

# Содержание

<b>Введение</b> . . . . .	2
<b>Глава 1. Семантическая сеть</b> . . . . .	4
1.1. Введение в семантические сети . . . . .	4
1.2. OWL . . . . .	6
1.3. SPARQL . . . . .	6
<b>Глава 2. Сравнение RDF с другими моделями хранилищ</b> .	8
2.1. Реляционные хранилища . . . . .	8

# Введение

Достижения в области биологических наук наряду с нарастанием объемов доступной для использования информации повышают необходимость в интеграции разрозненных источников данных. К ним относятся и интернет, и ведомственные, и корпоративные системы:

1. Медстатистика
2. Результаты клинических испытаний
3. Электронная история болезни
4. Фармацевтика
5. Разработка лекарственных средств

В настоящее время в России разработано и используется большое количество разрозненных медицинских информационных систем, разнообразных баз данных с описанием лекарственных препаратов, результатами научных исследований; написано множество научных трудов и статей в области здравоохранения (медицина, фармацевтика, медицинское страхование и др.), хранящихся в специализированных электронных библиотеках в различных форматах. Однако эффективные механизмы извлечения из таких источников знаний, хранения и предоставления к ним широкого доступа отсутствуют.

Требуется фундаментальный сдвиг от единичных попыток интеграции к единой функциональной области. Для решения этой проблемы разработан стандарт публикации данных Linked Data. Одним из важнейших достоинств этой технологии является ее открытость - возможность объединения в общую семантическую сеть распределенных семантических хранилищ,

созданных различными организациями (органы управления здравоохранением, ВУЗы, НИИ, МО) и профессиональными сообществами (ассоциации кардиологов, анестезиологов, медицинских IT-специалистов и др.) на основе единых открытых стандартов. Как показывает международный опыт, это позволяет системе саморазвиваться, постоянно пополняя количество доступных знаний и повышая их качество.

Межресурсные ссылки дают исследователям возможность перемещаться между источниками данных и открывать связи, которые не были замечены ранее. Существуют универсальные инструменты, такие как семантические веб-браузеры и поисковые движки, которые могут использоваться для задач представления и поиска данных.

Основная цель моей работы - это создание семантического хранилища медицинских знаний.

Для достижения поставленной цели решались следующие задачи:

1. Скачивание и парсинг информации с ресурса Webapteka
2. Разработка онтологии лекарственных препаратов
3. Конвертация html данных в rdf представление
4. Разработка SPARQL-запросов для извлечения информации и выявления дополнительных связей в RDF-хранилище.
5. Разработка пользовательского интерфейса
6. Кеширование элементов приложения для повышения производительности

# Глава 1

## Семантическая сеть

### 1.1. Введение в семантические сети

Семантическая сеть (англ. Semantic Web) — это набор технологий, позволяющих представлять информацию в виде пригодном для машинной обработки: RDF, OWL, SPARQL. RDF используется для представления информации, SPARQL - для доступа к ней, OWL - добавляет метаинформацию, связи между концептами.

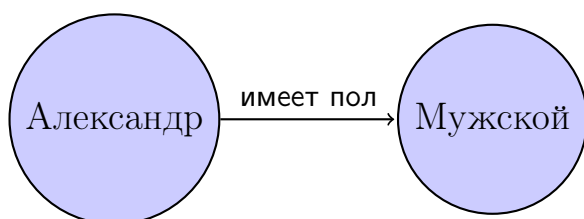
В RDF вся информация представляется в виде триплетов: субъект, предикат, объект. Триплеты по форме похожи на простое предложение. Например:

Субъект: Александр

Предикат: Имеет пол

Объект: Мужской

Триплет может быть выражен в виде графа

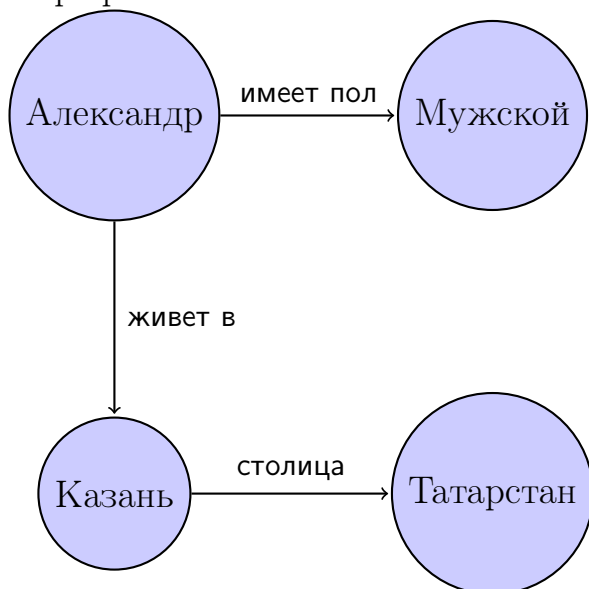


Субъекты и объекты могут быть представлены URI, либо литералом. URI - это уникальный идентификатор, который обозначает сущность: например URI для собаки может быть таким *'http://example.ru/animals/dog'*. Литерал - это просто строка, например *'Jack Nickolson'*, с возможными добавлениями, указывающими язык, тип данных (поддерживаемый XML, такие как integer и datetime). В идеале каждая между сущностями и URI

составлено взаимно однозначное соответствие: каждый URI принадлежит только одной сущности и каждая сущность имеет только один URI. Для обозначения предикатов всегда используются URI.

Использование URI в RDF облегчает нахождение документов, связанных с сущностью. Например, если кто-то (или чья-то программа) ищет информацию о собаках, то ему надо искать все триплеты содержащие URI *<http://www.example.ru/animals/dog>*.

RDF-документ представляет собой набор триплетов. Его можно выразить в виде графа, если представить URI как вершины, а предикаты как ребра графа.



Таким образом можно построить граф неограниченного размера. RDF как язык для хранилища имеет ряд преимуществ:

1. Простое агрегирование данных. Необходимо только добавить триплеты с указанием связи между сущностями.
2. Использование URI дает возможность объединять информацию о сущности с нескольких источников данных.
3. Поскольку RDF не имеет жестких, заведомо заданных требований к структуре данных, к наличию или отсутствию свойства, повышается

плотность хранения информации.

4. RDF предлагает единый язык для представления практически любого знания.

## 1.2. OWL

Технологии Semantic Web дают возможность выводить новые факты из базовых фактов, хранящихся в RDF. OWL добавляет к RDF информацию о классах, типах, логических зависимостях, доменов у свойств, пространстве возможных значений.

## 1.3. SPARQL

SPARQL - язык запросов к RDF-хранилищу. SPARQL, как и большинство языков такого типа, содержит переменные в тексте запроса, в которые подставляются извлеченные данные. Запрос вида

```
SELECT ?x WHERE {  
  ?x <http://www.example.com/has-gender> <http://www.example.com/male> .  
}
```

найдет все триплеты с указанным предикатом и объектом(ИМЕЕТ ПОЛ, МУЖСКОЙ) и вернет список субъектов. Информация возвращается в XML-формате.

Запрос может быть построен из нескольких триплетов. В следующем примере кода две конструкции, которые должны вернуть всех людей мужского пола.

```
SELECT ?x WHERE {  
  ?x <http://www.example.com/has-gender> <http://www.example.com/male> .  
  ?x <http://www.example.com/has-species> <http://www.example.com/human> .  
}
```

Это тип запросов основной в SPARQL. Хотя SPARQL беднее по функциональности чем SQL, он поддерживает схожий функционал для уточнения запроса: сортировка результатов, получение подмножества результатов, удаление дубликатов и т.д. Следующий запрос вернет всех мужчин, которые имеют больше, чем 20 книг и, если имеется информация о предпочтениях в еде, она вернется тоже.

```
PREFIX ex: <http://www.example.com/>
SELECT ?x ?foods WHERE {
  ?x ex:has-gender ex:male .
  ?x ex:has-species ex:human .
  ?x ex:has-book-count ?bookcount .
  FILTER (?bookcount < 20)
}
OPTIONAL {
  ?x ex:likes-food ?foods .
}
}
```

Преимущества, которые могут быть получены за счет использования этого языка запросов понятны: человек или компьютер могут соединиться с любым открытым репозиторием, сделать очень специфичный запрос и получить машино-обрабатываемые данные.

## Глава 2

# Сравнение RDF с другими моделями хранилищ

В любом хранилище данных, доступ к информации осуществляется в соответствии с некоторой моделью, логической концепцией. В этой главе описываются модели хранения данных, используемые в настоящее время. Исследуются сходства RDF-модели с остальными и определяется, в какой степени подходы для традиционных баз данных применимы к RDF-хранилищам.

### 2.1. Key-Value хранилища

### 2.2. Реляционные хранилища

Эта модель была предложена в 1970г. Э.Коддом. В этом подходе теория множеств и логика предикатов используются для определения логической структуры хранилища данных и операций, которые могут быть к нему применены. В частности, разделяются логическая структура и физическая. СУБД может выбрать любой способ физического хранения данных, но то, как информация отображается пользователю, остается неизменным.

Реляционная модель описывает данные в терминах реляций, состоящих из неограниченного числа кортежей и атрибутов. Реляции в целом схожи с таблицами, состоящих из строк и столбцов. Каждый кортеж является уникальным (ведь не имеет смысла один и тот же факт дважды). Запросы к СУБД пишутся на декларативном языке, позволяющим пользователям указать, какие данные они хотят получить, не заставляя их указы-



вать, каким способом это сделать. Как правило, это ответственность СУБД сделать запрос как можно более быстрым. Компонент, который выполняет оптимизацию, называется оптимизатором запросов.

Реляционная модель предназначена для поддержки запросов с перекрестными ссылками между блоками данных: *'Получить всех механиков, которые работали над машиной, содержащей деталь X.'*

## 2.3. RDF хранилище