

## **Семантический подход к организации и управлению личными знаниями**

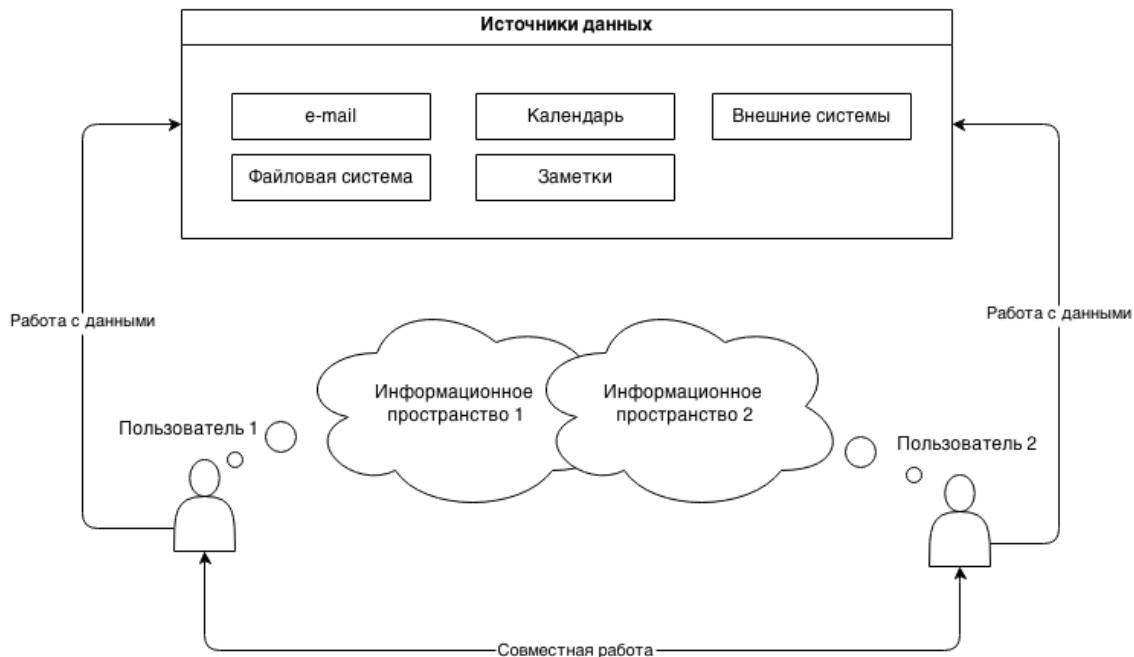
*А.А. Бездушный*

Московский физико-технический институт (государственный университет)

В ходе своей разноплановой деятельности человек регулярно сталкивается с необходимостью изучения новой информации, значительная часть которой хранится в цифровом формате и распределена по различным источникам. В процессе систематизации и изучения информации вырабатывается собственное понимание сведений, формируются связи между новой информацией и изученной ранее. Таким образом человек накапливает личные знания, которые в дальнейшем использует при решении различных задач профессиональной или повседневной деятельности. Со временем часть знаний неизбежно забывается, что порождает необходимость переноса знаний в цифровой формат и организации эффективной работы с ними. Вопросы организации личных знаний и ведения работы с ними рассматриваются в рамках задач управления личной информацией (Personal Information Management) [1], [2] и управления личными знаниями (Personal Knowledge Management) [3], [4].

В рамках данной работы был разработан прототип системы, предоставляющий возможности для работы с личной информацией и знаниями. Прототип является развитием решения, представленного в [5].

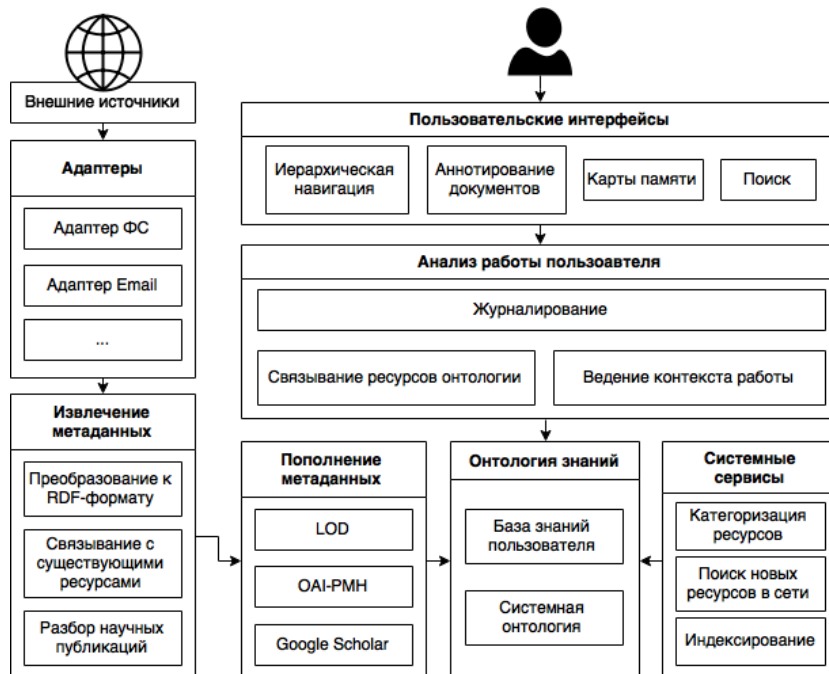
В данной работе приводится описание архитектуры предлагаемого решения. На Рис 1 приведен пример традиционной схемы работы с информацией – информационные пространства нигде формально не определены, только сам человек может определить, какие сведения связаны между собой и как именно. Вследствие этого, все дальнейшие взаимодействия с данными – поиск, совместная работа, структурирование, могут производиться лишь в рамках того приложения, которое отвечает за конкретный тип данных. Поскольку информация делится между различными приложениями, большое количество метаданных о ресурсах, таких как иерархическая организация, связи и зависимости между ресурсами, могут дублироваться в каждом из них.



**Рис 1** Схема работы пользователей без использования системы

Системы управления личной информацией вводят дополнительный уровень организации данных, позволяют пользователю формализовать информационное пространство, предоставляют интерфейсы для работы с ним.

Предлагаемое решение развивает идеи Semantic Desktop, дополняя функции системы управления информацией механизмами, обеспечивающими работу со знаниями. На Рис 2 приведена архитектура системы.



**Рис 2** Архитектура системы МемоPIM

В предлагаемой архитектуре, база знаний пользователя представляет собой RDF-репозиторий, структура которого задается OWL-онтологией. Системная онтология определяет базовые понятия, такие как «Проект», «Персона», «E-mail», пользователь

может расширять системные онтологию добавляя новые понятия или изменяя существующие.

Пополнение базы знаний возможно двумя способами: пользователь может самостоятельно создавать ресурсы с помощью интерфейсов системы, либо настроить автоматический импорт ресурсов из внешних источников (таких как файловая система, почтовый сервер и т.п.). За последнее отвечает уровень *адаптеров к внешним источникам*. Адаптеры в фоновом режиме производят синхронизацию ресурсов базы знаний с данными, созданными во внешних источниках. Это позволяет пользователям использовать для работы привычные приложения. Сведения из внешних источников приводятся к RDF формату, кроме того для некоторых типов данных производится дополнительный анализ импортированного ресурса. Например, из текста научной публикации выделяются название, авторы, список литературы.

После импорта информации в базу знаний, системой выполняется автоматическое пополнение метаданных сведениями из внешних источников. Например, после автоматического выделения метаданных из текста научной публикации, на основании названия публикации и авторов, осуществляется поиск полной библиографической информации, такой как год публикации, издательство и др., в глобальной сети.

Помимо информации, изначально хранящейся в цифровом формате (электронные документы, email-сообщения), система поддерживает работу с личными знаниями человека. Пополняя сведения, выгруженные из внешних источников дополнительными метаданными, и проставляя связи между ресурсами, пользователь формирует цифровое представление личных знаний.

Для работы с базой знаний системой предоставляются иерархические и графические пользовательские интерфейсы. Интерфейс аннотирования позволяет размечать и комментировать текстовое содержимое ресурсов базы знаний. Интерфейс поиска позволяет осуществлять полнотекстовый поиск по базе знаний, а также поиск по ресурсам с учетом их структуры.

По мере работы пользователя, системой ведется журналирование выполняемых им действий. Записи из журналов в дальнейшем используются для ведения контекста пользователя и поиска новых связей между ресурсами. Более подробно эти модули рассмотрены в следующих разделах.

Системные сервисы производят фоновый анализ базы знаний пользователя. Сервис автоматизированной категоризации отвечает за предоставление пользователю рекомендации по присвоению ресурсам тегов. Сервис поиска новых ресурсов в сети отвечает за предоставление пользователю потенциально интересных ему ресурсов, на основании анализа тех, которые хранятся в базе знаний. Например, на основании

имеющихся в базе знаний публикаций, в соответствии с рекомендациями Google Scholar, системой предлагаются другие публикации, потенциально интересные пользователю. Сервис индексирования отвечает за ведение индексов, необходимых для эффективного поиска информации в базе знаний.

## 1 База знаний пользователя

База знаний – это совокупностью всех сведений, с которыми человек работает сейчас или работал ранее. Любая информация имеющая важность для него, и выделяемая им в отдельную сущность, рассматривается как часть базы знаний. Ресурсами базы знаний могут быть файлы, заметки, ранее посещенные веб-страницы, письма и пр.

С помощью онтологий можно эффективно представлять сложные знания, поэтому, в рамках системы, в качестве ресурсов базы знаний, рассматриваются RDF-ресурсы. Таким образом, каждый пользователь ведет работу с собственным RDF-репозиторием данных. Структура базы знаний задается системной онтологией и ее пользовательскими расширениями. Системная онтология, вводит ряд понятий, наиболее часто используемых пользователями при работе с личной информацией и знаниями, например, Project, Person, Publication, File, Email и др. При необходимости, пользователь может расширять стандартную онтологию, добавляя в нее новые классы и свойства или переопределяя уже имеющиеся. Наполнение базы знаний происходит либо путем автоматического импорта информации, либо вручную, через пользовательский интерфейс системы.

Опишем формальную модель представления знаний. Введем следующие обозначения:

- $U$  – множество URI идентификаторов;
- $L$  – множество допустимых литералов;
- $2^L$  – множество всех подмножеств множества допустимых литералов.

Тогда онтология может быть представлена следующим кортежем:

$$O = \{C, P, I, O^L, A\},$$

где

- $C \subset U$  – множество концептов (классов) предметной области, которую представляет онтология  $O$ ;
- $P \subset U$  – множество отношений, соответствующих классам заданной предметной области. Отношения делятся на объектные (ObjectProperty) – связывают различные классы предметной области, и литеральные (DataProperty) – указывают на наличие связи между классом и литералом;
- $I \subset U$  – множество индивидов, представляющих объекты предметной области;
- $O^L \subset 2^L$  – множество литералов, используемых при определении индивидов;

- $A$  – множество аксиом, заданных на концептах, отношениях и индивидах онтологии  $O$ .

**Определение 1.** Будем говорить, что онтология  $O_2 = \{C_2, P_2, I_2, O_2^L, A_2\}$ , является простым расширением онтологии  $O_1 = \{C_1, P_1, I_1, O_1^L, A_1\}$ , если:

- $C_1 \subset C_2, P_1 \subset P_2, I_1 \subset I_2, O_1^L \subset O_2^L, A_1 \subset A_2$
- $\forall c_2 \in C_2 \setminus C_1 : c_2$  является простым определением класса -  $Class(*)$
- $\forall a_2 \in A_2 \setminus A_1 : a_2$  является реализацией одной из следующих аксиом  $\{SubClassOf, SubObjectPropertyOf, ObjectPropertyDomain, DataPropertyDomain, ClassAssertion, ObjectPropertyAssertion, DataPropertyAssertion\}$

Таким образом мы рассматриваем такие расширения онтологий, которые включают в себя определения простых новых классов или подклассов, а также расширяют область применения отношений и определяют новые экземпляры классов.

**Определение 2.** Обозначим  $O_S$  – системную онтологию, более подробно понятия, описываемые этой онтологией, представлены в разделе 6

**Определение 3.** Базой знаний пользователя будем называть онтологию  $O_U$ , представляющую собой простое расширение онтологии  $O_S$ .

Ограничения на доступные способы расширения системной онтологии накладываются с целью обеспечения простоты и интуитивной понятности работы с системой. Язык OWL является довольно тяжелым для изучения, поэтому использование сложных конструкций может привести к трудностям при работе с системой у пользователей, не обладающих достаточным знанием принципов OWL.

## 2 Адаптеры к внешним источникам

Внешними источниками для пополнения базы знаний пользователя являются файловая система (ФС), почтовый сервер и другие приложения, которые использует пользователь. Для каждого внешнего источника должен быть определен адаптер. По умолчанию, системой предоставляются адаптеры к файловой системе и почтовому серверу. В фоновом режиме адаптеры производят синхронизацию ресурсов базы знаний с внешними источниками.

После выгрузки информации из внешних источников она приводится к RDF-формату, в соответствии с системной онтологией. Далее, загруженные данные связываются с существующими ресурсами, например, отправитель письма ассоциируется с ресурсом, представляющим соответствующую персону в базе знаний. После этого, для некоторых

типов ресурсов из их содержимого, выделяются дополнительные метаданные. Например, из текста научной публикации выделяются название, авторы, список литературы.

Приведем формальное описание адаптеров к внешним источникам. Пусть  $\{E_i\}$  – множество внешних источников,  $ER_i$  – множество ресурсов доступных для загрузки из источника  $E_i$ .

**Определение 4.** Онтологию  $O_2 = \{C_2, P_2, I_2, O_2^L, A_2\}$  будем называть пополнением онтологии  $O_1 = \{C_1, P_1, I_1, O_1^L, A_1\}$ , если:

- $C_1 \equiv C_2, P_1 \equiv P_2, I_1 \subset I_2, O_1^L \subset O_2^L, A_1 \subset A_2$
- $\forall a_2 \in A_2 \setminus A_1 : a_2$  является реализацией одной из следующих аксиом  $\{ClassAssertion, ObjectPropertyAssertion, DataPropertyAssertion\}$

Т.е. при пополнении онтологии могут быть добавлены новые индивиды или изменены существующие, изменений в схему данных не вносится.

**Определение 5.** Адаптером к внешнему источнику будем назвать функцию:

$$EA_i(er_i, O_U^1) : ER_i \rightarrow O_U^2;$$

Где  $O_U^2$  – пополнение онтологии  $O_U^1$ ,  $er_i \in ER_i$ . Т.е. адаптеры производят отображение внешних ресурсов в индивиды базы знаний пользователя.

### 3 Пополнение метаданных

Следующим этапом, после импорта информации из внешних источников, является ее пополнение дополнительными метаданными. В качестве источников метаданных используются различные интернет ресурсы, предоставляющие открытый доступ к данным. Например, после автоматического выделения метаданных из текста научной публикации, на основании названия публикации, осуществляется поиск полной библиографической информации, такой как год публикации, издательство и др. В случае успеха, найденная информация переводится в RDF-формат и сохраняется в базе знаний пользователя. Более детально модуль, реализующий такое пополнение, описан в работе [6].

### 4 Системные сервисы

Системой предоставляется набор сервисов, осуществляющих анализ и пополнение базы знаний пользователя.

#### 4.1 Сервис анализа работы пользователя

По мере работы пользователя в системе, осуществляется автоматизированный сбор данных о выполняемых им действиях. Записи из журналов работы пользователя в

дальнейшем используются для анализа работы пользователя с системой – ведения контекста работы и поиска новых связей между ресурсами базы знаний.

Пусть  $E$  – множество журналируемых действий пользователя, например – открытие ресурса, поиск ресурсов. Данное множество определяется системной онтологией.

**Определение 6.** Действие пользователя может быть представлено кортежем

$$ua = \{R, e, t\},$$

где:

$R \subset U$  – множество ресурсов над которыми выполняется действие,

$e \in E$  – тип выполненного действия,

$t \in \mathbb{R}$  – время выполнения действия.

## 4.2 Сервис ведения контекста работы

Системой управления личной информацией и знаниями осуществляется поддержка пользователя при работе с разнородными данными, обеспечивается возможность быстрого доступа к нужной информации в нужный момент времени. По мере выполнения действий, формируется группа ресурсов, с которыми пользователь ведет работу чаще других в течении некоторого промежутка времени. Такие ресурсы представляют собой *текущий контекст работы пользователя* – наиболее актуальные для него сведения на конкретный момент времени. Системой обеспечивается хранение текущего контекста и упрощение доступа с ресурсами, входящими в него.

## 4.3 Сервис связывание ресурсов

Важным требованием для обеспечения эффективной ассоциативной навигации по базе знаний, является наличие связей между ресурсами. Часть связей может быть задана пользователем вручную, но для поддержки более тесной связанности, часть связей может быть сгенерирована автоматически. В модуле связывания ресурсов осуществляется расчет схожести между ресурсами базы знаний. Вычисленные значения схожести в дальнейшем используются при формировании перечня ресурсов для ассоциативной навигации. При осуществлении поиска пользователь помимо фильтрации по содержанию ресурса может задавать список ресурсов, связанных с ним. Например, «найти все письма, связанные с событием «Конференция», содержащие фразу «управление знаниями»» может найти некоторый ресурс с помощью атрибутивного поиска, и после этого продолжить навигацию по базе знаний с помощью предлагаемых системой ассоциаций.

## 4.4 Сервис автоматической категоризация

Традиционно выделяют два основных подхода к категоризации ресурсов – с помощью иерархических структур и с помощью тегов. Разрабатываемая система

объединяет оба вида представлений, позволяя создавать иерархические связи между тегами. По мере наполнения базы знаний пользователя ресурсами и соответствующими им тегами, появляется возможность проставлять теги к новым ресурсам автоматически. Эту задачу можно рассматривать как задачу рекомендации возможных тегов для ресурсов.

При подборе тегов для нового документа теги выбираются на основании анализа текстового содержимого категоризированных ресурсов. Для каждого слова в новом документе определяется набор тегов, с которыми это слово встречается чаще. Влияние слова на итоговый рейтинг тега складывается из двух факторов:

1. насколько это слово характеризует ресурс;
2. насколько часто слово встречается в ресурсах, помеченных тегом, а также насколько оно характеризует эти ресурсы.

#### **4.5 Поиск новых ресурсов в сети**

Поскольку информация в системе представлена в семантически богатом виде, для ряда ресурсов базы знаний системой могут автоматически формироваться рекомендации других, потенциально интересных пользователю ресурсов из глобальной сети. Например, на основании имеющихся в базе знаний пользователя научных публикациях, в соответствии с рекомендациями сервиса Google Scholar, системой предлагаются другие статьи, потенциально интересные пользователю.

#### **4.6 Сервис индексирования**

Сервис индексирования отвечает за ведение индексов, необходимых для эффективного поиска по базе знаний. Индекс ресурса состоит из записей о вхождении термов в текстовые атрибуты ресурса. Под термом понимается некоторая строка, которая является неделимой, и входит в состав значения атрибута. Разбиение фраз на термы необходимо для обеспечения возможности полнотекстового поиска по базе знаний.

### **5 Пользовательские интерфейсы**

Для работы с базой знаний системой предоставляются иерархические и графические пользовательские интерфейсы. В режиме иерархического представления пользователю выводится дерево ресурсов, соединенных иерархическими связями. Просмотр полной информации о ресурсе доступен с помощью его детального представления. В графическом интерфейсе работы, пользователь может создавать интеллект-карты, узлами которых являются ресурсы базы знаний.

Интерфейс аннотирования позволяет размечать и комментировать текстовое содержимое ресурсов базы знаний. Выделенные фрагменты заносятся в базу знаний и становятся доступны для дальнейшего использования, например при работе с интеллект-картами.



Интерфейс поиска позволяет осуществлять полнотекстовый поиск по базе знаний, поиск по ресурсам с учетом их структуры и поиск с учетом контекста. В режиме поиска с учетом контекста пользователю выводятся дополнительные фильтры, позволяющие указать ресурсы, которые входят в контекст искомого ресурса, т.е. использовались приблизительно в одно и то же время с искомым.

## 6 Онтология управления информацией и знаниями

Системная онтология описывает основные понятия и классы, используемые различными модулями системы. Как части системной онтологии выделяются *онтология верхнего уровня*, описывающая понятия, атрибуты и отношения общие для всех ресурсов базы знаний, и набор *онтологий предметных областей*, описывающих понятия, используемые в различных модулях системы.

При разработке онтологии использовались понятия из следующих публичных онтологий: Simple Knowledge Organization System (SKOS)<sup>1</sup>, Dublin Core Terms (DCTERMS)<sup>2</sup>, Friend of a Friend (FOAF)<sup>3</sup>, NEPOMUK Information Element Ontology (NIE)<sup>4</sup>, NEPOMUK File Ontology (NFO)<sup>5</sup>, NEPOMUK Contact Ontology (NCO)<sup>6</sup>, NEPOMUK Message Ontology (NMO)<sup>7</sup>, NEPOMUK Calendar Ontology (NCAL)<sup>8</sup>, Personal Information Model Ontology (PIMO)<sup>9</sup>, Task Model Ontology (TMO)<sup>10</sup>, FRBR-aligned Bibliographic Ontology (FaBiO)<sup>11</sup>, Citation Typing Ontology (CiTO)<sup>12</sup>, User Interaction Context Ontology (UICO) [7], Semantic Document Model (SDM) [8]. На Рис 3 представлены ключевые классы системной онтологии и связи между ними. Сплошными линиями обозначено наследование, пунктирными – связи между классами.

---

<sup>1</sup> <http://www.w3.org/TR/skos-reference/>

<sup>2</sup> <http://dublincore.org/documents/dcmi-terms/>

<sup>3</sup> <http://xmlns.com/foaf/spec/>

<sup>4</sup> <http://www.semanticdesktop.org/ontologies/2007/01/19/nie>

<sup>5</sup> <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo>

<sup>6</sup> <http://www.semanticdesktop.org/ontologies/2007/03/22/nco>

<sup>7</sup> <http://www.semanticdesktop.org/ontologies/2007/03/22/nmo>

<sup>8</sup> <http://www.semanticdesktop.org/ontologies/2007/04/02/ncal>

<sup>9</sup> <http://www.semanticdesktop.org/ontologies/2007/11/01/pimo/>

<sup>10</sup> <http://www.semanticdesktop.org/ontologies/2008/05/20/tmo/>

<sup>11</sup> <http://purl.org/spar/fabio>

<sup>12</sup> <http://purl.org/spar/cito>

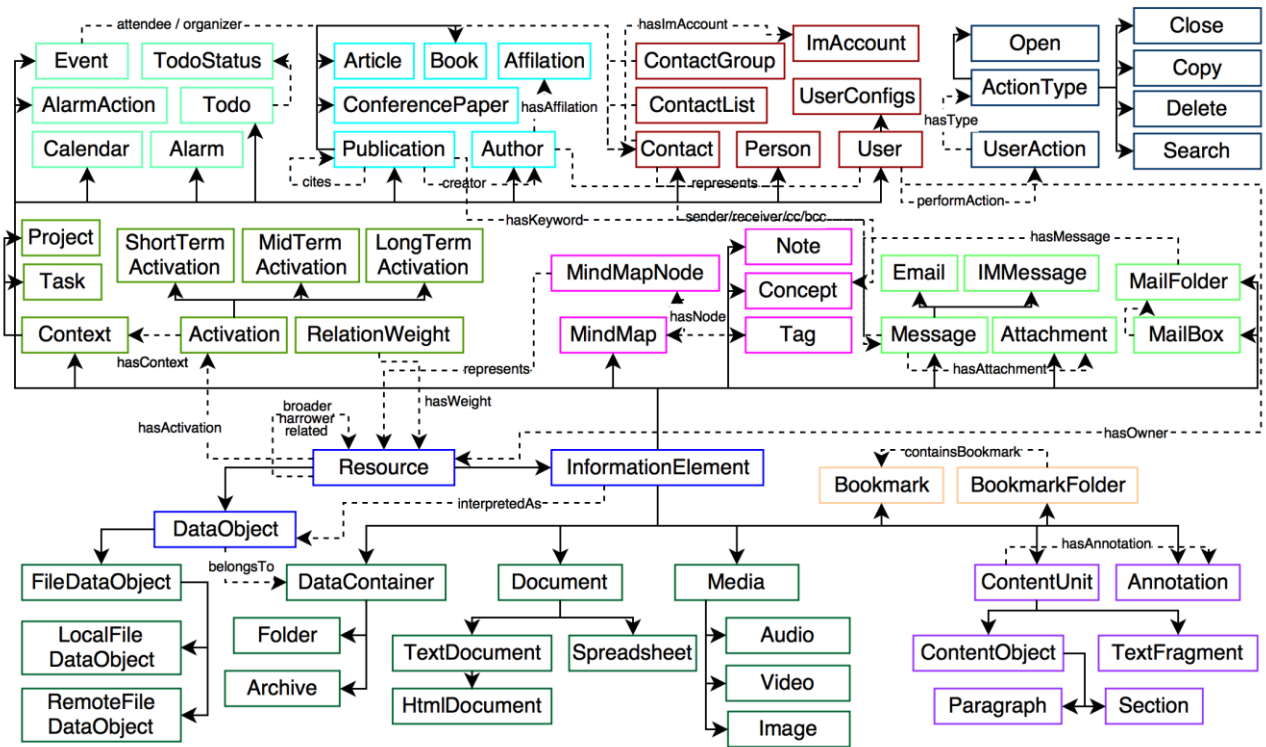


Рис 3 Системная онтология

Рассмотрим более подробно каждую из онтологий предметных областей.

### 6.1 Онтология верхнего уровня

Данная онтология описывает понятия, общие для всех ресурсов базы знаний. Ключевым классом онтологии является класс «Resource», который используется в качестве родительского для большинства классов в онтологиях предметных областей. Класс определяет наиболее общие атрибуты, такие как название ресурса, описание, дата создания, дата изменения и другие. Также в рамках данного класса определяются отношения «broader», «narrower» и «related», с помощью которых в базе знаний задаются иерархические и ассоциативные связи между ресурсами. Помимо класса «Resource», в онтологии верхнего уровня определяются классы «InformationElement» и «DataObject». Класс «DataObject» используется для представления сущностей, непосредственно содержащих данные, например файлов, приложений к email-сообщениям. Класс «InformationElement» используется для описания информации, содержащейся внутри экземпляров «DataObject», он связан с классом «DataObject» с помощью отношения «interpretedAs». Т.е. класс «DataObject» рассматривает ресурсы как массив байт, в то время как «InformationElement» определяет семантику этого ресурса.

В качестве примера применения этих классов, рассмотрим файл «статья.pdf», хранящийся где-то на компьютере пользователя. Для этого файла в базе знаний содержатся два объекта. Первый – экземпляр класса DataObject, представляющий файл как область данных на файловой системе. Данный объект содержит такие атрибуты как размер файла,

его расположение на файловой системе, и т.п. Второй объект – экземпляр класса InformationElement, представляющий файл как научную публикацию, данный объект содержит атрибуты характерные для научной статей, например год издания, список авторов и др.

## **6.2 Онтология файловой системы**

Онтология файловой системы описывает основные типы данных, хранящиеся на персональном компьютере пользователя. Класс «FileDataObject» наследуется от класса «DataObject» и используется для описания любых файлов, находящихся на файловой системе. В рамках данного класса файл рассматривается исключительно как набор байт, без какой-либо семантики. Информация, содержащаяся в файле, описывается с помощью классов «DataContainer», «Document», «Media», которые представляют каталоги файловой системы, текстовые документы или различные медиа ресурсы (аудио, видео, изображения).

## **6.3 Онтология знаний**

Онтология знаний определяет основные понятия, с помощью которых можно напрямую или косвенно формализовать личные знания. Для записи знаний в форме интеллект-карты используются классы «MindMap» и «MindMapNode», представляющие саму интеллект-карту и конкретные вершины в ней соответственно. Класс «MindMapNode» связан с классом «Resource» отношением «representsResource». Помимо интеллект-карт, для представления знаний могут быть использованы классы «Tag», «Concept», «Note», «Keyword», описывающие теги, понятия, заметки и ключевые слова соответственно.

## **6.4 Онтология научных трудов**

На данный момент разработано большое количество онтологий, описывающих научные труды и связанные с ними понятия. В данной работе используются онтологии FRBR-aligned Bibliographic Ontology (FaBiO) – для описания научных трудов и Citation Typing Ontology (CiTO) – для описания библиографической информации. На схеме приведены некоторые классы этой онтологии. Класс «Publication» представляет базовый класс научных трудов, а его специализации «Article», «ConferencePaper» и «Book» конкретизируют тип публикации. Для указания зависимостей цитирования использует отношение «cites» из онтологии CiTO.

## 6.5 Онтология контекста работы

Данная онтология определяет классы, используемые в задаче ведения контекста работы пользователя. Онтология описывает основные понятия, необходимые для применения механизма распространения активации (описанном в разделе **Ошибка! Источник ссылки не найден.**). Класс Context представляет контекст работы пользователя, контекстом может быть проект («Project») или задача («Task»). В рамках некоторого контекста, каждому ресурсу может быть проставлено значение *активации* («Activation»). Активация делится на краткосрочную («ShortTermActivation»), среднесрочную («MiddleTermActivation») и долгосрочную («LongTermActivation»). Класс «RelationWeight» позволяет задать вес связи, используемый при расчете активации.

## 6.6 Онтология действий пользователя

Онтология действий пользователя описывает множество журналируемых типов пользовательских действий. Системой поддерживаются следующие типы действий – открытие, закрытие, копирование, удаление и добавление ресурсов, а также поиск ресурсов.

## 6.7 Онтология аннотирования документов

Данная онтология вводит понятия, используемые при создании пользовательских аннотаций к текстовым документам. Базовым понятием онтологии, описывающим некоторую часть содержимого документа является класс «ContentUnit». Подклассы данного понятия конкретизируют, какая именно часть содержимого описывается. Классы «Paragraph» и «Section» представляют параграф или раздел документа, а класс «TextFragment» - фрагмент текста из произвольной части документа. Класс «Annotation» представляет факт аннотирования некоторого фрагмента документа, он связан с классом «ContentUnit» отношением «hasAnnotation».

## 6.8 Онтология сообщений

Данная онтология описывает понятия, связанные с отправкой и получением сообщений, таких как сообщения электронной почты или сообщения онлайн мессенджеров. В общем случае сообщение представляется классом «Message». Наследуемые от него классы «Email» и «IMMessage» конкретизируют специфику сообщения. Сообщение может иметь несколько получателей из числа экземпляров класса «Contact», определенного в онтологии контактов. Классы «MailBox» и «MailFolder» используются для представления почтового ящика и папки в почтовом ящике соответственно.

## 6.9 Онтология контактов

В данной онтологии определяются классы, описывающие пользователей системы, персон из списка контактов и связанные с ними понятия. Персону представляет класс «Person». Персона может быть связана с пользователем системы «User» с помощью отношения «representsPerson». Специализацией класса «Person» является класс «Contact», представляющий элемент списка контактов «ContactList».

## 6.10 Онтология календаря

Онтология описывает понятия, связанные с событиями календаря («Event»), напоминаниями («Alarm»), а также с ведением списка задач («Todo»). У события может быть назначен определен организатор («Organizer») и коллекция участников («Attendee»).

## Литература

1. Sauermann L., Grimnes G.A., Kiesel M. et al. Semantic desktop 2.0: The gnows is experience // The Semantic Web - ISWC 2006. 2006. pp. 887-900.
2. Dong X. L., Halevy A. A platform for personal information management and integration // Proceedings of Conference on Innovative Data Systems Research (CIDR). Asilomar, CA, USA. January 4-7, 2005. pp. 119-130.
3. H. Maurer. The heart of the problem: Knowledge management and knowledge transfer // Proceedings of ENABLE'99. Espoo, Finland. 2-5 June, 1999. Vol. 99. pp. 8-11.
4. Völkel M., Haller H. Conceptual data structures for personal knowledge management // Online Information Review, Vol. 33, No. 2, 2009. pp. 298-315.
5. Бездушный А.А. Концептуальные положения и архитектура системы семантического управления личной информацией // Программная инженерия, Т. 9, 2014. С. 30-37.
6. Бездушный А.А. Организация и управление личными каталогами научных публикаций с использованием технологий Semantic Web // XV Российская конференция с участием иностранных ученых «Распределенные информационные и вычислительные ресурсы» - DICR-2014. Новосибирск, Россия. 2014.
7. Devaurs D., Rath A. S., Lindstaedt S. N. Exploiting the user interaction context for automatic task detection // Applied Artificial Intelligence, Vol. 26, No. 1-2, 2012. pp. 58-80.
8. Nešić S. Semantic document model to enhance data and knowledge interoperability // Web 2.0 & Semantic Web, 2009. pp. 135-160.