



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский технологический университет»
МИРЭА

ОТЧЁТ
о преддипломной практике

студента 2 курса учебной группы ИКМО-02-15 института ИТ Университета
Пыжова Владислава Олеговича

.1. Практику проходил с 09.02.2017 по 15.05.2017 на кафедре ИППО
института ИТ МИРЭА.

.2. Задание на практику выполнил в полном объёме, качественно и в срок.

.2.1. Не выполнены следующие пункты задания: таковые отсутствуют.

.3. Содержание выполненной на практике работы и достигнутые
результаты:

Выполнены все пункты задания, а именно – изучены методические
указания и требования по подготовке, оформлению и представлению к защите
ВКР, разработан эскизный вариант аналитических обзоров к первой главе ПЗ
ВКР, а также методики и технологической поддержки составляющей ВКР. В
результате обеспечено достижение цели, обозначенной в задании, а именно –
произведены сбор, систематизация и обобщение материала, необходимого для
разработки и написания ВКР. Таким образом, согласно положениям программы
практики в результате её выполнения в порядке самооценки практикант
считает достигнутыми расширение профессиональных умений, навыков и
компетенций в области проведения аналитического исследования по теме
предстоящей ВКР, а также самостоятельного выбора и практического
применения методов исследования и методики инженерной составляющей
ВКР. Подробное изложение материалов и результатов практики представлено
далее в Пояснительной записке (ПЗ) и в её электронной версии, включающей
сформированные в её процессе тематические библиотеки практиканта, а также
библиографию.

.4. Заключение руководителя по практике студента представлено
отдельным документом.

Практикант Пыжов В.О., ИКМО-02-15 _____

Руководитель доцент Панов А.В. _____

Эл.версия планово-отчётной документации по практике проверена, принята и
размещена под экстранет защитой второго уровня на тематическом минипортале кафедры
ИППО: Сеницын А.В. _____

Оценка по итогам практики: _____ доц.Панов А.В.

УДК 004.6, 004.043, 004.89

Исполнитель: студент группы ИКМО-02-15 Пыжов В.О.

Руководитель: доцент кафедры ИППО Панов А.В.

Пыжов В.О., Архитектоника семантических менеджеров информационных ресурсов на основе блочного модульного конструирования / отчет о преддипломной практике по Кафедре Инструментального и прикладного программного обеспечения (ИППО) института Информационных технологий (ИТ) МИРЭА по направлению профессиональной подготовки ВШ РФ «Программная инженерия»/ Пояснительная записка (ПЗ) / Руководитель – доц. А.В. Панов М. МИРЭА. 2017 – 28 стр., 7 иллюст., 1 табл., 13 исп.ист..

Цель выпускной квалификационной работы магистра заключается в выработке архитектуры семантического менеджера информационных ресурсов и решению вопросов, связанных с ее реализацией. В рамках преддипломной практики выполняется аналитический обзор методик для реализации семантического описания информационных ресурсов и существующих семантических менеджеров информационных ресурсов. На основе этого анализа предложена новая концепция и архитектура, позволяющая ее реализовать.

The purpose of master's thesis is the development of semantic manager for information resources architecture and solving the tasks concerned with one. The work contains analysis of methods for semantic description of information resources and existed semantic managers for information resources. The new concept and the architecture that allows it to implement that concept are proposed based on this analysis.

МИРЭА: 119454, Москва, пр-т Вернадского, д. 78

кафедра Инструментального и прикладного программного обеспечения (ИППО)

Тираж: 1 экз. (на правах рукописи)

Файл: «ПреддипломПрактика_Пыжов_ИКМО-02-15.docx», исполнитель Пыжов В.О.

© В.О.Пыжов

Оглавление

Список терминов и сокращений	4
Введение	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1. Семантическое описание информационных ресурсов	7
1.2 Возможности файловых систем	8
1.2.1 Иерархические файловые системы	8
1.2.2 Семантические файловые системы	11
1.3 Семантические менеджеры информационных ресурсов	12
1.3.1 Property System	12
1.3.2 Tracker	14
1.3.3 NEPOMUK	15
1.4 Специализированные менеджеры для работы с конкретными видами информационных ресурсов	16
1.4.1 Griffith	16
1.4.2 Picasa	17
1.4.3 Mendeley	17
1.5 Выводы	18
2 КОНЦЕПЦИЯ И АРХИТЕКТУРА СЕМАНТИЧЕСКОГО МЕНЕДЖЕРА ИНФОРМАЦИОННЫХ РЕСУРСОВ	20
2.1 Предлагаемая концепция семантического менеджера информационных ресурсов	20
2.2 Предлагаемая архитектура семантического менеджера информационных ресурсов	21
Заключение	25
Список используемой литературы	27

Список терминов и сокращений

Архитектоника – это построение системы, соотношение частей системы, расположение и взаимную связь его компонентов (слагаемых), образующих вместе систему.

Информационные ресурсы – отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах);

ИР – информационный ресурс.

Метаданные – информация о другой информации, или данные, относящиеся к дополнительной информации о содержимом или объекте. Метаданные раскрывают сведения о признаках и свойствах, характеризующих какие-либо сущности, которые позволяют автоматически искать и управлять ими в больших информационных потоках.

API – интерфейс программирования приложений, представляющий собой набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования.

Введение

Информационные ресурсы окружают человека повсюду, это и книги, и фильмы, музыкальные произведения, песни, фотографии, документы и статьи. Традиционным представлением таких информационных ресурсов в электронно-вычислительных машинах является файловое представление в рамках той или иной иерархической файловой системы, т.е. информационные ресурсы представляются в виде файлов или массивов файлов, сгруппированных по различным каталогам. Это порождает множество сложностей при работе с ними. Начиная от сложностей, связанных с автоматизированной обработкой таких ресурсов, и заканчивая тем, что это противоречит естественной природе человека, поскольку люди склонны ассоциировать объекты в реальной жизни с набором различных их атрибутов, нежели с каким-то конкретным положением в пространстве, и когда возникает необходимость найти тот или иной информационный ресурс, пользователи зачастую просто не могут вспомнить, какое имя они дали файлу и где они его разместили [1,2].

Одним из направлений, позволяющих преодолеть эти и другие проблемы, является семантическое представление информационных ресурсов. Как правило, так же, как и в сети Интернет, оно включает в себя набор машиночитаемых метаданных, описывающих информационный ресурс, а также зачастую связей между этим и другими ресурсами [3].

Можно выделить три основных вектора, по которым ведется разработка таких решений для электронно-вычислительных устройств:

- семантические файловые системы;
- средства для работы с метаданными, базирующиеся на возможностях существующих несемантических файловых систем (как правило, семантическая информация хранится или внутри файлов или сцепляются с ними возможностями файловой системы, либо хранятся во внешнем файле, базе данных и т.п.);

– отдельным классом идут специализированные менеджеры для работы с какими-то определенными видами информационных ресурсов (коллекцией фильмов, фотографий, научных статей и т.п.)

Тем не менее, несмотря на большое количество таких решений, к сожалению, ни одно из них так и не получило широкого распространения и не смогло предоставить полноценного решения для перехода от файлового представления к семантическому и сломить традиционную парадигму работы с информационными ресурсами на электронно-вычислительных устройствах. Среди проблем можно отметить: невозможность массового внедрения в силу несоответствия используемым пользователями операционных систем или иных обстоятельств, нерешенность вопроса с автоматизированным получением семантического описания для информационных ресурсов и необходимость ручного их заполнения, привязанность к файловому представлению, узкая специализация решения и т.п.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Семантическое описание информационных ресурсов

Применение представления информационных ресурсов на базе их семантического описания позволяет значительно облегчить работу с ними как программным средствам, так и в первую очередь, пользователю в сравнении с традиционным иерархическим файловым представлением. Зачастую, семантическое описание включает в себя лишь метаданные, которые характеризуют информационный ресурс (авторство, издатель и т.п.), но в более сложных реализациях также могут присутствовать и отношения между информационными ресурсами.

Способы для реализации семантического описания информационных ресурсов с точки зрения их положения в иерархии операционной системы условно можно представить в виде следующей иерархии:

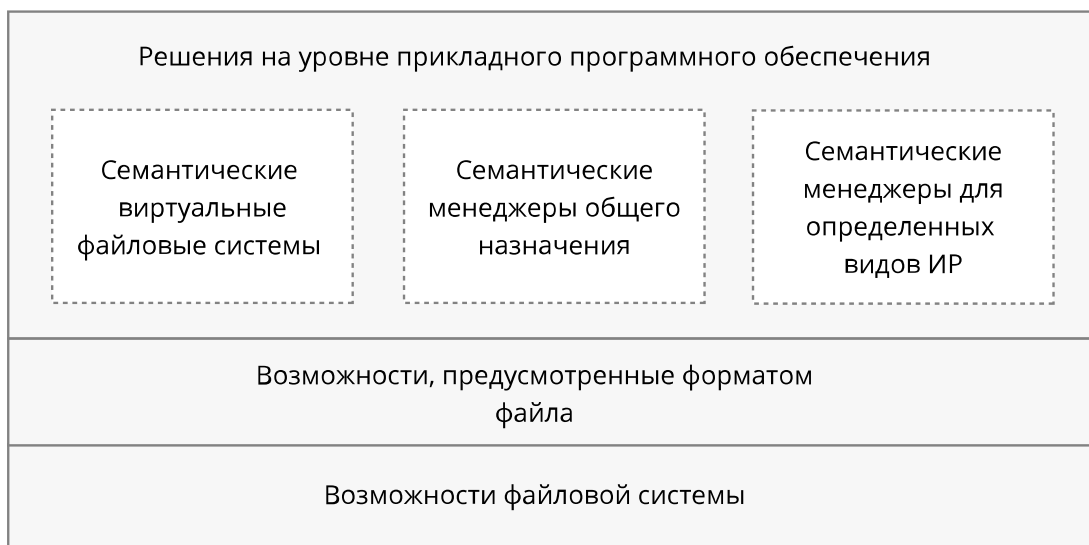


Рисунок 1.1 – Иерархия методов для реализации семантического описания информационных ресурсов

1.2 Возможности файловых систем

Файловые системы можно условно разделить на две группы [4].

К первой относятся иерархические файловые системы (т.е., такие в которых файлы располагаются в соответствии с иерархией, выстроенной благодаря каталогам, вложенным друг в друга), на базе которых функционируют современные операционные системы, ко второй

Ко второй группе относятся относительно новые так называемые семантические файловые системы, которые частично или полностью отказались от иерархической структуры в пользу подходов основанных на семантике и метаописании хранимых ресурсов.

1.2.1 Иерархические файловые системы

Изначально иерархические файловые системы не предоставляли инструментов для добавления пользователем или прикладным программным обеспечением метаданных для файлов. Единственным средством описания файлов были их имена, то место в иерархии именованных каталогов, которые они занимают, а также ряд служебных атрибутов таких как: дата создания файла, дата изменения файла и т.д.

Позже, когда стало очевидным, что чистая иерархическая структура далеко не всегда удобна для работы с большим числом различных файлов были предложены некоторые механизмы, которые были призваны упростить работу с ними и предоставить инструменты для более выразительного и подробного их описания, которые впоследствии будут рассмотрены далее.

Символические и жесткие ссылки

Одним из первых инструментов, позволяющих упростить организацию файлов в иерархических файловых системах стали так называемые символические и жесткие ссылки, которые позволили одному и тому же файлу находиться в разных каталогах.

Отличие между ними достаточно тонкое:

– символические ссылки, представляют собой специальный файл в файловой системе, в котором вместо пользовательских данных содержится путь к файлу, соответственно при обращении к символической ссылке система определяет, к какому объекту на самом деле запрашивается доступ, и его обеспечивает;

– жесткие ссылки, которые отличаются от символических ссылок, (имеющих основное имя файла), тем, что позволяют иметь для одного и того же файла несколько равноправных имен, отражающих, тем не менее одно и то же содержимое, но абсолютно независимы друг от друга.

Расширенные атрибуты и альтернативные потоки данных

Современные файловые системы зачастую предоставляют помимо возможности иерархического упорядочивания по каталогам и поддержки различных видов ссылок, возможность для снабжения файлов различными видами метаданных.

Пусть и относительно грубо, но можно разделить файлы системы, которые поддерживают:

– расширенные атрибуты (extended attributes), которые предоставляют средства для сцепления с файлами структурированных пар имя/значение¹

– альтернативные потоки данных (alternative data streams / file forks), которые позволяют хранить содержимое файла в отдельных областях для хранения данных рядом с содержимым файла.

Как правило, расширенные атрибуты файлов ограничены в размерах и значительно меньше содержимого, которое может храниться в альтернативных потоках данных, тем не менее, идейно, оба подхода близки друг к другу и с точки зрения реализуемых целей идентичны: они позволяют дополнять файлы метаописанием, однако эти средства, как правило, недостаточно выразительны, чтобы описать отношения между файлами или категоризовать их.

¹ Например, в файловой системе ext4 у узла, представляющего собой отдельный файл, используется ссылка `inode.i_file_acl` указывающая на блок с расширенными атрибутами (примечательно то, что имя ссылки происходит от ACL – Access Control List и связано это с тем, что раньше блок, на который она ссылалась первоначально использовался именно для хранения прав доступа)[5]

Таблица 1. Сравнение возможностей иерархических файловых систем и семантических для работы с семантическим описанием[4]

	Жесткие / символьные ссылки	Теги / ключевые слова	Атрибуты	Внутренние отношения	Внешние отношения	Несколько потоков	Полная поддержка чтения/записи	Виртуальная файловая иерархия	API	Язык структурированных запросов
Иерархические файловые системы										
FAT							✓	n/a		
NTFS	✓		✓			✓	✓	n/a	✓	
HFS	✓		✓			✓	✓	n/a	✓	
ext3/ext4	✓		✓				✓	n/a	✓	
Семантические файловые системы										
WinFS		✓	✓	✓			✓	✓	✓	✓
SFS	✓		✓					✓		✓
AttrFS	✓	✓	✓				✓	✓		✓
Presto			✓		✓			✓	✓	✓
LISFS			✓					✓		✓
TagFS	✓	✓	✓				✓	✓	✓	✓
libferris			✓				✓	✓	✓	✓
LiFS	✓		✓	✓		✓	✓	✓	✓	✓

Выше приведена таблица сравнения возможностей семантических файловых систем и иерархических файловых систем по возможностям работы с семантическим описанием. Подводя некоторый итог, можно сказать, что, несмотря на то, иерархические файловые системы значительно продвинулись в сторону семантической описания, иерархическая организация файлов остается основным механизмом

1.2.2 Семантические файловые системы

Семантические файловые в отличие от иерархических файловых систем дают пользователю возможность организовывать хранение и систематизацию файлов в соответствии с их содержанием, нежели какой-то иерархией, и предоставляют средства для того, чтобы без труда это делать [6]. То, как они реализуются, как выглядит их семантическое описание, и как решаются другие вопросы, варьируются от реализации к реализации, и далее для иллюстрации этого будут рассмотрены некоторые из наиболее известных реализаций семантических файловых систем.

Tagsistant

Tagsistant – это семантическая файловая система, предлагающая описывать файлы при помощи набора кратких идентификаторов «тегов» (например, для фотографии – это могут быть «море», «отдых», «горы» и т.д.). По выражению автора: *«для меня что семантическая файловая система – это, по существу, инструмент, который позволяет каталогизировать файлы и извлекать подмножества при помощи логических запросов»* [1,7].

Ключевое отличие используемой методики в том, что если в «классической» иерархии между файлами и директориями используется отношение «много файлов к одной директории, файл принадлежит директории», то здесь реализовано отношение «у файла много тегов, у тега много файлов». Причем примечательной особенностью данной файловой системы является то, что во многом логика работы с ней похожа на работу с иерархической файловой системой и используются те же самые команды.

Например:

- для создания тега создается директория в служебной директории «tags»;
- для назначения тега файлу он копируется в соответствующую директорию, обозначающую тот или иной тег;
- для просмотра файлов, имеющих один и тот же тег.

Другой особенностью данной файловой системы является то, что она работает в пользовательском пространстве. Такое становится возможным благодаря использованию специального модуля ядра — FUSE (File system in User Space), которое совместимо с операционными системами семейства Linux, BSD и OSX [insight]. Хранение же информации о тегах осуществляется в локальной базе данных SQLite, что как отмечают приводит к некоторым проблемам с производительностью [1].

1.3 Семантические менеджеры информационных ресурсов

Наибольший интерес представляют решения данной группы, причем они крайне разнообразны, начиная от простого расширения возможностей операционной системы по работе с метаданными и заканчивая менеджерами, реализующими относительно молодую концепцию семантического рабочего стола (semantic desktop).

1.3.1 Property System

Для работы с метаданными файла в операционных системах Windows используется система свойств (Property System), которая представляет собой расширяемую систему чтения/записи метаданных файлов. Для получения и редактирования метаданных используются COM-объекты, называемые обработчиками свойств (Property Handler), каждый из которых ассоциирован с конкретным типом файлов[8].

Полученные ими свойства отображаются в свойствах файла и могут быть использованы для поиска файлов по их метаописанию:

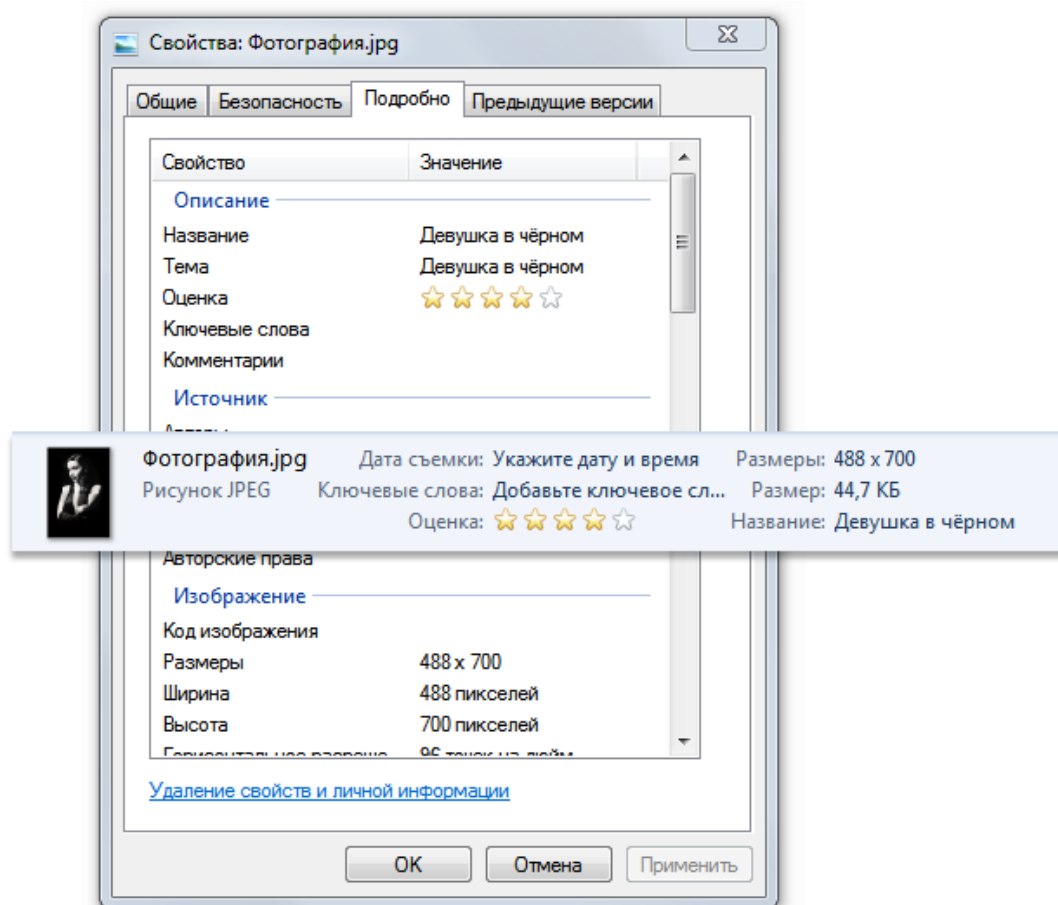


Рисунок 1.2 – Просмотр свойств изображения, сохраненного в формате .jpg

По умолчанию операционная система Windows Vista и более поздние её версии имеют соответствующие обработчики свойств для большого числа типов файлов, среди которых, например:

- исполняемые приложения и библиотеки (.exe, .dll, .lnk);
- файлы шрифтов (.ttf, .otf);
- основные файлы изображений (.jpg, .png, .bmp, .gif);
- аудио и медиафайлы (.mp3, .wma, .wmv, .mpg, .avi, и др.);
- ссылки на вебстраницы (.url).

Разработчики, создающие собственные форматы файлов, могут зарегистрировать соответствующий обработчик свойств для того, чтобы пользователи могли обращаться к метаданным такого рода файлов.

1.3.2 Tracker

Tracker – это одновременно поисковой движок и в определенном смысле система хранения метаданных для операционных систем семейства Linux. Подобно остальным рассмотренным средствам, он осуществляет поиск по содержанию файла и его метаданным (начиная от стандартных, таких как имя файла, директория, формат и т.д. и заканчивая типичными для определенных форматов файла, таких как разрешение изображения, длительность аудио-файла и т.д.)[9].

По функционалу он близок к тому, что предоставляется современными операционными системами семейства Windows.

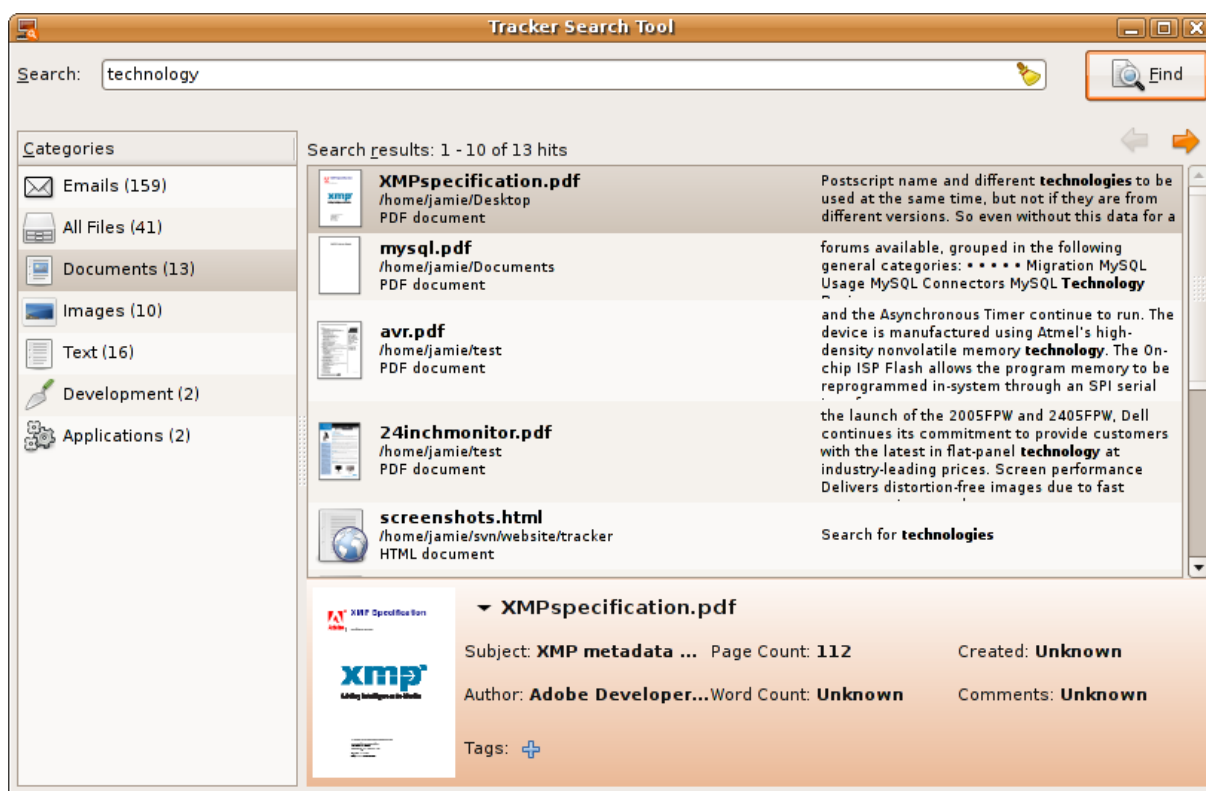


Рисунок 1.3 – Пример результата поиска

Среди особенностей можно отметить:

– формируемая база метаданных открыта для доступа со стороны других приложений, что позволяет другим приложениям использовать её в своей работе (например, добавлять скачанные музыкальные файлы в коллекцию и классифицировать их на основе метаданных);

- наличие удобной системы работы с тегами, которая не просто облегчает поиск файлов, а предоставляет возможность абстрагироваться от иерархической структуры файловой системы и просматривать файлы в структуре, сформированной на основе тегов.

Однако в свою очередь он имеет свои достаточно серьёзные недостатки:

- отсутствует гибкий интерфейс для создания средств по извлечению метаданных из файлов;

- метаданные (за исключением тегов) доступны только для чтения, таким образом, нет возможности оценить файл, добавить свои заметки, исправить некорректные данные и т.д.

1.3.3 NEPOMUK

NEPOMUK – открытая свободная реализация стандартов и API для KDE. Разработка акцентируется на активном использовании метаданных для описания файлов. На текущем этапе разработка концентрируется на использовании метаданных. Существуют три вида метаданных, которые можно найти на рабочем столе:

- метаданные, которые можно найти в файлах, хранящихся на локальном жестком диске, такие как тег, информация в аудио файлах, временные метки, или получить их простой индексацией текста. Эти метаданные могут быть извлечены и проиндексированы в любое время. Работа Strigi основывается на этом типе метаданных.

- метаданные созданные вручную пользователем. В наиболее простом случае это может быть комментарий к файлу или сообщению электронной почты.

- метаданные, которые не могут быть легко получены путём индексации и не создаются пользователем вручную. Сюда входит, например, адрес файла, который будет скачан с Интернета. Как только файл сохраняется на жестком диске, эта информация теряется. То же самое можно сказать о приложении

электронной почты: после того как вложение по электронной почте сохраняется на жесткий диск

1.4 Специализированные менеджеры для работы с конкретными видами информационных ресурсов

В этом разделе пойдёт речь о третьем подходе, к которому относятся специализированные менеджеры, предназначенные для работы с конкретными типами файлов.

1.4.1 Griffith

Griffith представляет собой свободный менеджер медиа-файлов с открытым исходным кодом, позволяющий систематизировать коллекцию фильмов[10].

Все записи хранятся в базе данных и при этом реализована возможность их экспорта в различные форматы. Одной из его отличительных и примечательных особенностей является автоматическое получение метаданных о фильме из интернета.

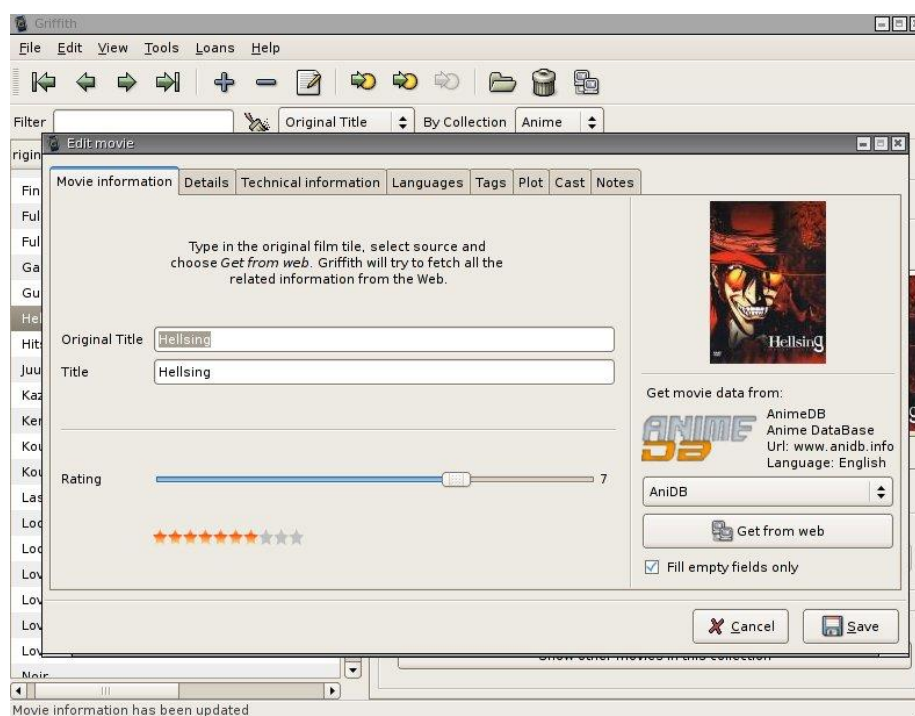


Рисунок 1.4 – Графический интерфейс Griffith

1.4.2 Picasa

Picasa – это бесплатная программа для просмотра и обработки цифровых фотографий, а также их систематизации. Последнее может быть реализовано при помощи поиска файлов с изображениями и сортировки фотографий по времени съёмок, а также вручную, путём группировки по папкам и альбомам, а также по изображенным на фотографиях людям[11].

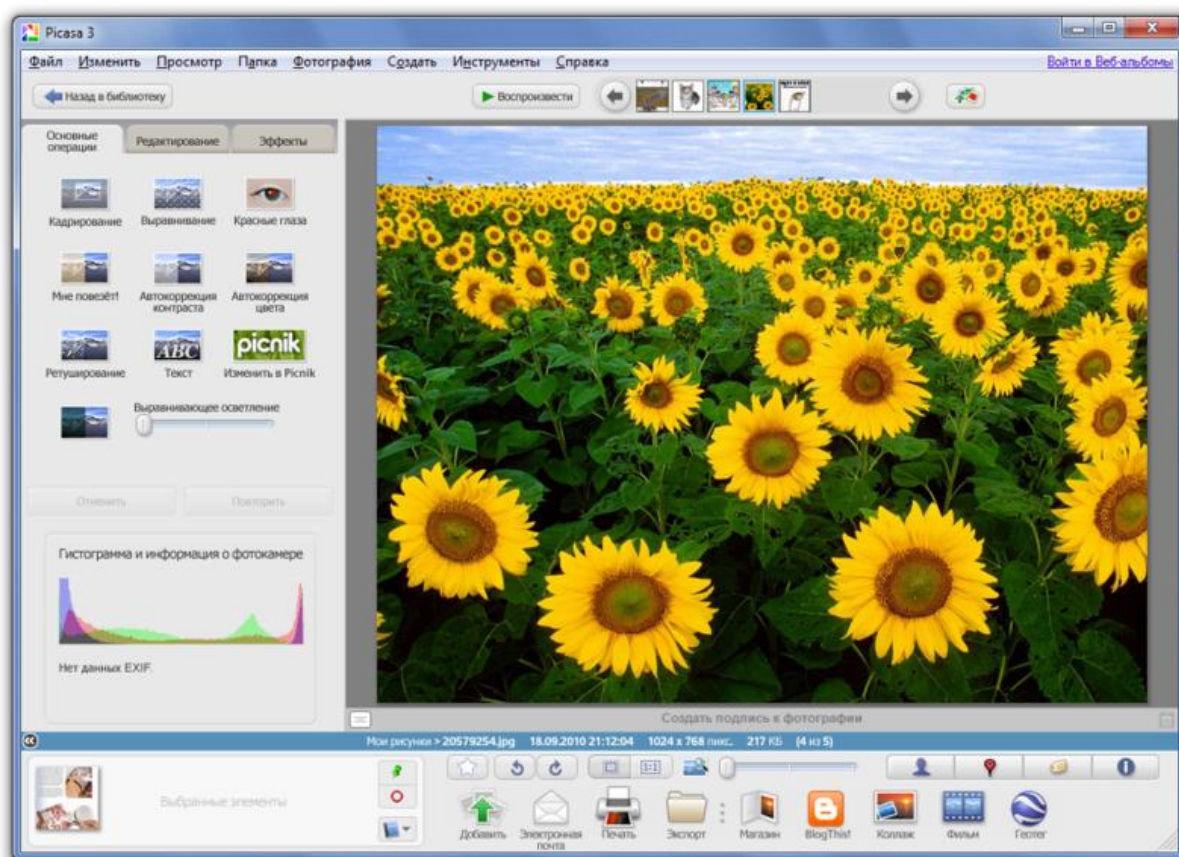


Рисунок 1.5 – Графический интерфейс Picasa

1.4.3 Mendeley

Mendeley представляет собой бесплатную программу для управления библиографической информацией и позволяет хранить и просматривать исследовательские труды, сохраненные в формате PDF, а также взаимодействовать с другими учёными в собственной международной социальной сети [12].

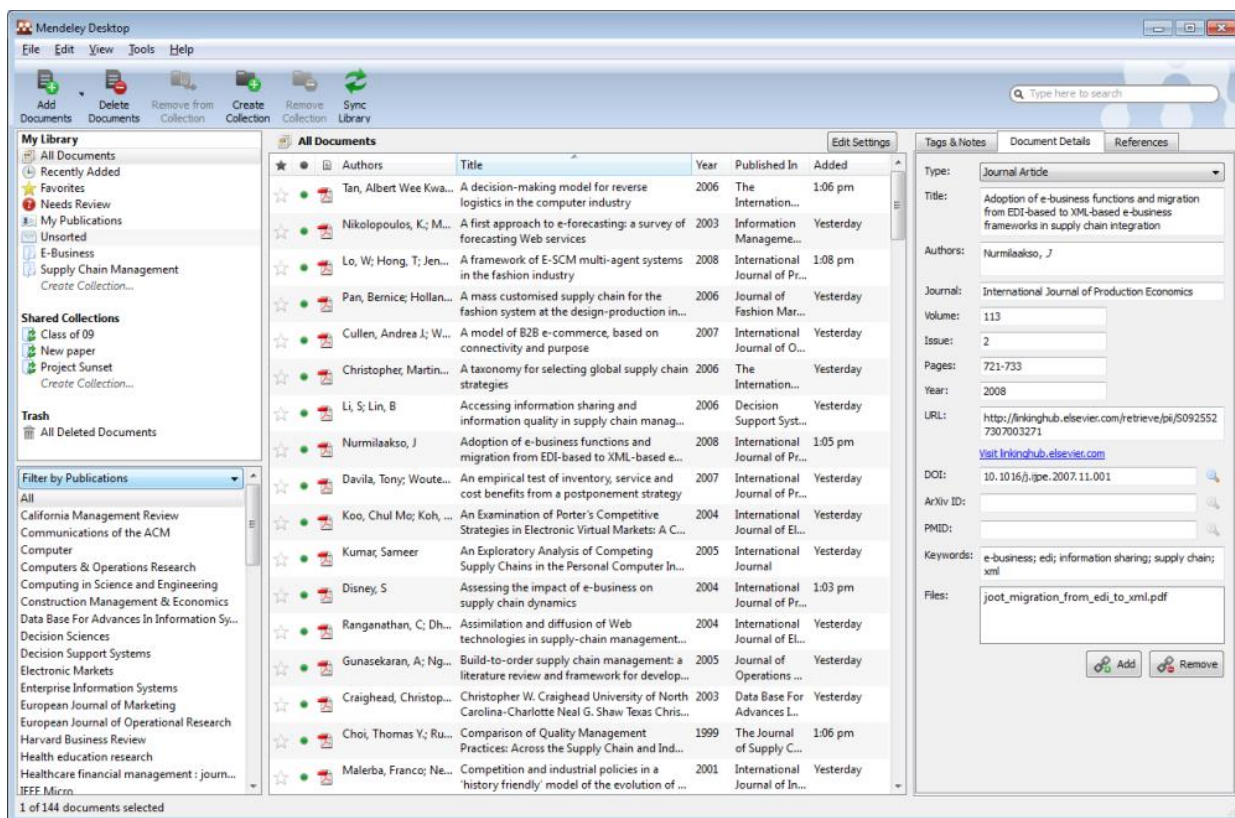


Рисунок 1.6 – Графический интерфейс Mendeley

Но Mendeley также как и остальные представленные примеры выполняет узкий спектр задач, что является основным недостатком такого рода менеджеров, хотя их функционал и гораздо богаче, чем у представленных ранее инструментов.

1.5 Выводы

Несмотря на огромное число представленных решений, можно выделить следующие нерешенные в них проблемы:

- многие решения охватывают только ограниченное число операционных систем или файловых систем, что, ограничивает возможность их практического применения;

- практически ни одно решение не предоставляет возможность для автоматического заполнения семантического описания информационных ресурсов, что значительно затрудняет применение этих решений на практике;

– ряд решений крайне узкоспециализированные и предназначены только для конкретных типов информационных ресурсов;

– значительное число решений не отказываются от иерархической парадигмы работы с файлами, а лишь снабжает их метаданными для облегчения работы с ними.

Нерешенность этих проблем обосновывает актуальность данного исследования, целью которого и является выработка такой концепции и архитектуры (а также решения связанных с их реализацией вопросов) семантического менеджера информационных ресурсов, которая позволила бы преодолеть проблемы известных решений и предоставить возможность удобной и полноценной работы с информационными ресурсами как с самостоятельными объектами на базе их семантического описания.

2 КОНЦЕПЦИЯ И АРХИТЕКТУРА СЕМАНТИЧЕСКОГО МЕНЕДЖЕРА ИНФОРМАЦИОННЫХ РЕСУРСОВ

Основу данной работы составляет описание новой концепции для построения семантического менеджера информационных ресурсов, разработка архитектуры, на базе которой она может быть воплощена, а также проработка самых различных вопросов, которые связаны с ее реализацией.

2.1 Предлагаемая концепция семантического менеджера информационных ресурсов

В основе предлагаемой концепции семантического менеджера информационных ресурсов лежат три базовых идеи.

Первая идея – это предоставление пользователю возможности работы с различными видами информационных ресурсов как с отдельными сущностями на базе их семантического описания. Данная идея постулирует в первую очередь то, что семантический менеджер информационных ресурсов должен предоставлять абстракцию над файловой системой и являться удобной альтернативой традиционным файловым менеджерам, которая бы позволила работать с информационными ресурсами как с отдельными сущностями. При этом их файловое представление вторично и оно может либо вовсе отсутствовать, либо напротив один информационный ресурс может быть представлен различными файлами или массивами файлов (например, одна и та же книга может быть представлена в различных форматах файлов).

Вторая идея состоит в реализации бесшовного перехода между файловым представлением и семантическим. Дело в том, что на данный момент абсолютное большинство информационных ресурсов, с которыми человек повседневно работает, представлены в виде файлов или массивов файлов, и для того, чтобы предлагаемый семантический менеджер информационных ресурсов действительно стал востребованным пользовательским инструментом необходимо подстраиваться под современные реалии и существующую экосистему и сделать абсолютно прозрачным и легким переход между

файловым представлением и семантическим (а также обратный). Сложность в реализации данной идеи состоит в том, что файл, которым представлен информационный ресурс, как правило, не содержит достаточного количества метаданных для автоматического формирования его семантического описания, и бесшовный переход становится невозможным на практике, поскольку среднестатистический пользователь вручную заполнять семантическое описание не станет.

Третья идея направлена на преодоление этой проблемы и заключается в создании глобального центра-хранилища информационных ресурсов, решающего как одновременно задачу хранения семантического описания бесчисленного количества информационных ресурсов, так и задачу автоматического поиска новых ресурсов и автоматизированного пополнения своей информационной базы.

2.2 Предлагаемая архитектура семантического менеджера информационных ресурсов

Наиболее естественной парадигмой для реализации перечисленных трех идей является классическая клиент-серверная архитектура. В ней на сервере размещается глобальный центр-хранилище информационных ресурсов, а в качестве множества взаимодействующих с ним клиентов выступают семантические менеджеры информационных ресурсов, которые с одной стороны представляет собой альтернативу файловым менеджерам в духе первой из идей концепции, а с другой посредством взаимодействия с сервером реализуется бесшовный переход между файловым представлением информационных ресурсов и семантическим.

В упрощенном виде данная архитектура представлена на компонентной диаграмме далее (рисунок 2.1).

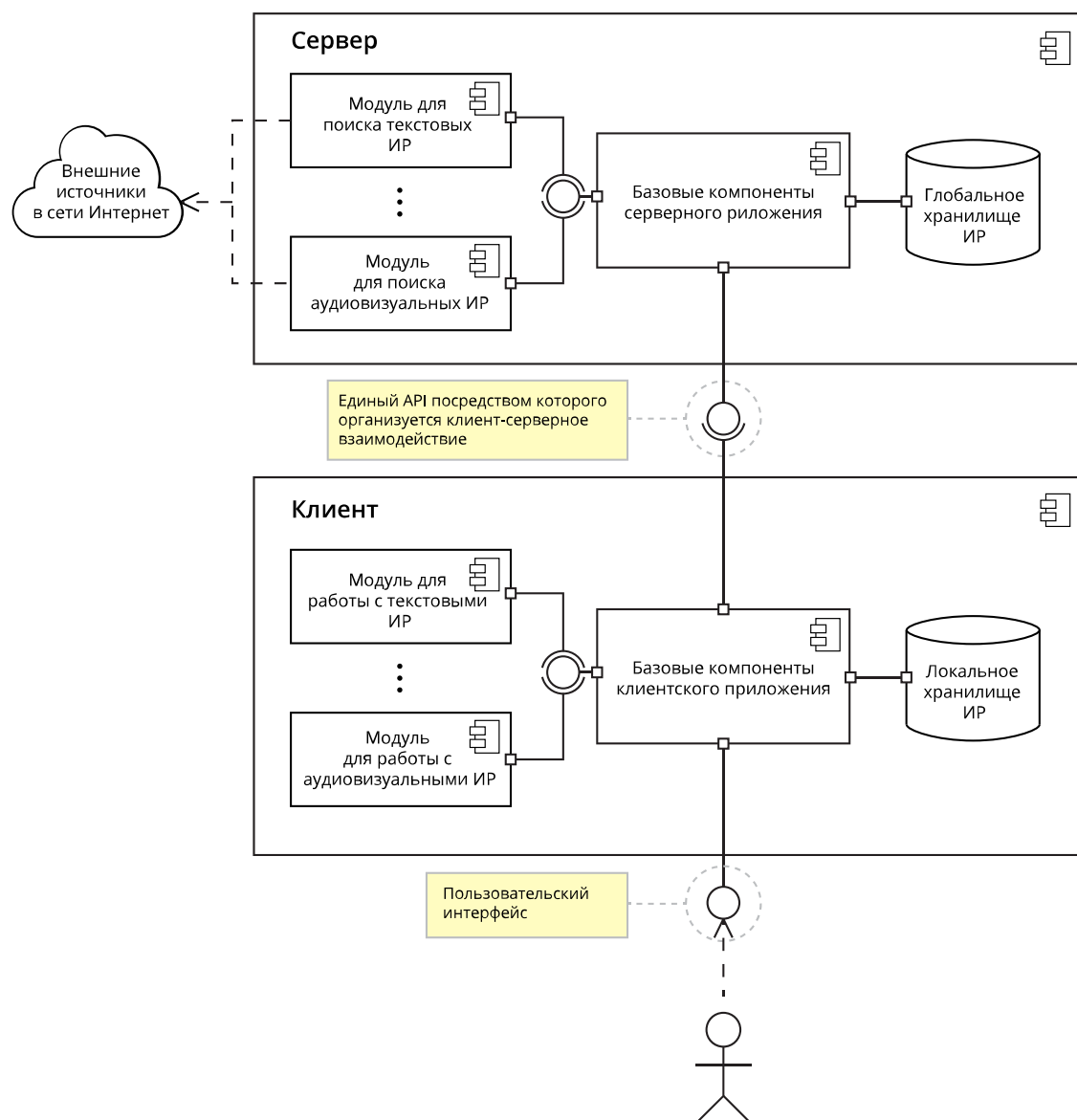


Рисунок 2.1 – Архитектура семантического менеджера информационных ресурсов

2.2.1 Клиентское приложение

В центре решения – клиентское приложение, которое предоставляет высокоуровневую абстракцию над пользовательской файловой системой и позволяет пользователю забыть о необходимости выстраивания иерархий и попытках как-то классифицировать все множество книг, фильмов и прочих информационных ресурсов на своем компьютере, планшете или другом устройстве. Вместо этого оно предлагает сосредоточиться на семантическом

описании, включающем метаданные и связи между информационными ресурсами.

Таким образом, клиентское приложение фактически выступает в качестве альтернативы файловому менеджеру, используемому в современных операционных системах.

С точки зрения внутренней архитектуры клиентское приложение (так же, как и серверное) модульное. Базовый функционал, необходимый для работы со всеми типами информационных ресурсов, реализуется при помощи набора стандартных компонентов, посредством них же организуется взаимодействие между локальным файловым хранилищем, базой данных и клиентским приложением, а также взаимодействие с сервером. Для расширения же возможностей, предоставляемых стандартными компонентами, и реализации специфического для каждого из типов информационных ресурсов функционала используются модули, каждый из которых соответствует определенному типу информационных ресурсов.

Под влиянием модульной архитектуры был выбран и гибкий подход к форме метаданных для описания информационных ресурсов. Вместо заведомо тщетных попыток создания единого формата метаданных для всех возможных типов информационных ресурсов и при этом сохранения удобства работы с ними пользователя, каждый модуль наделен возможностью определения подходящего для того или иного типа информационных ресурсов формата метаданных. Для унификации же и возможности последующего использования семантического описания информационных ресурсов вовне рассматриваемой системы в качестве требования ко всем таким форматам метаданных было выдвинуто требование соответствия одному из наиболее распространенных на сегодняшний день форматов для описания метаданных – Дублинским ядром [13]. Это требование означает, что при разработке нового формата необходимо определить соответствие между элементами описания Дублинского ядра и элементами описания нового формата (например, для формата описания текстового информационного ресурса элементы «Код рубрикатора», «УДК» и

«Ключевые слова» соответствуют такому элементу описания Дублинского ядра как «Тема»).

2.2.2 Серверное приложение

Серверное приложение используется для реализации глобального пространства-хранилища информации, на базе которого также реализуется и бесшовный переход между файловым представлением и семантическим.

То, как реализуется данный переход, определяется конкретным модулем. Как правило, первоначальным этапом идет вычисление значения хеш-суммы файла. Если таковое присутствует в базе данных, содержащей соответствие значений хеш-суммы файлов и информационных ресурсов, то семантическое описание искомого информационного ресурса возвращается клиенту. В противном случае применяются продвинутое методологии по извлечению метаданных из файла и затем полнотекстового поиска по базе информационных ресурсов на основе полученного «слепок». Например, для текстовых информационных ресурсов применяется технология цифрового зрения для извлечения блоков текста и их распознавания, которые затем классифицируются как «заголовок», «автор» и т.д. и используются в качестве параметров поиска по базе информационных ресурсов.

Кроме того, каждый модуль обладает возможностью поиска информационных ресурсов во внешних Интернет-ресурсах, и если полнотекстовой поиск по внутренней базе информационных ресурсов не дал результатов, эта возможность задействуется, и в случае успешного поиска, результат отправляется на клиентское приложение, а база информационных ресурсов на сервере пополняется.

Заключение

В рамках данной практической работы были выполнены одновременно две задачи, с одной стороны был выполнен анализ предметной области, а с другой предложена концепция семантического менеджера информационных ресурсов, сформулированы основные ее положения и представлена архитектура, позволяющая ее реализовать.

Основным предметом анализа стали существующие возможности и решения для каталогизации и семантического описания информационных ресурсов и файлов или массивов файлов, которыми они представлены. Результатом данного анализа в первую очередь стало выявление актуальных потребностей, существующих недостатков, для решения которых и была предложена изложенная концепция.

Предложенная концепция, хотя и имеет определенные недостатки, например, излишнюю централизацию, она, тем не менее, предлагает реальные решения, для рассмотренных проблем и в рамках выпускной квалификационной работы данная концепция будет подробнее раскрыта и рассмотрены ключевые вопросы, связанные с ее эффективной реализацией.

География и глубина

Таблица 2 – География и глубина поиска информационных источников

	1988 -1992	1993 -1997	1998 -2002	2003 -2007	2008 -2012	2013 -2017	Итого
Методики семантического описания ИР и связанные с этим вопросы	2	0	1	2	5	0	10
Машинное обучение, интеллектуальный анализ данных	0	1	1	2	5	2	11
Программная инженерия, архитектура программного обеспечения	0	0	0	2	2	1	5
Цифровое зрение	0	0	0	1	2	8	11
Стандарты и спецификации	0	1	0	3	2	0	6
Итого:	2	2	2	10	16	11	43

Список используемой литературы

1. Ingram D. Insight: A Semantic File System / D. Ingram. Imperial College London, 2008 – 32с.
2. Activity based metadata for semantic desktop search / L/Sauermann, G.A.Grimnes, M.Kiesel и др. // European Semantic Web Conference. – Springer Berlin Heidelberg, 2005 – С. 439-454.
3. Frand J. L., Lippincott A. Personal knowledge management: a strategy for controlling information overload / J. L. Frand, A. Lippincott – Anderson School at UCLA, 2002.
4. Schandl B., Haslhofer B. Files are siles: Extending file systems with semantic annotations //Semantic-Enabled Advancements on the Web: Applications Across Industries. – IGI Global, 2012. – С. 73-93.
5. Ext4 Disk Layout // Ext4 (and Ext2/Ext3) Wiki URL: https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout (дата обращения: 15.05.2017)
6. Gopal B., Manber U. Integrating content-based access mechanisms with hierarchical file systems //OSDI. – 1999. – Т. 99. – С. 265-278.
7. TagSistant: a semantic filesystem for Linux kernels // Официальный сайт URL: <http://home.gna.org/tagfs/index.shtml> (дата обращения: 14.05.2017)
8. Understanding Property Handlers // Центр разработки для Windows URL: <https://msdn.microsoft.com/ru-ru/library/windows/desktop/cc144129.aspx> (дата обращения: 29.12.2015).
9. Tracker // Gnome.org URL: <https://wiki.gnome.org/Projects/Tracker> (дата обращения: 30.12.2015).
10. Griffith // Официальный сайт URL: <http://www.griffith.cc/> (дата обращения: 30.12.2015).
11. Picasa // Официальный сайт URL: <https://picasa.google.com/> (дата обращения: 30.12.2015).
12. Mendeley // Официальный сайт URL: <https://www.mendeley.com/> (дата обращения: 30.12.2015).

13.ГОСТ Р ИСО 15836-2011 Набор элементов метаданных –М.:
Стандартинформ, 2014 – 6 с.