Часть 7; М-во обр. и науки РФ. - Тамбов: Изд-во ТРОО "Бизнес-Наука-Общество", 2013. – С. 114-116.
9. Минашкин С.А. Алгоритм сопоставления концептуальных понятий онтологий // Актуальные научные вопросы и современные образовательные технологии: сб. науч. тр. по материалам международной научно-практической конференции 28 июня 2013 г.: в 7 частях. Часть 3; М-во обр. и науки РФ. Тамбов: Изд-во ТРОО "Бизнес-Наука-Общество", 2013. – С. 89-91.
10. Минашкин С. А. Анализ модели пользователя и определение его потребности при поиске в Интернете // Естественные и технические науки № 5 (67).- 2013. - С. 296-298.
11. Минашкин С. А. Определение отношений между понятиями для программной реализации словаря синонимов // Наука и образование в XXI веке: сб. науч. тр. по материалам международной научно-практической конференции 30 сентября 2013 г.: в 34 частях. Часть 32; М-во обр. и науки РФ. - Тамбов: Изд-во ТРОО "Бизнес-Наука-Общество", 2013. – С. 109-110.

**Минашкин Сергей Александрович**

**Математическое и программное обеспечение интеллектуальных  
поисковых систем на основе использования мультиагентной  
архитектуры**

Автореферат  
диссертации на соискание ученой степени  
кандидата технических наук

Подписано в печать 15.11.13. Формат бумаги 60x84/16.  
Бумага офсетная. Печать трафаретная. Усл. печ. л. 1.0.  
Тираж 100 экз.

Рязанский государственный радиотехнический университет.  
390005, г. Рязань, ул. Гагарина, 59/1.  
Редакционно-издательский центр РГРТУ