

И.А. ЩЕРБАТОВ,  
И.О. БЕЛЯЕВ

**Многоагентная поисковая система:  
применение Фибоначчиевых куч**

УДК 004.942

ФГБОУ ВПО  
«Астраханский  
государственный  
технический  
университет»,  
г. Астрахань

*В работе показана архитектура многоагентной системы, реализованной для решения задачи информационного поиска. Реализованы три класса типовых агентов, а именно «агент анализатор», «агент хранитель», «агент обработчик индексов», используемые в информационно-поисковой системе, показана обобщенная схема взаимодействия указанных агентов в составе системы. Предложено применение фибоначчиевых куч для решения задачи информационного поиска. Показан расчетный пример, подтверждающий эффективность предложенного подхода.*

### **Введение**

Проблема информационного поиска заключается в обеспечении поиска неструктурированной документальной информации, отвечающей требованиям некоторого запроса, формулируемого зачастую в свободной форме. Для обеспечения процедуры информационного поиска существует большое число информационно-поисковых систем, применяемых, например, в научной деятельности [1], поиске web-ресурсов [2] и пр.

Многоагентные системы представляют эффективную модель для решения слабоформализуемых задач с помощью распределенных вычислений [3]. Покажем возможности применения многоагентного подхода к построению системы информационного поиска робототехнической системы, применяемой при

формировании ответов на голосовые поисковые запросы пользователя [4].

### Терминологический базис

Ключевым понятием в любой многоагентной системе является агент. Агент должен обладать следующими основными характеристиками [5]:

- *коммуникабельность*, которая выражается в общении с другими агентами по определенному протоколу;

- *активность*, это означает, что агент должен целенаправленно получать знания о других агентах и о свойствах постоянно изменяющейся среды;

- *автономность* - агент, на основе своей базы знаний о внешних объектах (других агентах, внешней среды, связями между агентами и средой), а также знаний о предметной области решаемой задачи, должен с помощью правил вывода получать логические выводы независимо от других агентов;

- *адаптация и обучаемость*.

Каждый агент решает небольшую строго определенную задачу.

*Многоагентная архитектура* представляет собой совокупность *автономных* агентов, решающих различные *подзадачи*, соединенными между собой *каналами связи*. Одним из самых распространенных механизмов общения между агентами является механизм «*Доски объявлений*» [6].

В рамках разработки многоагентной поисковой системы введем следующие понятия [7]:

- *лексема* – слово, словосочетание, формула, устойчивая последовательность символов и пр;

- *запрос* – набор лексем (терминов), по которым происходит поиск по коллекции документов;

- *документ* – это последовательность лексем, связанных между собой семантической связью;

- *семантический вес лексемы* – это условная степень важности лексемы, входящей в документ;

- *термин* - это лексема в документе, имеющая семантический вес больше порогового. Пороговое значение выбирается в зависимости от обрабатываемой информации;

- *индекс* – это компактное представление связей между терминами и документами;
- *релевантность (отклика)* – числовая величина, характеризующая соответствие документа введенному запросу.

### Постановка задачи

Каждому документу  $d_i$  ставится в соответствие набор терминов  $T = (t_1, t_2, \dots, t_n)$ , в то время как с каждым термином  $t_j$  ассоциируется набор документов  $D = (d_1, d_2, \dots, d_m)$ , в которых он встречается. Семантический вес термина выражается в виде дробного числа в заданном диапазоне.

Поэтому задача заключается в определении документа (группы документов) с *наибольшим семантическим весом по ключевому термину*.

### Многоагентная поисковая система

Разработанная многоагентная система содержит в себе некоторое количество специфицированных агентов, обладающих определенным функциональным назначением. Число агентов зависит от размерности решаемой задачи, а сами агенты могут генерироваться (уничтожаться) в процессе функционирования информационно-поисковой системы. Для решения поставленной задачи информационного поиска синтезировано три класса агентов:

- *анализатор* - в его задачу входит семантический анализ документов и получение индекса для компактного хранения знаний о документе;

- *хранитель знаний* - в его задачу входит добавление новых знаний, полученных от агента-анализатора и быстрый поиск по индексам в базе знаний;

- *обработчик индексов* - в его задачу входит получение индексов от хранителя знаний по ключу.

Фрагмент многоагентной системы информационного поиска, иллюстрирующий схему взаимодействия агентов базовых классов представлен на рис. 1. При этом абсолютно не важно, как общаются между собой эти агенты, либо через прямые каналы связи, либо через «Доску объявлений».

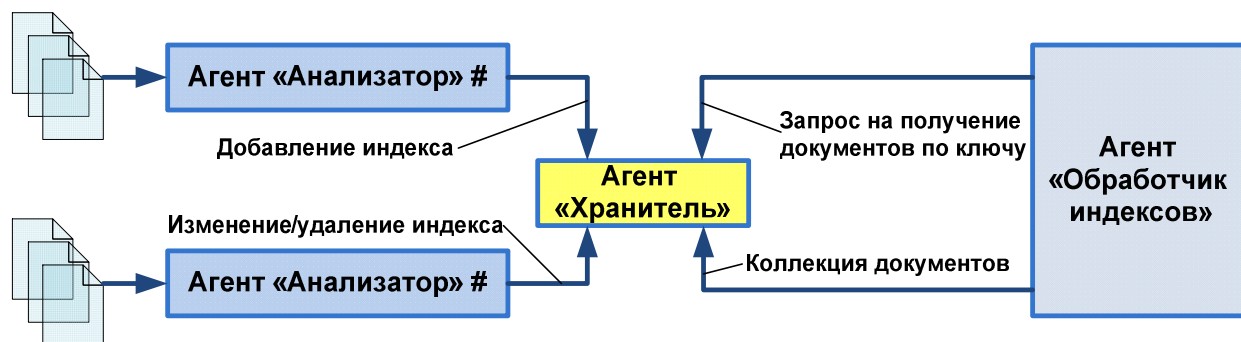


Рис. 1. Схема взаимодействия базовых агентов.

В рамках агента «хранителя знаний» необходимо обеспечить эффективные механизмы следующих операций для коллекции документов по конкретному ключевому термину:

- добавление нового документа в коллекцию с конкретным семантическим весом;
- определение документа с наибольшим семантическим весом;
- определение первых  $k$  документов с наибольшим семантическим весом;
- удаление документа из коллекции с любым семантическим весом термина;
- изменение семантического веса ключевого термина в документе;
- объединение двух и более коллекций документов, относящихся к ключевому термину, принадлежащих разным агентам «хранителям знаний».

### Использование Фибоначчиевых куч в информационном поиске

Для решения поставленной задачи предлагается использовать структуру данных – *фибоначчиева куча*. В общем случае этот вид кучи работает с однородными объектами, которые имеют отношение порядка на множестве, другими словами с теми объектами, которые можно сравнить. Данный вид структуры данных появился в 1984г. и был впервые описан в работе [8]. Фибоначчиева куча представляет собой набор деревьев.

В рамках поставленной задачи в качестве однородных объектов выступают документы, а в качестве параметра, по которому будут сравниваться документы, семантический вес термина. В табл. 1 представлена асимптотическая сложность операций, требуемых для решения поставленной задачи.

**Сложность операций для фибоначчиевой кучи из N элементов**

№	Название операции	Асимптотическая сложность
1	Добавление нового элемента	$O(1)$
2	Определение максимального элемента	$O(1)$
3	Определение k максимальных элементов	$O(k \cdot \log(N))$
4	Удаление элемента	$O(\log(N))$
5	Изменение приоритета	$O(\log(N))$
6	Объединение двух куч	$O(1)$

В соответствии с данными, представленными в таблице 1, асимптотическая сложность всех операций (включая операцию объединения двух фибоначчиевых куч) не связанных с удалением и изменением приоритета выполняется за константное время. Поэтому рассматриваемая структура данных является эффективным инструментом, использование которого в системе информационного поиска представляется целесообразным.

**Эксперимент и анализ результатов**

Эксперимент, подтверждающий эффективность полученных в работе результатов, проводился на рабочей станции с процессором Intel Core2Duo T5250 1.5 ГГц, RAM до 2ГБ на платформе Windows 7. В ходе эксперименте для решения поставленной задачи сгенерированы 2 агента «анализатора», 1 агент «хранитель знаний» и 8 агентов «обработчиков индексов». В результате работы многоагентной поисковой системы агенты обрабатывали заданное количество документов, получая индексы по 1024 терминам. Каждый документ содержит не более 200 терминов.

Общий процент всех операций в рамках агента «хранителя знаний» был максимально приближен к реальным условиям и в среднем был распределен так:

- 17.04 % - добавление новых документов по заданному термину;
- 53.74 % - получение документа с наибольшим семантическим весом по заданному термину;
- 9.68 % - получение первых k ( $k \leq 10$ ) документов с наибольшим семантическим весом заданного термина;

- 12.81 % - удаление документов, связанных с заданным термином;
- 6.73% - изменение семантического веса в документе по заданному термину.

На рис. 2 представлена зависимость времени, затраченного на обработку, от заданного количества документов на уровне агента «хранителя знаний».



Рис. 2. Зависимость времени обработки от количества документов

В ходе эксперимента использовались высокоскоростные помехоустойчивые каналы связи. Поэтому временем на обмен сообщениями в данном случае можно пренебречь. Как видно из графика, представленного на рис. 2, зависимость практически линейная. Это в первую очередь связано с тем, что только одна из четырех обрабатываемых операций имеет логарифмическую сложность.

### Заключение

Предложенная в работе архитектура многоагентной системы, позволяющая генерировать требуемое число агентов трех базовых классов, используемая для решения задачи информационного поиска с применением Фибоначчиевых куч, обеспечивает ускорение поиска документов с заданным входением терминов. В качестве дальнейшего направления исследования будет реализовано применение специальных алгоритмов, обеспечивающих существенное повышение быстродействия поиска.

## Литература

1. Шохин Ю.И. Проблемы поиска информации / Ю.И. Шохин, А.М. Федотов, В.Б. Барахнин. Новосибирск: Наука, 2010. – 220 с.
2. Кузнецов М.А., Нгуен Тан Там Математические модели информационного поиска web-ресурсов Прикаспийский журнал: управление и высокие технологии. - 2013. - № 2 (22). - С. 25-30.
3. Аграновский А.В., Болотин М.А., Букатов А.А. Организация сетевых вычислений на основе многоагентных систем // Известия Южного федерального университета. Технические науки. -2000. - № 2. Том 16. - С. 26-29.
4. Проталинский И.О., Кирилин С.А., Елизаров Д.В. Универсальная мобильная платформа для роботов, обслуживающих социальную и бытовую сферу // Вестник Астраханского Государственного Технического Университета. Сер. Управление, вычислительная техника и информатика. 2011. Вып. 2. С.34-38.
5. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям/ Науки об искусственном. – М.: Эдиториал УРСС, 2002. – 352 с.
6. Хьюз К. и Хьюз Т.. Параллельное и распределенное программирование с использованием C++. – М.:Вильямс: 2004 – 672 с.
7. Маннинг К.Д., Рагхаван П., Шютце Х.. Введение в информационный поиск. – М.: Вильямс: 2011 – 528с.
8. Fredman, M. L.; Tarjan (1987). Fibonacci heaps and their uses in improved network optimization algorithms. CACM 34 (3): 596–615.

ЩЕРБАТОВ И.А.

ТЕЛ. 8-917-187-27-93, (8512) 614-231,

SHERBATOV2004@MAIL.RU

БЕЛЯЕВ И.О.

ТЕЛ. (8512) 614-231,

UNDERFELIXABOVE@YANDEX.RU