

УДК 004.421.4, 004.421.6

Особенности поиска информации в метапоисковой ГРИД-системе «Программирование для ОС «Android»

И.А. Коломойцева, Т.В. Завадская
Донецкий национальный технический университет
bolatiger@gmail.com

Коломойцева И.А., Завадская Т.В. Особенности поиска информации в метапоисковой ГРИД-системе «Программирование для ОС Android». В данной работе рассматривается решение задачи метапоиска с помощью ГРИД-системы для предметной области «Программирование для ОС Android». Приведена общая схема ГРИД-системы для решения этой задачи. Предложена структура тезауруса для представления естественно-языковой информации, используемой для синтеза запросов метапоисковой системы, и формирования результата её работы. Описана реализация отправки запросов и получения результатов для поисковых машин Google, Yandex, Yahoo, Bing.

Ключевые слова: метапоиск, ГРИД-система, тезаурус, Android, объекты, семантические отношения, Google, Yandex, Yahoo, Bing.

Введение

Наше время можно назвать эпохой мобильных устройств. Они вошли в нашу жизнь и стали её неотъемлемой частью. Большинство мобильных устройств работают под управлением операционной системы Android. По данным сайта <http://4pda.ru> их доля на рынке превысила 86%.

Растёт также количество Android-разработчиков. Основную информацию о программировании под Android они берут из Интернета. Учитывая разнообразие мобильных устройств (смартфоны, планшеты, часы, очки и т.п.), их широкие аппаратные возможности (наличие GPS, акселерометров, NFC и т.п.), частоту выхода версий ОС Android (в среднем раз в два-три месяца), разработчикам приложений приходится очень часто обращаться с запросами к всемирной паутине. С другой стороны, существует огромное количество Интернет-ресурсов, содержащих сведения о программировании для ОС Android. Информация на них часто повторяется и, чтобы найти нужные сведения иногда приходится тратить много времени, просматривая ресурсы, перефразируя запросы, обращаясь к разным поисковым машинам.

Облегчить Android-разработчикам поиск сведений об особенностях программирования под Android может метапоисковая система.

Метапоисковая система (метапоисковая машина) — это поисковая система, которая в отличие от классических поисковых машин не имеет собственной базы данных и собственного

поискового индекса, а формирует поисковую выдачу за счет смешивания и переранжирования результатов поиска других поисковых систем [2].

Среди разработанных метапоисковых систем следует отметить Web Scout (ищет новости, конференции, аукционы), 1 SECOND, search.da.ru, exactus [1].

Задача метапоиска ресурсозатратная, обладает естественным распараллеливанием. Кроме этого, при реализации автоматических запросов к стандартным поисковым машинам возникают проблемы, связанные с ограничением, установленным этими машинами. Они ограничивают количество автоматических запросов, которые можно выполнить в единицу времени с одного IP-адреса.

Снять это ограничение и увеличить скорость работы метапоисковой системы можно помощью ГРИД-системы.

Направление развития ГРИД-технологии происходит по трём направлениям [2]:

- 1) вычислительные ГРИД;
- 2) семантические ГРИД;
- 3) ГРИД для интенсивной обработки данных.

ГРИД первого направления предназначены для решения вычислительных задач, например с использованием численных методов (решение уравнение математической физики) [2].

Семантические ГРИД предназначены для оперирования данными из различных баз данных.

ГРИД третьего направления предназначены для обработки огромных объёмов данных несложными программами, которые можно

реализовать на персональном компьютере. Сложность этого направления – доставка данных для обработки и пересылка результатов [2].

ГРИД-системы используются для решения задач прогноза погоды, обработка космических, астрофизических, геофизических данных и для БАК, индексирование новостных БД, в физике высоких энергий, биомедицинских науках, вычислительной химии, ядерного синтеза, финансово-экономических исследований [2, 3, 4, 5].

Задача метапоиска информации о программировании для ОС Android требует обработки большого объёма информации. Поэтому её решение с помощью ГРИД-системы является актуальной задачей.

Целью данной статьи является описание структурной схемы ГРИД-системы, решающей задачу метапоиска информации о программировании для ОС Android, структуры текстовой информации, которой она оперирует, а также описание использования поисковых машин Google, Yandex, Yahoo и Bing в рассматриваемой

системе.

Общая постановка задачи

ГРИД-система для решения задачи метапоиска делится на следующие подсистемы (рисунок 1):

- подсистема интерфейса пользователя, которая обеспечивает ввод запроса от пользователя и вывод результатов поиска;
- подсистема генерации запросов, которая по введенному пользователем генерирует запросы, семантически идентичные исходному;
- подсистема работы брокера (планировщика), которая распределяет запросы между узлами и принимает результат работы;
- подсистема работы узлов Грид-системы, выполняющих поиск;
- подсистема отбора релевантных результатов, которая получает их от брокера и после обработки передаёт подсистеме интерфейса пользователя.

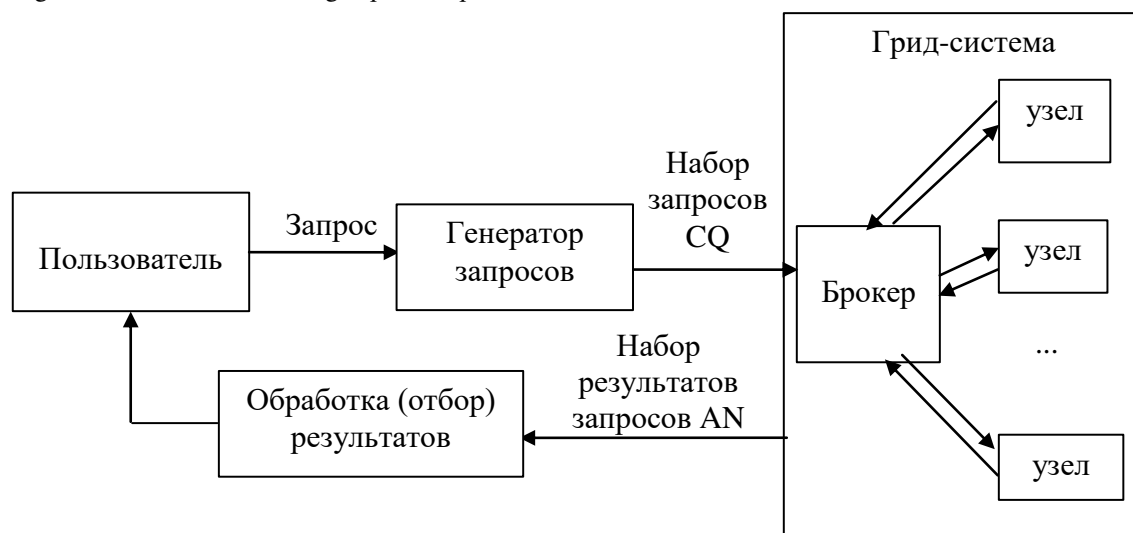


Рисунок 1 – Общая схема ГРИД-системы, решающей задачу метапоиска

Базой для генерации запросов для подсистемы генерации запросов является тезаурус. Тезаурус обычно привязан к некоторой предметной области. Например, в работе [6] предложена объектная модель семантического анализа естественно-языкового текста, используемая для тезауруса запросов в таких предметных областях, как “заболевания” [6] и “лекарства” [7]. Описание тезауруса для предметной области «Программирования для ОС Android» приведено в этой статье дальше.

Узлы ГРИД-системы при решении задачи метапоиска могут выполнять две задачи:

- запускать стандартную поисковую

машину с конкретными запросами и получать результат;

- в БД текстов, которая хранится на узлах, осуществлять поиск по конкретному запросу и выдавать список документов, ему удовлетворяющих.

В [8, 9] предлагается структура подсистемы, которая на основе объектной модели из [6], выполняет анализ текстовой БД и отбирает нужные документы. Отобранные документы вместе с результатами работы стандартных поисковых машин, отправляются в подсистему отбора релевантных результатов.

Отбор результатов в подсистеме отбора релевантных результатов может осуществляться

на основе двух подходов:

- объектного;
- функционального.

В работах [6-9] описан метод отбора релевантных результатов на основе анализа выдач с помощью тезауруса, построенного посредством объектной модели.

Альтернативой является функциональный подход, представленный в работе [9]. В соответствии с этим подходом выдачи результатов запросов можно представить в виде функций, аргументы которых должны быть сопоставлены с образцами. Однако этот подход имеет более высокую вычислительную сложность.

Алгоритм работы брокера приведен в [10].

Объекты и семантические отношения в тезаурусе «Программирование для ОС Android»

Чтобы использовать естественный язык в качестве основы для построения языка представления знаний, в нем предлагается выделить несколько классов-элементов. Эти классы можно разделить на две категории: семантически значимые объекты предложения и семантические отношения. Объекты еще называют именами [11] и именованными сущностями [12]. Примеры объектов, представленных в ЕЯ-текстах с описанием элементов программирования для ОС Android,

приведены в таблице 1.

Часть из приведенных в таблице 1 объектов имеют сложную иерархическую структуру. К таким объектам можно отнести ВИДЖЕТЫ. Из таблицы 1, например, видно, что к ВИДЖЕТАМ относятся ИНДИКАТОРЫ, а ИНДИКАТОРЫ, в свою очередь, делятся на ProgressBar, RatingBar, SeekBar. Имеют свои разновидности СПИСКИ, ТЕКСТОВЫЕ_ПОЛЯ, КНОПКИ и другие ВИДЖЕТЫ.

Объекты связываются между собой с помощью семантических отношений. Выдвинута гипотеза, согласно которой множество отношений, в отличие от множеств объектов (имен), конечно [11]. В [11] выделено около 200 не сводимых к друг другу отношений. В [13] 200 отношений из [11] сведены к семнадцати.

Классы-объекты можно представить в виде древовидной структуры. Особенностью данного дерева является то, что в узлах дерева находятся названия классов, а листьями являются понятия данного класса, что позволяет достаточно четко их определять. Все листья, которые определены в данном классе, являются синонимами.

В естественно-языковых текстах из предметной области «Программирование для ОС Android» можно выделить следующие семантические связи: генеративную, результативную, инструментальную, каузальную, комитативную [13].

Таблица 1. Примеры объектов, представленных в ЕЯ-текстах с информацией о программировании для ОС Android

№ п/п	Название объекта	Примеры объектов
1	КОМПОНЕНТ	Деятельность (Activity), служба (Service), приёмник широковещательных намерений (Broadcast Receiver), контент-провайдер (Content Provider)
2	СТАТУС_ПРОЦЕССА	Активный процесс, фоновый процесс, пустой процесс
3	РАЗРЕШЕНИЕ	Permission-tree, permission-group
4	РЕСУРС	Изображение, звук
5	РАЗМЕТКА	Фрейм (FrameLayout), линейная (LinearLayout), табличная (TableLayout), относительная (RelativeLayout)
6	ВИДЖЕТ	Текстовые поля, списки, кнопки, индикаторы
7	ИНДИКАТОР	ProgressBar, RatingBar, SeekBar
8	УВЕДОМЛЕНИЕ	Вплывающее уведомление, уведомление в строке состояния
9	ДИАЛОГОВОЕ_ОКНО	AlertDialog, DatePickerDialog, TimePickerDialog, ProgressDialog
10	МЕНЮ	Меню опций, контекстное меню, подменю
11	ДАТЧИК	SensorManager
12	ПЕРЕДАЧА ДАННЫХ	Bluetooth, Wi-Fi, NFC
13	ТЕЛЕФОНИЯ	Звонок, SMS
14	ДАННЫЕ	SQLite, геолокационные данные

Генеративная связь имеет место, когда один компонент обозначает лицо или предмет, принадлежащий некоторой совокупности, категории, обозначаемой вторым компонентом.

Результативная присутствует в тех предложениях, где один компонент выражает следствие действия второго.

Инструментальная означает, что один компонент обозначает орудие действия, обозначаемого другим компонентом.

Каузальная имеет место, когда один

компонент обозначает причину появления другого компонента спустя какое-то время.

Комитативная встречается в тех предложениях, когда один компонент обозначает сопровождающее другой компонент действие, сопутствующий предмет, сопровождающее лицо.

Примеры объектов естественно-языковых текстов из предметной области «Программирование для ОС Android», связываемых семантическими отношениями, представлены в таблице 2.

Таблица 2. Семантические отношения и связываемые ими объекты

Семантическая связь	Связываемые объекты
Результативная	СОБЫТИЕ → ДЕЙСТВИЕ
Инструментальная	РАЗРЕШЕНИЕ → ДЕЯТЕЛЬНОСТЬ
Каузальная	УВЕДОМЛЕНИЕ → ДЕЯТЕЛЬНОСТЬ
Комитативная	МЕНЮ → ПОДМЕНЮ

Описание тезауруса предметной области «Программирование для ОС Android»

Знания о предметной области «Программирование для ОС Android» необходимо представить в виде, пригодной для автоматической обработки. Онтологии являются именно такой формой представления знаний.

Онтология – соглашение об общем использовании понятий, которое содержит средства представления предметных знаний и договоренности о методах соображений. Она может рассматриваться как определенное описание взгляда на мир в конкретной сфере интересов, который состоит из набора терминов и правил использования этих терминов, которые ограничивают их значение в рамках конкретной предметной области [11].

Согласно «Современному словарю иностранных слов» тезаурус определяется как полный систематизированный набор данных о какой-либо области знаний, позволяющий человеку или вычислительной машине в ней ориентироваться. Таким образом, тезаурус можно рассматривать как частный случай онтологии.

Существует множество стандартов на формат представления тезаурусов. Основными документами, регламентирующими формат представления тезауруса, являются стандарты ISO 2788-1986 для описания одноязычных тезаурусов и ISO 5964-1985. При разработке тезауруса предметной области «Программирование для ОС Android» будем опираться на первый из перечисленных стандартов.

Стандарт ISO 2788-1986 определяет тезаурус как набор терминов, связанных между собой соответствующими отношениями.

Каждый термин характеризуется комментарием и ссылкой на понятие верхнего уровня.

Основные понятия, представленные в тезаурусе «Программирование для ОС Android»: компонент, статус_процесса, разрешение, ресурс, разметка, виджет, уведомление, диалоговое_окно, меню, датчик, передача_данных, телефония, данные.

Связи, которые представлены в тезаурусе «Программирование для ОС Android»:

– USE – связь с очень близким по смыслу синонимом (термины А и В, представляющие некоторое понятие, обозначают практически одно и то же; например, у одного виджета может быть несколько названий, как минимум английское и русское);

– UF – связь, обратная USE;

– BT – связь термина с понятием высокого уровня (например, термин «текстовое поле» есть уточнение понятия «виджет»);

– NT – связь, обратная BT;

– Res – результативная;

– Ins – инструментальная;

– Caus – каузальная;

– Com – комитативная.

Программно тезаурус представлен таблицами termin, comment и links.

Таблица termin содержит поля, в которых присутствует информация о названии понятия, ссылка на комментарий, ссылка на информацию о связях, идентификатор уровня.

Таблица comment содержит словесное представление комментария.

Таблица links содержит информацию о типе связи, идентификаторы связываемых понятий.

Разработанный тезаурус можно

использовать для генерации запросов и формирования окончательного результата для вывода пользователю, в метапоисковой системе для предметной области «Программирование для ОС Android».

Использование существующих поисковых систем при решении задачи метапоиска

При решении задачи метапоиска для собственно поиска информации в сети Интернет используются стандартные поисковые машины. В метапоисковой системе «Программирование для ОС Android» с качестве таких поисковых машин используются Google, Yandex, Bing и Yahoo. Эти поисковые машины популярны в мире и предоставляют программные средства для организации автоматического выполнения запросов и получения их результатов.

Метапоисковой системе «Программирование для ОС Android» на вход

подаётся запрос от пользователя. Используя тезаурус, описанный выше, генерируется N похожих по смыслу запросов. Количество запросов N зависит от числа слов в запросе, количества синонимов в тезаурусе для каждого из слов и типа семантической связи между словами запроса. Каждый из этих N запросов запускается на всех поисковых машинах: Google, Yandex, Bing и Yahoo. Эти 4*N запросов брокер ГРИД-системы равномерно распределяет между доступными в текущий момент узлами системы. Подробно алгоритм распределения запросов между узлами ГРИД-системы описан в [10].

Опишем как организован программной генерации запроса и получения результата в каждой из поисковых машин.

Google использует методы GET или POST при отправлении запроса, результат формируется с помощью технологии REST. Функция организации поиска для поисковой машины Google приведена на рисунке 2.

```
public void Search()
{
    int count = 0;
    while(count < 32)
    {
        HttpWebRequest Request = BuildRequest(Query, count);
        HttpWebResponse Response = (HttpWebResponse)Request.GetResponse();
        StreamReader sr = new StreamReader(Response.GetResponseStream());
        JObject responseObj = JObject.Parse(sr.ReadToEnd());
        JToken responseData = (JToken)responseObj["responseData"];
        JArray resultArray = (JArray)responseData["results"];
        for (int i = 0; i < resultArray.Count; i++)
        {
            string cnt = (string)responseData["cursor"]["estimatedResultCount"];
            Uri url = new Uri((string)resultArray[i]["unescapedUrl"]);
            Results.Add(new
                SearchResult(BoldText((string)resultArray[i]["title"]),
                    url.ToString(), BoldText((string)resultArray[i]["content"]), (string)resultArray[i]["visibleUrl"], "Google", Convert.ToInt32(cnt), Rating));
            count += 8;
        }
        Results.RemoveRange(30, 2);
        isSearching = false;
    }
}
```

Рисунок 2 – Функция поиска Google

```
<?xml version="1.0" encoding="windows-1251"?>
<request>
  <query>yandex</query>
  <groupings>
    <groupby attr="d" mode="deep" groups-on-page="10" docs-in-group="1"
  />
  </groupings>
</request>
```

Рисунок 3 – Пример простого запроса

Yandex использует технологию Яндекс.XML. Пример простого запроса приведен на рисунке 3.

Yahoo использует технологии REST и SOAP для составления запросов и получения результатов. Разработчиками поисковой машины

предоставлена библиотека, позволяющая реализовать поиск в Интернете, изображений, видео и другой полезной информации практически на любом языке программирования.

Функция организации поиска для поисковой машины Yahoo приведена на рисунке 4.

```
public void Search(string Query)
{
    YahooSearch = new YahooSearchService();
    Yahoo.API.WebSearchResponse.ResultSet resultSet =
    YahooSearch.WebSearch(AppId, Query, "all", 30, 1, "all", false, false, "ru");
    foreach (Yahoo.API.WebSearchResponse.ResultType result in
    resultSet.Result)
    {
        Results.Add(new
        SearchResult(result.Title, result.Url, result.Summary));
    }
}
```

Рисунок 4 – Функция поиска для поисковой машины Yahoo

```
private HttpWebRequest BuildRequest(string Query)
{
    string requestString = "http://api.search.live.net/xml.aspx?"
    // Common request fields (required)
    + "AppId=" + AppId
    + "&Query=" + Query
    + "&Sources=Web"
    // Common request fields (optional)
    + "&Version=2.0"
    + "&Market=ru-ru"
    + "&Adult=Moderate"
    + "&Options=EnableHighlighting"
    + "&Web.Count=30"
    + "&Web.Offset=0"
    +
    "&Web.Options=DisableHostCollapsing+DisableQueryAlterations";
    HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(
    requestString);
    return request;
}
```

Рисунок 5 – Функция построения запроса для поисковой машины Bing

```
private List<SearchResult> GetResults(XmlNode root, XmlNamespaceManager nsmgr)
{
    List<SearchResult> SResults = new List<SearchResult>();
    SearchResult SResult = new SearchResult();
    // Add the Web SourceType namespace to the namespace manager.
    nsmgr.AddNamespace(
    "web",
    "http://schemas.microsoft.com/LiveSearch/2008/04/XML/web");
    XmlNode web = root.SelectSingleNode("./web:Web", nsmgr);
    XmlNodeList results = web.SelectNodes(
    "./web:Results/web:WebResult",
    nsmgr);
    foreach (XmlNode result in results)
    {
        SResult.Title = result.SelectSingleNode("./web:Title", nsmgr).InnerText;
        SResult.URL = result.SelectSingleNode("./web:Url", nsmgr).InnerText;
        SResult.Body = result.SelectSingleNode("./web:Description", nsmgr).InnerText;
        SResults.Add(new SearchResult(SResult.Title, SResult.URL, SResult.Body));
    }
    return SResults;
}
```

Рисунок 6 – Получение результатов поиска для поисковой машины Bing

Bing использует технологии JSON, XML и SOAP. Функция построения запроса для

поисковой машины Bing приведена на рисунке 5, а функция получения результата – на рисунке 6.

Результаты работы поисковых машин, полученные брокером от узлов ГРИД-системы, передаются подсистеме формирования ответа, которая формирует итоговый html-документ для предоставления пользователю.

Выводы

В данной работе рассмотрена организация ГРИД-системы, решающей задачу метапоиска для предметной области «Программирование для ОС Android». В частности выделены основные её структурные элементы.

Разработан тезаурус для предметной области «Программирование для ОС Android». Этот тезаурус будет использован для синтеза запросов метапоисковой системы и для формирования результата её работы, который будет предоставлен пользователю.

Литература

1. Пасько В.П. Энциклопедия ПК. Аппаратура. Программы. Интернет. – Киев, издательская группа БНУ; СПб: Питер, 2004. – 800 с.
2. Котляр В.В. Применение Грид-технологий для задач интенсивной обработки данных – Режим доступа: <http://litcey.ru/geografiya/5431/index.html>
3. Castellano M. Biomedical Text Mining Using a Grid Computing Approach / M. Castellano, G. Mastronardi, Decataldo G., Pisciotto L., Tarricone G., Cariello L., Bevilacqua V. – Режим доступа <http://cdn.intechopen.com/pdfs-wm/12928.pdf>.
4. Hughes B. Grid-based Indexing of a Newswire Corpus / B. Hughes, S. Venugopal, R. Buyya. – Режим доступа: <http://www.cloudbus.org/papers/nlp-newswire-grid.pdf>.
5. Li Q. The Future-Oriented Grid-Smart Grid / Q. Li, M. Zhou // Journal of computers, Vol 6. – №1. – 2011. – p.98-105.
6. Коломойцева И.А. Объектная модель семантического анализа естественно-языкового медицинского текста / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2007). Выпуск 8 (120). – Донецк: ДонНТУ, 2007. –

С. 141-150.

7. Коломойцева И.А. Компьютерный тезаурус для предметной области «ЛЕКАРСТВА» / И.А. Коломойцева // Моделирование и компьютерная графика: Материалы 4-й международной научно-технической конференции, г. Донецк, 5-8 октября 2011 г. — Донецк, ДонНТУ, Министерство науки и образования, молодёжи и спорта, 2011. – с. 161-165.
8. Коломойцева И.А. Объектная модель естественно-языкового медицинского текста на примере системы «ФармАналитик» / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2009). Выпуск 10 (153). – Донецк: ДонНТУ, 2009. – С. 303-306.
9. Коломойцева И.А. Функциональная модель медицинского естественно-языкового текста / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2008). Выпуск 9 (132). – Донецк: ДонНТУ, 2008. – С. 237-241.
10. Коломойцева И.А., Дацун Н.Н. Алгоритм работы брокера ГРИД-системы для решения задачи метапоиска // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2014). – Донецк: ГВУЗ «ДонНТУ», 2014. – Вып. 1 (19) – С. 92-96.
11. Поспелов Д. А. Логико-лингвистические модели в системах управления. М.: Энергоиздат, 1981. 232 с.
12. Хорошевский В.Ф. Оценка систем извлечения информации из текстов на естественном языке: кто виноват, что делать // Труды Десятой национальной конференции по искусственному интеллекту с международным участием (КИИ-2006). М.: Физматлит, 2006. Т. 2. С. 464-478.
13. Осипов Г.С. Приобретение знаний интеллектуальными системами: Основы теории и технологии. М.: Наука. Физматлит, 1997. 112 с.

Коломойцева И.А., Завадская Т.В. Особенности поиска информации в метапоисковой ГРИД-системе «Программирование для ОС Android». В данной работе рассматривается решение задачи метапоиска с помощью ГРИД-системы для предметной области «Программирование для ОС Android». Приведена общая схема ГРИД-системы для решения этой задачи. Эта схема включает подсистемы пользовательского интерфейса и генерации запросов, брокер, узлы и подсистему генерации результатов. Подсистема пользовательского интерфейса предназначена для взаимодействия с пользователем. Подсистема генерации запросов создаёт множество запросов на основе пользовательского запроса, используя описанный в статье тезаурус. Брокер распределяет запросы между узлами ГРИД-системы и получает от них результаты работы. Подсистема генерации результатов отбирает среди полученных данных работы ГРИД-системы

релевантные пользовательскому запросу ответы. Предложена структура тезауруса для представления естественно-языковой информации, используемой для синтеза запросов метапоисковой системы, и формирования результата её работы. Описана реализация отправки запросов и получения результатов для поисковых машин Google, Yandex, Yahoo, Bing.

Ключевые слова: метапоиск, ГРИД-система, тезаурус, Android, объекты, семантические отношения, Google, Yandex, Yahoo, Bing.

.Kolomoitseva I.A., Zavadskaja T.V. Information Search Features in metasearch GRID-system "Programming for Android". This article describes the solution metasearch tasks using GRID-system for domain "Programming for Android". The General scheme of the GRID to solve this problem was developed. This scheme includes the subsystem of user interface, the subsystem of generating of requests, the subsystem of the broker, the subsystem of nodes, the subsystem of selection of relevant results. The subsystem of user interface is designed for user interaction with the grid system. The subsystem of request generation creates a lot of queries by the user's request. The subsystem of the broker distributes requests between the nodes and takes the result of the work. The subsystem of nodes executes a search using the standard search engines. The subsystem selection of relevant results selects relevant results from a variety of responses that came from the broker. The structure of a thesaurus to represent natural language information was developed. This thesaurus used for the synthesis queries of metasearch system, and the formation of the result of work this system. This article describes the implementation of sending requests and receiving the results of the Google search engine, Yandex, Yahoo, Bing.

Keyword: metasearch, GRID-system, thesaurus, Android, objects, semantic relations Google, Yandex, Yahoo, Bing.

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко