

## **Опыт создания персональной поисковой библиографической системы, ориентированной на конкретную область научных или инженерных знаний**

**Филимонов И.А.**

*Московский авиационный институт (национальный исследовательский университет), МАИ, Волоколамское шоссе, 4, Москва, А-80, ГСП-3, 125993, Россия  
[e-mail:lja199@bk.ru](mailto:lja199@bk.ru), [e-mail: ilafilimonov@mai.education](mailto:ilafilimonov@mai.education)*

*Статья поступила 03.07.2020*

### **Аннотация**

В статье рассмотрен опыт создания и примеры применения персональной поисковой библиографической системы, работающей на основе банка научных публикаций AI Corpus, созданного в Институте искусственного интеллекта Пола Аллена. Получена система, дополняющая такие поисковики, как Google, и, благодаря использованию открытых компонентов, может служить платформой для развития поисковых и справочно-аналитических функций более высокого уровня. Предложен подход, позволяющий производить поиск информации в два этапа, на первом этапе - путём выборки из глобального хранилища, а затем, на втором этапе – путём проведения поиска внутри этой выборки с тонким учётом специальных интересов владельца системы. Объектом исследований являются системы информационного и аналитического обслуживания в области программной инженерии. Реализованная система может также использоваться для поиска, отбора и анализа документов в области аэрокосмической техники.

**Ключевые слова:** поисковая система, банк ресурсов, двухэтапный поиск, банк научных публикаций, интерфейс пользователя.

## Введение

В составе мирового парка, равно как и в таксономии информационных систем, видное место занимает широкая группа систем, основная потребительская ценность которых заключается в выполнении функций поиска и анализа найденной информации. Указанные функции в реальных системах могут сочетаться по-разному, а «титульные» наименования систем – информационно-справочные, информационно-поисковые, аналитические и тому подобные – не дают достаточно точного представления о степени и характере сочетания этих функций в системе.

Для уточнения проблематики данной статьи предлагается частная классификация информационных систем по характеру и роли выполняемого в них информационного поиска.

Информационная система рассматривается как среда, в которой определены:

1. Банк ресурсов – множество информационных ресурсов, для которого задана эффективная процедура перебора.
2. Оператор (механизм) поиска, на вход которого подаётся поисковый запрос. Без существенного ограничения общности, его можно считать строкой. Оператор реализует процедуру перебора банка информационных ресурсов и на каждом ресурсе определяет, удовлетворяет ли ресурс запросу. Все ресурсы,

удовлетворяющие запросу, образуют в совокупности результат обработки запроса (поиска).

Рассматриваются следующие уровни реализации такой среды.

Вырожденный поиск. Поисковый запрос является идентификатором (номером, или адресом) в процедуре перебора. Поиск не является ассоциативным, поскольку контент информационного ресурса не обязан быть связанным с поисковым признаком. Если механизм поиска таков, что время перебора не зависит от значения поискового запроса (то есть адреса в банке ресурсов), то среда может называться пространством адресов, памятью прямого доступа (direct access), или памятью случайного доступа (random access).

Простой ассоциативный поиск. Поисковый запрос регулярно входит в состав контента информационных ресурсов, например, является заголовком страницы. Каталоги товаров в web могут служить примерами таких систем. Механизм поиска в этом случае сравнивает строки запроса со строкой в тексте контента, находимой относительно простым способом.

Продвинутый ассоциативный поиск. По сути, это ассоциативный поиск произвольной сложности. Поисковый запрос трактуется как словосочетание (сочетание лексем). Предикат является функцией произвольной сложности, параметрами которой являются лексеммы из строки запроса. Как правило, предикат имеет возможность вызова соответствующих лексических анализаторов.

Ассоциативный поиск в терминах поисковых систем - это процесс поиска информации при помощи синтаксически корректных (и/или некорректных) запросов для нахождения ранжированного множества релевантных совпадений.

Общеизвестные глобальные системы поисковых услуг Google, eBay, Yahoo, Jandex, Rumbler обеспечивают продвинутый ассоциативный поиск. Глобальные системы, решая задачу равно быстрого выполнения самых разных поисковых запросов, развили сложные сочетания каталогов и индексов, требующих огромных пространств внешней памяти и процессорных мощностей. Но трудно предположить, что они далеко пойдут по линии адаптации к интересам (моделям) пользователя, которая может привести к столкновению с защитой персональных данных. Поэтому развитие технологий глубокого «релевантного» поиска скорее найдёт применение в другом классе систем – в «персональных» системах. Актуальность и возможность создания систем такого рода рассмотрены в [4], [5].

Необходимо также отметить, что названные выше глобальные поисковые системы выполняют некоторый анализ найденных ресурсов, подсчитывают их количество, выделяют ближайший контекст, в котором встретились лексемы, служившие поисковыми признаками, и выдают их пользователю – источнику поисковых запросов. Однако развитие аналитического и справочного аппарата опять же предполагает определённую ориентацию информационной системы на нужды конкретного пользователя.

## Требования к поисковой системе

В данной статье предлагается рассмотреть опыт создания персональной поисковой библиографической системы "ПоискУМ", ориентированной на интересы конкретного научного или инженерного работника, которая на данном этапе своего развития обеспечивает продвинутый ассоциативный поиск, но, будучи открытой, могла бы служить платформой для развития поисковых и соответствующих справочно-аналитических функций более высокого уровня. Информационными ресурсами, на которые нацелен поиск, считаются документы, в общем случае мультимедийные.

Библиографические поисковые системы такие, как [6], [7], [8] задают как фактический стандарт минимальный набор поисковых функций, которыми должна обладать любая современная поисковая система:

1. В части организации GUI. Пользователю предоставляется экранный интерфейс [1], [3] в виде системы окон.
2. В части ведения поиска. Поисковый запрос [2] вводится в виде строки произвольной структуры в специальное окно;
3. В части осуществления поиска.

3.1 Запрос обрабатывается как потенциально осмысленное высказывание, подвергаясь лексическому анализу с установлением основных грамматических форм [15] выделенных лексем [16] (с целью последующего осуществления словарного поиска). Слова состоят из символов латинского и кириллического алфавитов, цифр

и спецсимволов, принятых в DOS/UNIX, длина строки поискового запроса - не более 256 символов (а-я, А-Я, a-z, A-Z, 0-9);

#### 4. В части результата поиска.

4.1 Для каждого найденного документа составляется отчёт – «образ документа», содержащий, как минимум, сетевой адрес документа,

4.2 Образы документов доступны пользователю в специальном окне в оговоренной последовательности, сохраняются и могут быть экспортированы полностью или выборочно в виде текстовых файлов;

Применение поисковой системы в качестве персональной, ориентированной на профессиональные интересы конкретного пользователя, предъявляет к ней дополнительные требования в части пертинентности [5] и информативности результатов поиска, а также управляемости процессом выполнения запросов.

### **Принципы организации системы “ПоискУМ”**

Для повышения пертинентности поиска в системе “ПоискУМ” предложено использовать предварительную селекцию информационных материалов, происходящую при формировании банка данных AI Corpus [11], который, занимая 134 МиБ, содержит в настоящее время метаданные для приблизительно 40 000 статей по программной инженерии с 2000 по 2020 годы.

Система “ПоискУМ” проводит вторую, персонально ориентированную селекцию, выбирая метаданные из исходного банка, в данном случае из банка

данных AI Corpus, и создаёт 2-уровневый кэш, который состоит из кэша первого уровня, так называемого «большого кэша», и «сверхоперативного» кэша второго уровня. «Большой кэш» (далее – Кэш) состоит из банка паспортов документов, банка .pdf файлов документов и банка внутренних образов документов.

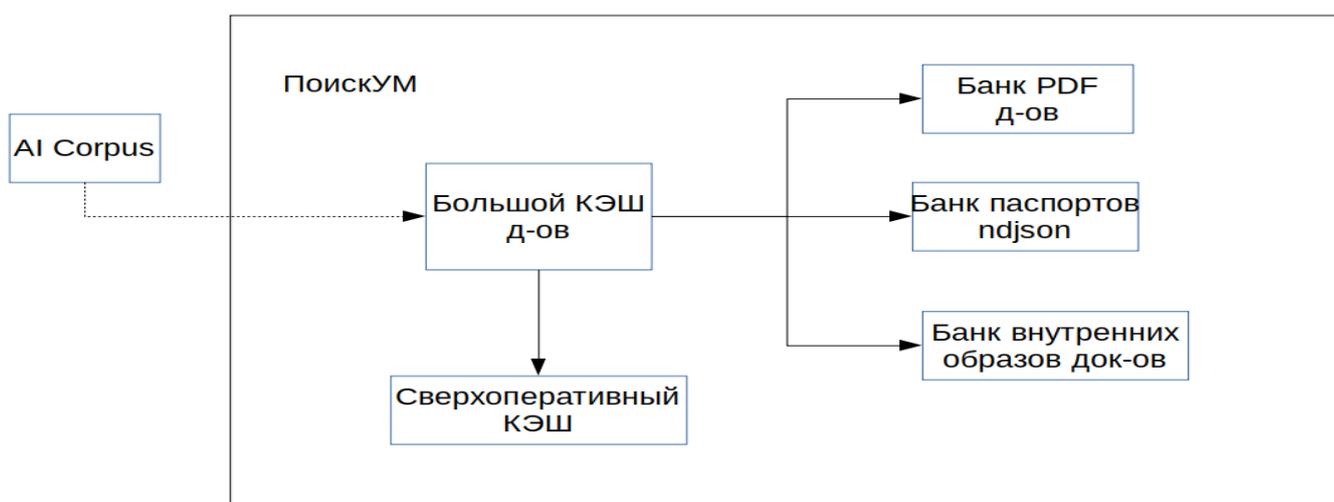


Рисунок 1 - схема устройства кэша в системе ПоискУМ

Паспорта документов организованы в "Большом кэше" на основе модели JSON с nd-синтаксисом [9], [10]. Такой синтаксис предполагает в качестве разделителей между строками паспорта использовать символ новой строки, так что строка будет иметь вид (рис. 1):

{имя\_документа, год, авторы, doi, цитаты, область, абстракт, pdfURL-ссылка, библиография}

В текущей версии системы «ПоискУМ» содержание Кэша получено выборкой из на дату 04/2020 из банка данных AI Corpus. 20 000 статей доступны для скачивания и входят в кэш в формате pdf.

Генерация и пополнение Кэша из корпусов метаданных осуществляется командами, выдаваемыми с помощью интерфейса командной строки:

1. `wget` ['URL' или `-i` 'файл, содержащий URLы'] - получение ndjson файла банка паспортов документов (в виде массива строк, разделённых символом новой строки) корпуса метаданных из глобальной сети; обеспечивает получение паспортов документов от адресованного web-сайта.
2. `grep` ['фильтр' > 'файл (куда сохраняется выборка)'] - поиск и выборка из базы строк; это команда UNIX, выбирает все паспорта документов, представляющих интерес для данного случая использования системы, они сохраняются в виде одного файла; из получившихся паспортов извлекаются командой `grep` все URL ссылки (поиском по фильтру) на .pdf тексты, которые доступны для свободного скачивания и являются рабочими ссылками, они заносятся в текстовый файл - файл списка URL-ссылок на .pdf тексты документов; в случае недоступности или нежелательности скачивания некоторых документов, документ заносится в чёрный список (`filter_blacklist`); полученный текстовый файл со списком URL-ссылок на .pdf тексты документов подаётся на загрузку, результатом выполнения этого шага является загрузка банка .pdf документов в директорию на локальном диске;
3. `split` [n] - разделение банка паспортов НТД на подбанки нужного размера;
4. `mount` - подключение каталога НТД в ядро для обеспечения доступности каталога для поиска; итоговый банк pdf включается (с помощью команды

mount) в каталог системы для последующего формирования внутреннего банка образов документов и поиска.

Сверхоперативный кэш реализован как журнал веб-сервера системы (текстовый файл), который производит регистрацию фактов обращения пользователя к системе с сохранением в файле всех его формулировок поисковых запросов (операций GET), а также технических характеристик его аппаратного и программного обеспечения.

Внутренний образ документа включает в себя следующие данные:

1. Информацию о типе документа (текстовый файл, изображение, видео, аудио);
2. Информацию о языке документа (en,ru,fr,it,es и т.д.);
3. Имя ресурса (сведения об имени файла или ресурса документа);
4. Заголовок документа (например, название научной статьи);
5. Информацию об авторах документа (персонах, работающих над документом);
6. Абстракт (аннотацию документа);
7. Библиографические списки документа (ссылки);
8. Токены и цепочки слов токенов документа (ключевые слова, синонимы, слова-производные, выделенные жирным или курсивом в тексте слова);

9. Наименования организаций, работающих над документом;
10. Контактные адреса электронной почты авторов документа;
11. Веб-сайты (ресурсы), связанные с текстом документа;
12. Контактные физические адреса (местоположения) организаций и персон, работающих над документом;
13. Сведения о годах и датах написания документа;
14. Секции документа (абстракт, основная часть, заключение, библиография);
15. Локальные ссылки на местоположения документа в файловой системе.

Для формирования банка образов документов в системе предусмотрены команды ETL (Extract, Transform, Load — дословно «извлечение, преобразование, загрузка»), выдаваемые с помощью интерфейса командной строки:

1. `etl-file 'filename'` или `'directory'` - построить образы документов, содержащихся в файле или, соответственно, директории на локальном диске;
2. `etl-file-monitoring 'directory'` или `'file'` - построить образы всех документов, содержащихся в файле директории, с учётом новых или изменённых файлов.
3. `etl-web 'uri'` - построить образ удалённого документа, хранящегося в глобальной сети (выход в глобальную сеть)

Работа команд ETL характеризуется следующей последовательностью действий:

1. Фильтрация обрабатываемого файла в банке по чёрному списку (filter\_blacklist);
2. Фильтрация обрабатываемого файла в банке (filter\_file\_not\_modified);
3. Извлечение текста (extension\_text);
4. Оптическое распознавание (если необходимо) OCR (Enhance\_ocr);
5. Генерация финального образа документов (Enhance\_rdf);
6. Формирование внутреннего банка образов документов для последующего его использования при поиске.

В целом система «ПоискУМ» реализована как клиент-серверное приложение, основное окно, показанное на Рис. 2, разворачивается любым web-браузером, обращающимся к серверу системы.

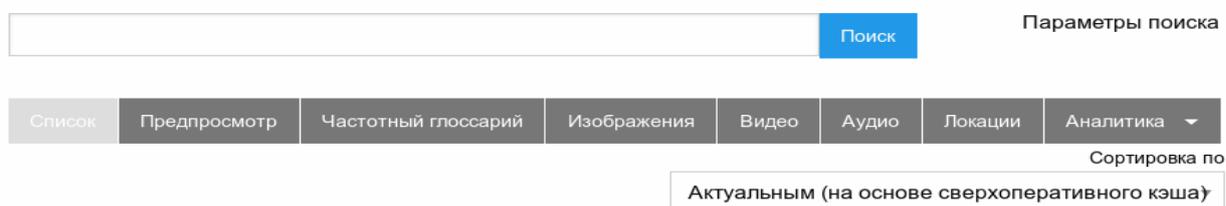


Рисунок 2 - Основное поисковое окно системы "ПоискУМ".

Окно содержит строку для ввода поискового запроса и линейку кнопок для управления системой.

## Ассоциативный поиск в системе “ПоискУМ”

Текущая работа пользователя – поиск и анализ библиографической информации – осуществляется в Кэше с применением компонентов открытой системы OpenSemanticSearch [12] и фреймворка Apache Tika [13], [14]. С помощью этих компонентов реализованы минимальные требования к информационно-поисковым системам, указанные в начале статьи, а также дополнительные требования по обеспечению информативности результатов поиска в виде реализации следующих возможностей:

1. Выдача результата запроса в виде списка изображений, аудио или видео материалов.
2. Распознавание языка документа и соответствия его стандартам MIME [17], [18].
3. Выдача частотных глоссариев [19] по категориям.
4. Проведение нечёткого поиска [20].
5. Выявление трендов появления документов по годам.
6. Получение «облаков тэгов», сопровождающих искомые лексемы.

Система "ПоискУМ" обеспечивает продвинутое возможности также по управлению процессом выполнения поисковых запросов:

1. Локализация пространства поиска (в кэше, в сети Интернет),
2. Возможность параллельного выполнения запросов,
3. Возможность остановить, прервать и возобновить обработку запроса,
4. Возможность возврата к запросу для его редактирования и повторения,

## 5. Возможность выбирать вариант опознавания документа при поиске.

Реализация этих возможностей актуальна в связи с тем, что сложная обработка запросов может потребовать больше времени, чем обработка запросов в таких системах, как Google, которая пользователю представляется мгновенной.

### Пример работы системы “ПоискУМ”

Пример поискового запроса и сравнение результатов поиска с поисковыми системами общего назначения приведён на Рис. 3 – 5. В качестве примера поискового запроса взят запрос "lifecycle". Выполнение ввода поискового запроса и поиска в системе "ПоискУМ":

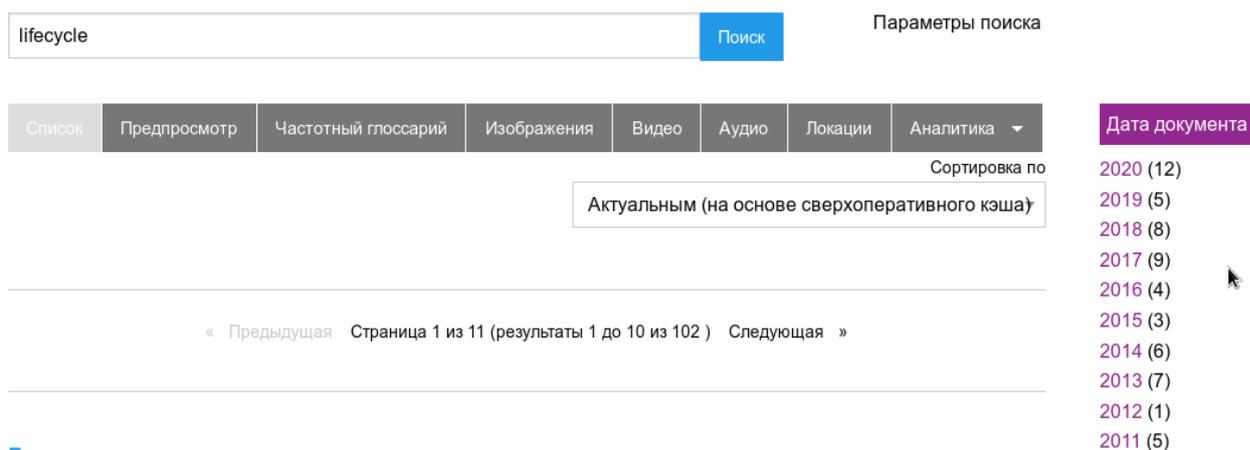


Рисунок 3 - Ввод поискового запроса в систему "ПоискУМ", нажатие на кнопку "Поиск" и показ начального вида результата после ввода запроса

Весь результат отображается в виде списка, вся выдача может быть просмотрена путём прокрутки экрана (рис. 4). На рис 4 сработала функция, вырабатывающая тезисы документа. В правой части выдачи размещаются персоналии с частотой упоминания по всей выдаче.

[TribitsLifecycleModel\\_eScience\\_2012.pdf](#)

2019-11-08T23:24:54Z

[TribitsLifecycleModel\\_eScience\\_2012.pdf](#)

- Обзор TriBITS **Жизненный цикл** Модель A Lean / Agile Software **Жизненный цикл** Модель программного обеспечения для вычислительной науки и техники, основанного на исследованиях Роско А. Барлетт Национальная лаборатория Ок-Риджа, почтовый ящик 2008 г. Ок-Ридж, TN 37831 Email
- начинается жизнь как набор требований к исследованиям, а затем превращается в надежную высококачественную возможность, которая является одновременно обычной и чрезвычайно сложной. Хотя неявный **жизненный цикл** очевидно, используется в любых условиях, проблемы этого процесса - уважение конкурирующих потребностей исследований против
- в конечном итоге производить полезные качественные возможности. Модель связана с TriBITS, системой сборки, интеграции и тестирования, которая служит прочной основой для этого **жизненного цикла** модель, и аспекты этого **жизненного цикла** Модель укоренилась в системе TriBITS. Действительно это **жизненный цикл** процесс, если следуют

Больше

**Авторы:** Лица : Жизненный цикл II Т. Тусано М. М. Героух Дж. М. Willenbring M. VanDerVanter К. Шаабер Р. Мартин М. Бидл М. Пилл

**Организации:** Раздел VI Исследования Стабильного TriBITS Раздел IV KAI C ++ этапов жизненного цикла IEEE Computer Society C

самоподдерживающейся Software МО CA VERA **Расположение:** Калифорния Новая функциональность Вашингтон США Нью-Мексико Ливермор ICSEW CSE USA Agile **Электронная почта:** bartletta@oml.gov **Электронная почта:** .oml.gov **Телефоны**

: 87185-1320 3.1415926-2.3-1.40.12 : : :

Открыть | Tagging & annotation | Предпросмотр

**Интерактивные эволюционные вычисления в разработке программного обеспечения на ранних этапах жизненного цикла**

2020-05-20T00:48:40Z

[00SimonsThesisOpeningPages.pdf](#)

- Интерактивные эволюционные вычисления в начале **жизненного цикла** разработка программного обеспечения Перейти к основному содержанию
- Расширенный поиск Интерактивные эволюционные вычисления в начале **жизненного цикла** разработка программного обеспечения Simons
- Абстрактный дизайн имеет основополагающее значение для разработки программного обеспечения. Действительно, рано **жизненный цикл** разработка программного обеспечения имеет решающее значение и оказывает значительное влияние на последующие разработки. Ухудшение дизайна может привести к вредным последствиям. Поэтому улучшение

Больше

Рисунок 4 - Списковая (тезисная) выдача результатов по запросу "lifecycle"

- Получить (0) -  
 IEEE (3) -  
 Diebold, Филипп (2) -  
 Мигель Гулао (2) -  
 И Ричард С. Лингер (Национальная лаборатория Ок-Риджа) (1) -  
 - (1) -  
 Амират (1) -  
 Эндрю Дж. Корнецки (Университет Аэронавтики им. Эмбри-Риддла) (1) -  
 Арнон Штурм (1) -
- Показать меньше (-) | больше (+)

- Персоналии**
- Proc (28) -  
 IEEE Software (22) -  
 M. (21) -  
 Разработка программного обеспечения (21) -  
 Springer (20) -  
 Конф (19) -  
 Спрингер-Верлаг (18) -  
 F. (16) -  
 IEEE Транзакции (16) -  
 K. (16) -

Список Предпросмотр Частотный глоссарий Изображения Видео Аудио Локации Аналитика

« Предыдущая result 1 из 102 Следующая »

2018-06-14T08:33:55Z [softwarepractice2012\\_abstract2.pdf](#)  
 Neil Chue Hong:  
**Untitled**

Content Plain text Tags & Annotations Meta Import & analysis process (ETL)

1 из 1 Автоматически

**Overview of the TriBITS Lifecycle Model: A Lean/Agile Software Lifecycle Model for Research-based Computational Science and Engineering Software**

Roscoe A. Bartlett, Oak Ridge National Laboratory  
 Michael A. Heroux, James M. Willenbring, Sandia National Laboratories

**Abstract**

Software **lifecycles** are becoming an increasingly important issue for computational science & engineering (CSE) software. The process by which a piece of CSE software begins life as a set of research requirements and then matures into a trusted high-quality capability is both commonplace and extremely challenging. Although an implicit **lifecycle** is obviously being used in any effort, the challenges of this process—respecting the competing needs of research vs. production—cannot be overstated.

Here we describe a proposal for a well-defined software **lifecycle** process based on modern Lean/Agile software engineering principles. What we propose is appropriate for many CSE software projects that are initially heavily focused on research but also are expected to eventually produce usable high-quality capabilities. The model is related to TriBITS, a build, integration and testing system, which serves as a strong foundation for this **lifecycle** model, and aspects of this **lifecycle** model are ingrained in the TriBITS system. Indeed this **lifecycle** process, if followed, will enable large-scale sustainable integration of many complex CSE software efforts across several institutions.

**Авторы**

Neil Chue Hong

**Персоналии**

James M. Willenbring  
 Michael A. Heroux

**Организации**

Research-Sandia National Laboratories  
 Abstract Software  
 CSE  
 Computational Science and Engineering Software  
 Oak Ridge National Laboratory  
 TriBITS

**Язык**

en

**Расширения документов**

pdf

**Группа типа контента**

Text document

**Content type**

application/pdf

Рисунок 5 - Документовая выдача списка результатов по запросу "lifecycle"

Выполнение аналогичного поискового запроса и поиска в системе Google показана на Рис. 6.

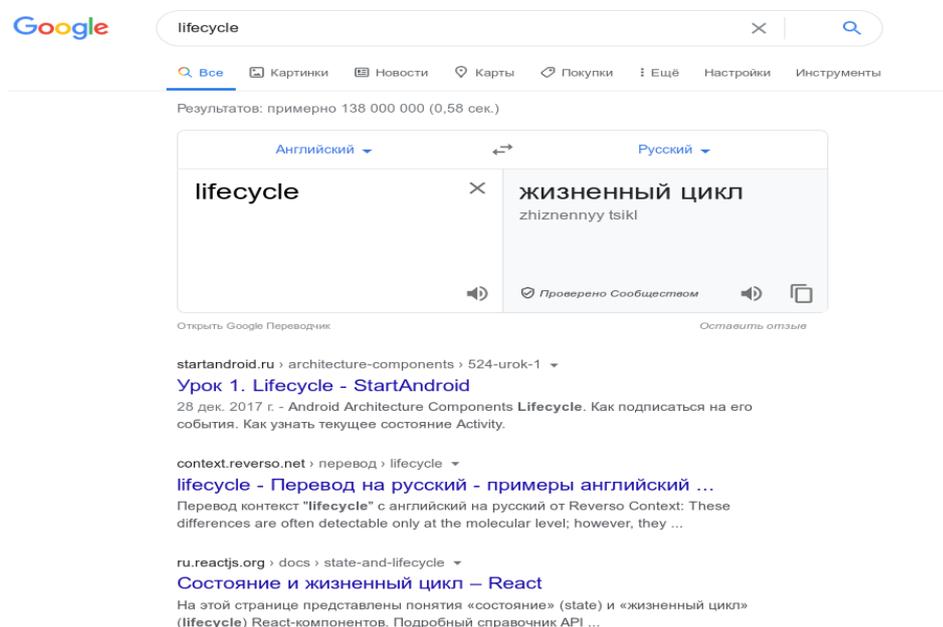


Рисунок 6 - Результаты поисковой выдачи для запроса "lifecycle" в Google.

Сравнение результатов выполнения поисковых запросов в системе "ПоискУМ" и Google показано на Табл. 1.

Сравнительный анализ поисковых систем. Таблица 1.

	Скорость обработки запроса	Привязка результатов к контексту научной отрасли (программной инженерии)	Возможность чтения текста статей прямо из поискового интерфейса	Сложность навигации непосредственно до текста результата	Возможность сортировки результатов запроса	Возможность индексации своего банка документов
ПоискУМ	1,2 сек	Да	Да	Нет (текст документа открывается сразу при нажатии на заголовок результата запроса)	Да (по сверхоперативному кэшу, по новизне, по старине)	Есть
Google	0.60 сек	Нет	Нет	Требуется дополнительная навигация по сайтам, выдаваемым в результатах запроса	Нет (сначала показываются самые популярные результаты)	Частично (только веб-сайт)

## Ключевые особенности пользовательского интерфейса реализации системы "ПоискУМ"

### Ведение поиска

Поиск запускается нажатием экранной кнопки "Поиск" возле строки ввода поискового запроса на основном окне системы. В результате нажатия на кнопку "Поиск" в окне поиска запрос отправляется на обработку поисковому движку. Последовательность действий, происходящих после нажатия на кнопку Search, отражена на рисунке 7.

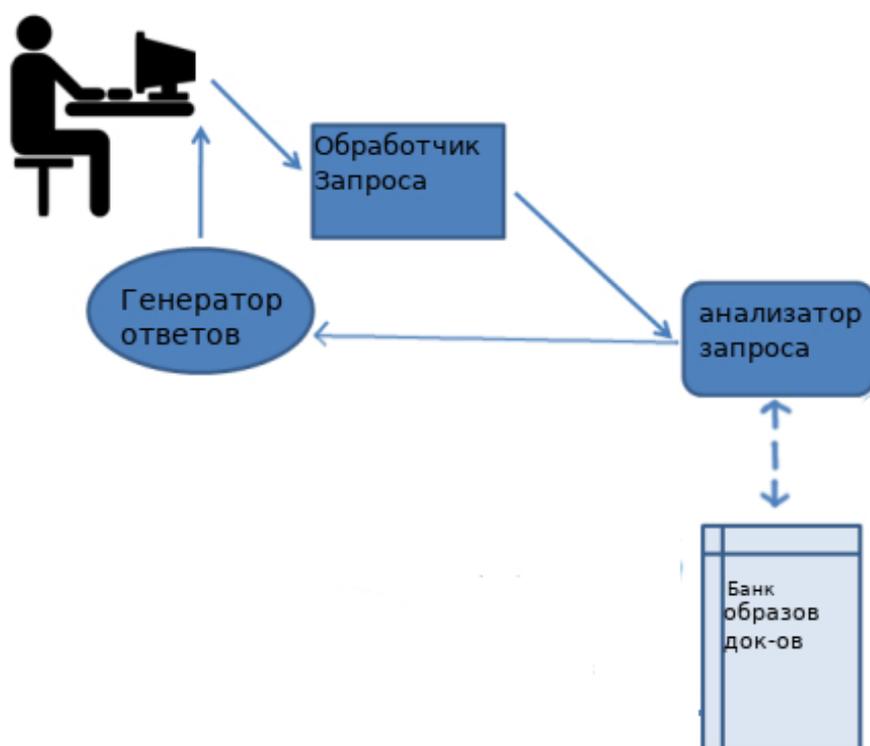


Рисунок 7 - Последовательность действий при обработке движком поискового запроса

Обработчик запросов осуществляет разбор по пробелам и по словам.

1. Пользователь может локализовать пространство поиска: в кэше, в сети Интернет (везде или в определённом диапазоне адресов), в кэше и в сети Интернет. Это реализуется следующим образом. В случае поиска по пространству кэша, никаких дополнительных манипуляций в интерфейсе системы, кроме начального монтирования банка Кэша, не требуется. В случае расширения пространства поиска на глобальную сеть Интернет, пользователю необходимо обратиться к интерфейсу командной строки и ввести команду "etl-web 'URL'", где URL - это ссылка на удалённый ресурс, подлежащий индексированию поисковым движком;
2. Запросы выполняются последовательно или параллельно (в зависимости от выбранного режима). В случае последовательного выполнения запроса, сразу после открытия веб-браузера появляется начальное окно системы, которое содержит в себе поисковую строку, как на рисунке 2. Параллельное выполнение запросов реализовано в системе в части нечёткого поиска через пункт меню "Аналитика -> "Нечёткий поиск по спискам имён" (рисунок 16). Для случая параллельного поиска поисковым движком предусмотрено использование нескольких потоков центрального процессора;
3. После ввода поискового запроса и нажатия на кнопку «Поиск», система отобразит количество найденных документов и их список образов документов, как на рисунке 8. Система выдаёт информацию на языке

найденного документа, но существует возможность русифицировать её средствами машинного перевода (Google, S3);

software  Параметры поиска

Список Предпросмотр Частотный глоссарий Изображения Видео Аудио Локации Аналитика ▾

Сортировка по

« Предыдущая Страница 1 из 142 (результаты 1 до 10 из 1418) Следующая »

**Программная инженерия**  
2020-02-24T21:45:53Z  
[программно-engineering.pdf](#)  
CourseLeaf

- Программное обеспечение инженерия Программное обеспечение Инженерное дело 1 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ENGINEERING Программное обеспечение Инжиниринг - это компьютерная дисциплина, которая занимается теоретическими и практическими аспектами строительства высокого качества. программное обеспечение системы, вовремя и в рамках
- бюджет. Программное обеспечение перед инженерами стоит задача детального анализа, проектирования, внедрения, тестирования, обслуживания и управления программное обеспечение проекты по разработке продуктов для широкого круга компьютерных приложений в обществе. Нарастающее давление доставлять качественно, надежно программное обеспечение товары
- в более короткие сроки быстро подпитывает спрос на компьютерных специалистов со специальной подготовкой в программное обеспечение инжиниринг и опыт работы в командах. Это давление обусловлено таким широко распространенным развитием, как программное обеспечение для требовательных и критически важных приложений, которые делают его

Автор (ы): CourseLeaf Persons: Proc Sec Требования Рrас ENGR Организации: DDC Лекция / Лабораторный курс естествознания Общая химия IIB GEOL 3501 Data Struc & Aily Anlys Бакалавр наук в области разработки программного обеспечения Последовательность разработки игр Общая химия

Дата документа

2020 (353)  
2019 (103)  
2018 (96)  
2017 (88)  
2016 (70)  
2015 (84)  
2014 (64)  
2013 (59)  
2012 (62)  
2011 (63)  
2010 (50)  
2009 (35)  
2008 (39)  
2007 (29)  
2006 (42)  
2005 (45)  
2004 (30)  
2003 (29)  
2002 (23)  
2001 (10)  
2000 (17)  
1999 (8)  
1997 (1)  
1994 (1)

Рисунок 8 - Главное окно системы после ввода поискового запроса и нажатия кнопки "Поиск", показано количество результатов и страниц листинга

4. Пользователь имеет возможность остановить выполнение запроса, продолжить или прекратить выполнение. Эта возможность реализована с помощью веб-обозревателя. После нажатия кнопки "Поиск", можно нажать на "X" в правом верхнем углу обозревателя, чтобы прекратить выполнение

запроса (оборвать обращение к веб-серверу). Продолжить запрос можно, нажав на клавиатуре клавишу F5, тем самым переиницировав запрос к веб-серверу. Рисунок 9 показывает реализацию данных возможностей.

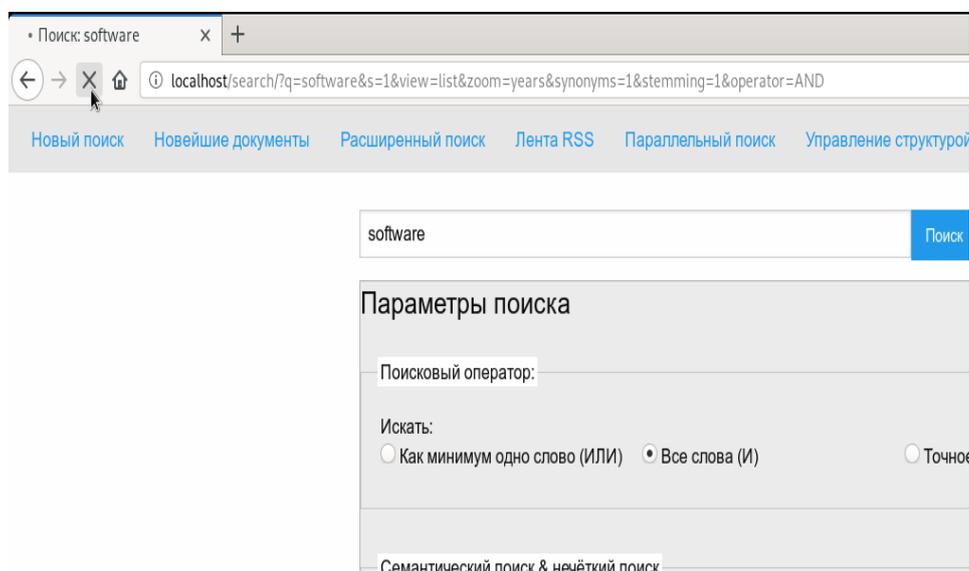


Рисунок 9 - остановка, прерывание и возобновления запроса

В составе сверхоперативного кэша ведётся журнал запросов. Доступ к осуществляется при помощи обращения к лог-файлу, расположенному по пути `"/var/log/apache2/access.log"`.

Пример журнала показан на Рис. 10.

```
127.0.0.1 - - [26/Jun/2020:15:16:09 +0000] "GET /search/?view=list&zoom=years&q=software+engineering&operator=AND&stemming=stemmin
years&q=software+engineering&operator=AND&stemming=stemming&synonyms=synonyms" Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/201
127.0.0.1 - - [26/Jun/2020:15:16:11 +0000] "GET /search/css/jquery/shadow.png HTTP/1.1" 404 487 "http://localhost/search/css/app.c
127.0.0.1 - - [26/Jun/2020:15:16:14 +0000] "GET /search-apps/annotate/rdf?uri=file%3A%2F%2Fhome%2Fuser%2FDocuments%2Fsf_SEpaper:
HTTP/1.1" 200 312 "-" "rdflib-4.2.2 (http://rdflib.net/; eikeon@eikeon.com)"
127.0.0.1 - - [26/Jun/2020:15:16:21 +0000] "GET /search-apps/annotate/rdf?uri=file%3A%2F%2Fhome%2Fuser%2FDocuments%2Fsf_SEpaper:
127.0.0.1 - - [26/Jun/2020:15:16:21 +0000] "GET /search-apps/annotate/rdf?uri=file%3A%2F%2Fhome%2Fuser%2FDocuments%2Fsf_SEpaper:"
```

Рисунок 10 - содержимое сверхоперативного кэша (истории обращений к веб-серверу)

Возможен возврат к запросу для редактирования и повторения. Пример возврата к запросу для редактирования и повторения представлен на рисунке 11.

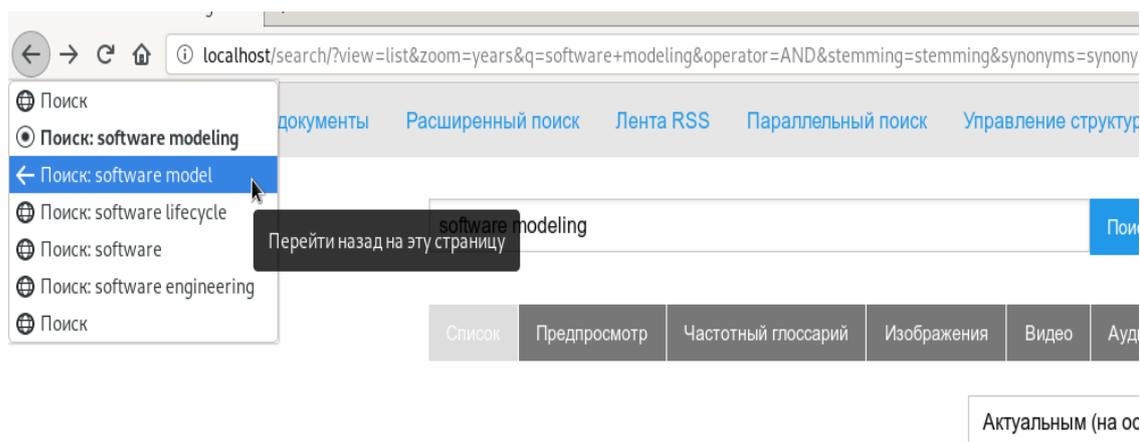


Рисунок 11 - пример возможности возврата к запросу (стрелка вверх слева)

### Осуществление поиска

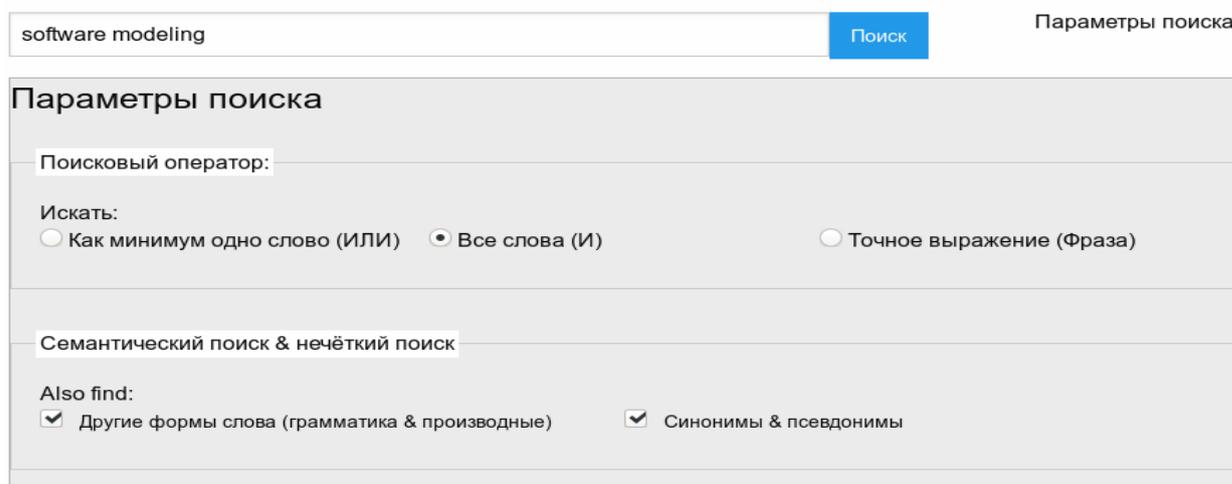
Соответствующим образом подготовленный, запрос передаётся поисковому движку.

Обеспечиваются следующие основные варианты опознавания документа, отвечающего поисковому запросу:

1. в испытуемом документе встречается хотя бы одно слово из поискового запроса;
2. в испытуемом документе встречаются все слова из поискового запроса;
3. в испытуемом документе встречается поисковый запрос целиком, как фраза.

Дополнительно может проверяться встречаемость грамматических форм поискового слова или однокоренных слов, а также синонимы и антонимы.

Вариант опознавания определяется перед запуском поиска в окне, которое открывается экранной кнопкой "Параметры поиска" в основном окне системы. Формат этого дополнительного окна показан на Рис. 12.



software modeling Поиск Параметры поиска

**Параметры поиска**

Поисковый оператор:

Искать:

Как минимум одно слово (ИЛИ)  Все слова (И)  Точное выражение (Фраза)

Семантический поиск & нечёткий поиск

Also find:

Другие формы слова (грамматика & производные)  Синонимы & псевдонимы

Рисунок 12 - Интерфейс расширенного поиска

### Результаты поиска

Результат поиска предъявляется пользователю в виде последовательности:

1. образов документов (соответствует результату поиска в публичных системах типа Google): навигация по последовательности (списку) документов осуществляется при помощи колёсика мыши и клавиш "Предыдущая", "Следующая" (см. рис 11), увидеть весь документ можно, нажав на заголовок каждого очередного результата поиска;
2. текстов документов («предпросмотр»): при предпросмотре показывается документ целиком прямо во вложенном окне просмотрщик PDF. Навигация по документу осуществляется путем нажатия на стрелки (см. рисунок 5).

Навигация по результатам поиска осуществляется аналогично, колёсиком мыши и кнопками "Предыдущая", "Следующая".

Последовательность может быть упорядочена:

1. по актуальности (на основе сверхоперативного кэша);
2. по убыванию даты публикации;
3. по возрастанию даты публикации.

Представление результатов поиска определяется выбором экранной кнопки в основном окне системы. На Рис. 3 выбрана кнопка "Список".

Если нажата кнопка "Список", то образы найденных документов будут

The screenshot displays a search results interface. The main content area shows a list of document thumbnails, each with a title, date, and PDF icon. The first document is titled "Европейская ассоциация науки и технологии программного обеспечения" (European Association of Science and Technology of Software Engineering) with a date of 2020-05-20T14:19:05Z. The second document is "Вы - цифровая библиотека" (You are a digital library) with a date of 2020-05-20T14:00:51Z. The third document is "Процесс прямой оценки результатов обучения по программам, основанный на принципах и практике программной инженерии" (Direct assessment of learning results by programs based on principles and practice of software engineering). To the right of the thumbnails is a sidebar with filters. The "Организации" (Organizations) filter is active, showing a list of organizations with their respective document counts: ACM (401), D (274), P (209), IEEE (205), J (190), UML (178), Microsoft (174), S (163), A (160), and C (156). Other filters include "Локации (местоположения)" (Locations) and "Язык" (Language).

выданы в окне выдачи результатов поиска, как показано на Рис. 13.

Рисунок 13 - Образы найденных документов выданы списком.

Если нажата кнопка "Предпросмотр", то образы найденных документов будут выданы в окне выдачи результатов поиска, как показано на Рис. 14

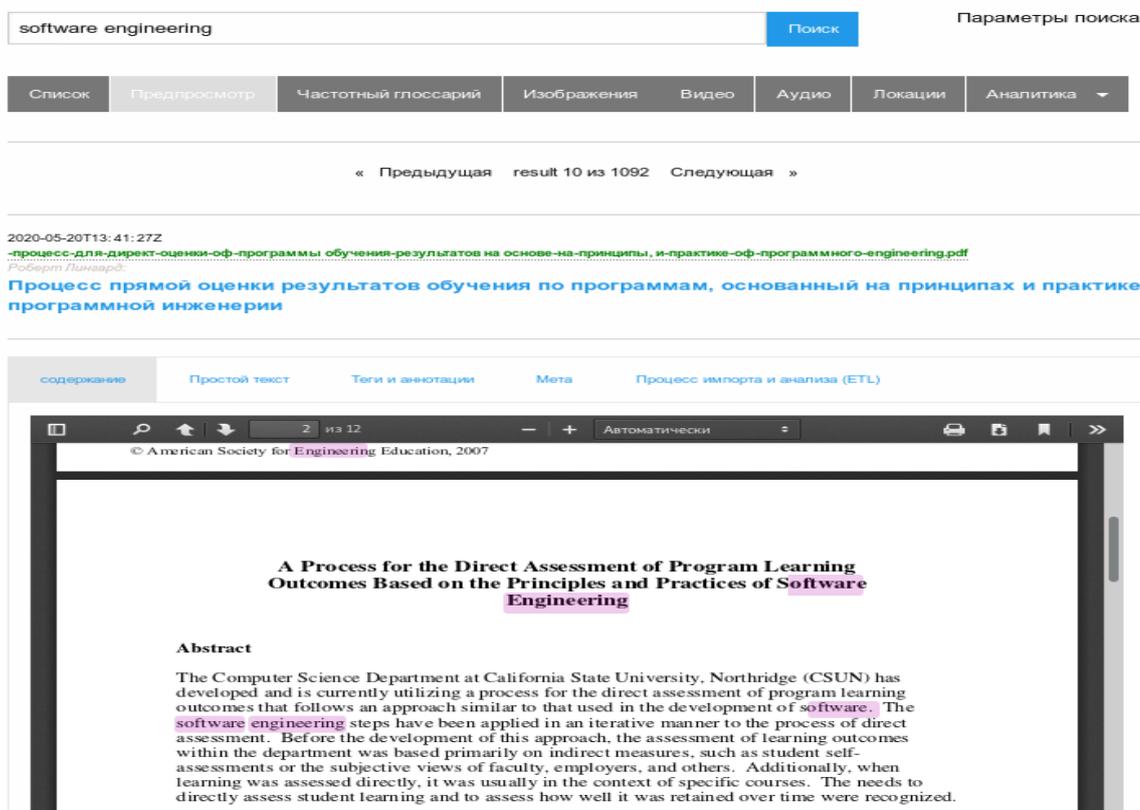


Рисунок 14 - Найденный документ, выданный в режиме Preview

(предпросмотра).

При нажатии на кнопку "Изображения" в основном окне результаты поискового запроса выдаются в виде списка изображений. Поскольку Кэш содержит преимущественно документы в формате pdf, этот вариант эквивалентен нажатию на кнопку List.

При нажатии на кнопку "Видео" или "Аудио" в основном окне результаты поискового запроса выдаются в виде списка видео или, соответственно, аудиоматериалов (при наличии таковых в банке документов).

## Поиск вариантов следующих терминов или названий

The screenshot shows a search interface with the following elements:

- Search input field containing "lifecycle" and a "Поиск" (Search) button.
- Navigation tabs: "Список", "Предпросмотр", "Частотный глоссарий", "Изображения", "Видео", "Аудио", "Локации", "Аналитика" (selected), and "Дата документа".
- A dropdown menu for "Аналитика" is open, showing options: "Нечёткий поиск по спискам имён", "Частотные глоссарии", "Связи (граф)", "Тенденции", "Таблица", and "Слова (список слов и облако слов)".
- Below the search bar, there is a message "Нет результатов" (No results) and a note: "Актуальным (на основе свёрхпер..." (Current (based on superper...)).
- Under "СОСТАВНЫЕ СЛОВА" (Component words), there are two checked options:
  - Префикс подстановочный знак (Пример: поиск прогулки найдет лунатизм)
  - Суффикс подстановочный знак (Пример: поиск сна найдет лунатик)
- At the bottom right, there is a note: "части запроса. Так что его недостаточно, чтобы соответствовать только поисковой записи списка." (parts of the query. So it is not enough to correspond only to the search record of the list.)

На Рис. 15 показан пример расширенного анализа при обработке поискового запроса, задан поиск по словосочетанию "lifecycle" и нажата кнопка "Аналитика".

Рисунок 15 - Начало поиска с расширенным анализом.

*Нечёткий поиск по спискам имён* – Нечеткий поиск по спискам имен (параллельный). Позволяет ввести список слов-синонимов, расширяя классический поиск. В качестве результатов поиска также будут выведены результаты, содержащие морфологические производные от вводимых в поисковом запросе слов. Например, запрос списком "software engineering, soft, engineer" в качестве результата выдаст не только результаты с прямым соответствием запросу, но также и содержание морфологические производные, как показано на Рис. 16.

Рисунок 16 - нечеткий поиск по спискам имён

*Частотные глоссарии* – Именованные объекты. При нажатии на эту кнопку, будут выведены именованные объекты, "тэги", соответствующие поисковому запросу. Это такие теги, как "Наименования организаций", "Контакты", "Страны", "Электронные адреса" тех сообществ людей, которые занимаются решением научных проблем, сформулированных в поисковом запросе.

<p><b>Персоналии</b></p> <ul style="list-style-type: none"> <li>• Springer (-) in 194 document(s)</li> <li>• Proc (-) in 157 document(s)</li> <li>• M. (-) in 152 document(s)</li> <li>• Java (-) in 142 document(s)</li> <li>• Software Engineering (-) in 130 document(s)</li> <li>• K. (-) in 124 document(s)</li> <li>• Springer-Verlag (-) in 124 document(s)</li> <li>• IEEE Transactions (-) in 114 document(s)</li> <li>• IEEE Software (-) in 108 document(s)</li> <li>• F. (-) in 92 document(s)</li> </ul> <p>Show less (-)   more (+)</p>	<p><b>Организации</b></p> <ul style="list-style-type: none"> <li>• ACM (-) in 401 document(s)</li> <li>• D. (-) in 274 document(s)</li> <li>• R. (-) in 209 document(s)</li> <li>• IEEE (-) in 205 document(s)</li> <li>• J. (-) in 190 document(s)</li> <li>• UML (-) in 178 document(s)</li> <li>• Microsoft (-) in 174 document(s)</li> <li>• S. (-) in 163 document(s)</li> <li>• A. (-) in 160 document(s)</li> <li>• C. (-) in 156 document(s)</li> </ul> <p>Show less (-)   more (+)</p>
<p><b>Локации (местоположения)</b></p> <ul style="list-style-type: none"> <li>• USA (-) in 278 document(s)</li> <li>• New York (-) in 246 document(s)</li> <li>• al. (-) in 188 document(s)</li> <li>• Germany (-) in 184 document(s)</li> <li>• UK (-) in 171 document(s)</li> <li>• Canada (-) in 136 document(s)</li> <li>• London (-) in 122 document(s)</li> <li>• Java (-) in 121 document(s)</li> <li>• NY (-) in 116 document(s)</li> <li>• Italy (-) in 104 document(s)</li> </ul> <p>Show less (-)   more (+)</p>	<p><b>Электронный адрес</b></p> <ul style="list-style-type: none"> <li>• permissions@acm.org. (-) in 18 document(s)</li> <li>• aabran@ele.etsmtl.ca (-) in 5 document(s)</li> <li>• Bx@PLW (-) in 4 document(s)</li> <li>• CSE@SE (-) in 4 document(s)</li> <li>• avi@info.ucl.ac.be (-) in 4 document(s)</li> <li>• fshull@fc-md.umd.edu (-) in 4 document(s)</li> <li>• m.daneva@utwente.nl (-) in 4 document(s)</li> <li>• p.lago@vu.nl (-) in 4 document(s)</li> <li>• perry@ece.utexas.edu (-) in 4 document(s)</li> <li>• Permissions@acm.org. (-) in 3 document(s)</li> </ul> <p>Show less (-)   more (+)</p>

Рисунок 17 - Окно "Частотный глоссарий"

Кнопка "Частотный глоссарий" в основном окне аналогична кнопке "Аналитика - Частотные глоссарии" в выпадающем списке.

*Тенденции* – Диаграмма трендов показывает результаты поиска соответствующих документов, опубликованных в научной среде в определенный день, например, год, месяц, день или час.

2014

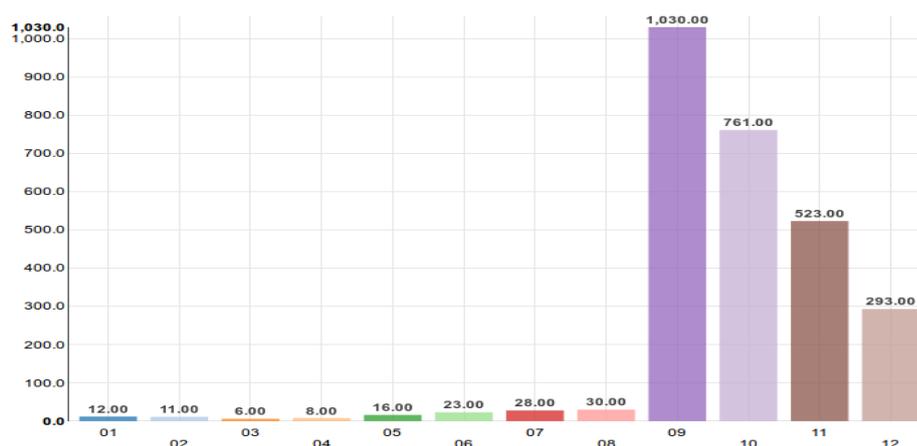


Рисунок 18 - диаграмма трендов

Система не просто выбирает ключевые слова без привязки к контексту, а морфологически и лексически, с учетом типов форматов документа разбирает все НТД в базе и выдаёт тезисно основные моменты и тенденции, обнаруженные в научно-технических статьях, а также «облака тегов», такие как: авторы, организации, контакты персон, занимающихся активными разработками в той области, которая соответствует введённому поисковому слову.. Пример такой тезисной выдачи (запрос: "Software engineering", выдержка одного результата из 1092) представлен на рисунке 19:

2010-12-06 12:49:03Z

940.pdf

www

- Конференция по технологиям и практике (STEP 2003). Цель данного документа - предложить уровни таксономии Блума для Руководства по Программное обеспечение инженерия Совокупность знаний (SWEBOK) темы для трех программное обеспечение Профили инженеров: новый выпускник, выпускник с четырехлетним опытом работы и опытный
- член программное обеспечение инжиниринг группа процессов. Уровни таксономии Блума предлагаются по темам четырех областей знаний Руководства SWEBOK: программное обеспечение поддержание, программное обеспечение инжиниринг управление, программное обеспечение инжиниринг процесс и программное обеспечение качественный. Предлагая таксономию Блума таким образом,
- Цель статьи - показать, как такие профили можно использовать в качестве инструмента для определения должностных инструкций, программное обеспечение инжиниринг описания ролей в программное обеспечение инжиниринг определение процесса, пути профессионального развития и программы обучения. Ключевые слова Программное обеспечение инженерия Тело

Больше

Автор (ы) : www Персоны : J. Mason F. Coallier S. Ikiz T. Aytaç D. Frailey J.-M. Лавуа С. Рамакришнан Дж. Гарбайоса С. Луди

Бенчмаркинг : Больше Организации : Информационные технологии Программное обеспечение жизненного цикла процессов

КАНАДА pbourque@ele.etsmtl.ca С AP S VI оценивает Руководство по техническому обзору программного обеспечения Университета Исландии

Образование Конференции Руководство SWEBOK Больше Места : Практика США Амстердам, Канада SWEBOK Knowledge Email

Электронная почта : pbourque@ele.etsmtl.ca aapril@ele.etsmtl.ca luigi.buglione@computer.org aabran@ele.etsmtl.ca Домен электронной почты

Рисунок 19 - Тезисная выдача (списком) основных тезисов по результату запроса

## Заключение

Несмотря на то, что информационно-справочная система "ПоискУМ" не выходит за рамки продвинутого ассоциативного поиска, результаты поискового запроса на примере запроса "lifecycle" свидетельствуют о том, что полученный опыт открывает новые возможности организации специализированных персональных поисковых систем. Поисковые системы общего назначения имеют лицензионные ограничения по контенту и не поддерживают возможностей, требуемых научному работнику для быстрого и простого накопления банка документов.

Система "ПоискУМ" позволяет, не затрачивая время на навигацию по сторонним веб-сайтам в поиске текста научных статей, находить актуальные документы по конкретной отрасли науки и техники, обозревать аннотации, библиографии, ключевые слова, тэги и контакты научных коллег, выкладывать тексты документов на свой рабочий стол для дальнейшей научной работы.

Полученная система может использоваться научно-техническими сотрудниками для поиска, выборки и анализа научных документов, в частности, в области аэрокосмической техники (как пример, в области программного инжиниринга средств авионики).

Система содержит предпосылки её дальнейшего развития и перехода в класс информационно-аналитических систем с возможностью подключения онтологий и семантического поиска.

Таким образом, разработанная и сформированная система "ПоискУМ", с учётом сформированных требований и нововведений позволяет вывести поисковый процесс по научно-техническим документам с перспективой развития до информационно-аналитической системы.

### **Библиографический список**

1. Коробейников П.А., Шишаев М.Г. Исследование семантической структуры навигационных интерфейсов типовых веб-ресурсов // Труды Кольского научного центра РАН. 2013. № 5 (18). С. 98 - 102.
2. Беляев Д.В. Ассоциативная модель смысловых контекстов и ее применение в задаче уточнения поисковых запросов // Труды МАИ. 2005. № 18. URL: <http://trudymai.ru/published.php?ID=34186>

3. Шишаев М.Г., Ломов П.А., Диковицкий В.В. Формализация задачи построения когнитивных пользовательских интерфейсов мультимедийных информационных ресурсов // Труды Кольского научного центра РАН. 2013. № 5 (18). С. 90 - 97.
4. Диковицкий В.В. Методы интеллектуальной обработки и представления информации в мультимедийных информационных системах промышленных предприятий // Труды СПИИРАН. 2015. № 5 (42). С. 56 - 76.
5. Диковицкий, В.В. и др. Современные методы создания мультимедийных вебресурсов на базе визуализации и обработки формализованной семантики // Вестник Кольского научного центра РАН. 2011. № 3 (6). С. 63 – 73.
6. Гагарин А.П., Сердюков. В.В. Сбор и анализ научно-технической информации на основе динамической онтологии // Современные информационные технологии и ИТ-образование. 2018. Т. 14. № 3. С. 644 - 651.
7. Сизиков Е.В., Сошников Д.В. Онтологическая поисковая система Jewel для реализации интеллектуального поиска в Интернет- и интранет-сетях // Труды МАИ. 2002. № 7. URL: <http://trudymai.ru/published.php?ID=34628>
8. Набатов А.Н., Веденяпин И.Э., Мухтаров А.Р. Применение онтологического подхода к процессу проектирования информационной системы // Труды МАИ. 2018. № 102. URL: <http://trudymai.ru/published.php?ID=99177>
9. Реакт П., Люк К. NDJSON. Newline-разделители в формате JSON. Стандарт для ограничивающей JSON в протоколах потока, 2018. URL: <https://clue.engineering/2018/introducing-reactphp-ndjson>

10. Crockford D. The application/json Media Type for JavaScript Object Notation (JSON) // Internet Engineering Task Force, 2006. DOI:[10.17487/RFC4627](https://doi.org/10.17487/RFC4627)
11. Open Research Corpus. SemanticScholar, 2019. URL: <http://s2-public-api-prod.us-west-2.elasticbeanstalk.com/corpus>
12. Mandalka M. Open Semantic Search: Free Software for Search Engine, Explorer for Discovery of large document collections, Media Monitoring, Text Analytics, Document Analysis & Text Mining platform, 2018. URL: <https://www.opensemanticsearch.org>
13. Chris Mattmann, Jukka Zitting. Tika in action, Manning Publications, 2011, 256 p.
14. Парамошкин Д.С., Шибанов С.В., Хмелевской Б.Г. Анализ применения метаданных в электронных библиотеках информационных ресурсов // Труды Международного симпозиума «Надежность и качество». 2009. Т. 1. С. 211 - 214.
15. Sidorov G. Syntactic Dependency Based N-grams in Rule Based Automatic English as Second Language Grammar Correction // International Journal of Computational Linguistics and Applications, 2013, vol. 4, no. 2, pp. 169 – 188.
16. Мкртчян Г.А., Вечерина Е.А., Добряшкина Е.Н., Чепракова Л.А. Лингвистическая теория: минимум базовых знаний для студентов технических вузов, изучающих иностранный язык // Труды МАИ. 2007. № 27. URL: <http://trudymai.ru/published.php?ID=34001>
17. Freed N., Borenstein N. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC2046, Innosoft, First Virtual Holdings, November 1996, URL: <http://tools.ietf.org/html/rfc2046>

18. Resnick P., Walker A. The text/enriched MIME Content-type. RFC-1896, February, 1996. URL: <https://tools.ietf.org/html/rfc1896.html>
19. Kilgarriff A. Putting Frequencies in the Dictionary // International Journal of Lexicography, 1997, no. 10 (2), pp. 135 - 155. DOI: [10.1093/ijl/10.2.135](https://doi.org/10.1093/ijl/10.2.135)
20. Фролов А.С. Разработка алгоритма нечеткого поиска на основе хэширования // Молодой ученый. 2016. № 13 (117). С. 357 - 360.

# **Experience of developing personal bibliographic retrieval system, oriented on specific area of scientific or engineering knowledge**

**Filimonov I.A.**

*Moscow Aviation Institute (National Research University), MAI,*

*4, Volokolamskoye shosse, Moscow, 125993, Russia*

*e-mail: [lja199@bk.ru](mailto:lja199@bk.ru)*

*e-mail: [ilafilimonov@mai.education](mailto:ilafilimonov@mai.education)*

## **Abstract**

The article regards a case of personalized bibliographic retrieval system, as well as the examples of its application. The system is accommodated with the resources of the AI Corpus scientific publication, established at the Pole Allen Artificial Intelligence Institute. It complements such retrieval systems, as Google, and by engaging of certain open components can be extended by the functions of searching, referencing and analyzing of a higher level. The author proposes an approach, allowing perform a two-stage information retrieval, i.e. by the search in the Global Information Bank, and thereafter by the fine search within the boundaries of the information asset being retrieved with regard to the special areas of the system user interests.

To clarify of the problems of this article, a partial classification of informational systems by the nature and role of the informational retrieval being performed in them was proposed. The basic requirements to the implemented system were formulated. A technique for scientific documents passports cataloging was proposed. The author performed comparative analysis of the implemented system with the general-purpose retrieval systems. Systems of information analysis services for a specialist in the area of

software engineering are the research objects, dedicated to the problem under discussion. Physical purpose of the study consists in implementing experimental individualized bibliographic retrieval systems for both scientific and technical workers. The implemented system can be employed as well by the scientific and technical personnel for search, selection and analysis of scientific documents in the field of aerospace engineering.

**Keywords:** search engine, resource collection, two-stage search, bank of scientific publications, user interface.

### References

1. Korobeinikov P.A., Shishaev M.G. *Trudy Kol'skogo nauchnogo tsentra RAN*, 2013, no. 5 (18), pp. 98 - 102.
2. Belyaev D.V. *Trudy MAI*, 2005, no. 18. URL: <http://trudymai.ru/eng/published.php?ID=34186>
3. Shishaev M.G., Lomov P.A., Dikovitskii V.V. *Trudy Kol'skogo nauchnogo tsentra RAN*, 2013, no. 5 (18), pp. 90 - 97.
4. Dikovitskii V.V. *Trudy SPIIRAN*, 2015, no. 5 (42), pp. 56 - 76.
5. Dikovitskii, V.V. et al. *Vestnik Kol'skogo nauchnogo tsentra RAN*, 2011, no. 3 (6), pp. 63 – 73.
6. Gagarin A.P., Serdyukov V.V. *Sovremennye informatsionnye tekhnologii i IT-obrazovanie*, 2018, vol. 14, no. 3, pp. 644 - 651.

7. Sizikov E.V., Soshnikov D.V. *Trudy MAI*, 2002, no. 7. URL: <http://trudymai.ru/eng/published.php?ID=34628>
8. Nabatov A.N., Vedenyapin I.E., Mukhtarov A.R. *Trudy MAI*, 2018, no. 102. URL: <http://trudymai.ru/eng/published.php?ID=99177>
9. Reakt P., Lyuk K. *NDJSON. Newline-razdeliteli v formate JSON. Standart dlya ogranichivayushchei JSON v protokolakh potoka*, 2018. URL: <https://clue.engineering/2018/introducing-reactphp-ndjson>
10. Crockford D. The application/json Media Type for JavaScript Object Notation (JSON), *Internet Engineering Task Force*, 2006. DOI: [10.17487/RFC4627](https://doi.org/10.17487/RFC4627)
11. *Open Research Corpus. SemanticScholar*, 2019. URL: <http://s2-public-api-prod.us-west-2.elasticbeanstalk.com/corpus>
12. Mandalka M. *Open Semantic Search: Free Software for Search Engine, Explorer for Discovery of large document collections, Media Monitoring, Text Analytics*, Document Analysis & Text Mining platform, 2018. URL: <https://www.opensemanticsearch.org>
13. Chris Mattmann, Jukka Zitting. *Tika in action*, Manning Publications, 2011, 256 p.
14. Paramoshkin D.S., Shibanov S.V., Khmelevskoi B.G. *Trudy Mezhdunarodnogo simpoziuma "Nadezhnost' i kachestvo"*, 2009, vol. 1, pp. 211 - 214.
15. Sidorov G. Syntactic Dependency Based N-grams in Rule Based Automatic English as Second Language Grammar Correction, *International Journal of Computational Linguistics and Applications*, 2013, vol. 4, no. 2, pp. 169 – 188.
16. Mkrtchyan G.A., Vecherinina E.A., Dobryashkina E.N., Cheprakova L.A. *Trudy MAI*, 2007, no. 27. URL: <http://trudymai.ru/eng/published.php?ID=34001>

17. Freed N., Borenstein N. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC2046*, Innosoft, First Virtual Holdings, November 1996, URL: <http://tools.ietf.org/html/rfc2046>
18. Resnick P., Walker A. *The text/enriched MIME Content-type. RFC-1896*, February, 1996. URL: <https://tools.ietf.org/html/rfc1896.html>
19. Kilgarriff A. Putting Frequencies in the Dictionary, *International Journal of Lexicography*, 1997, no. 10 (2), pp. 135 - 155. DOI: [10.1093/ijl/10.2.135](https://doi.org/10.1093/ijl/10.2.135)
20. Frolov A.S. *Molodoi uchenyi*, 2016, no. 13 (117), pp. 357 - 360.