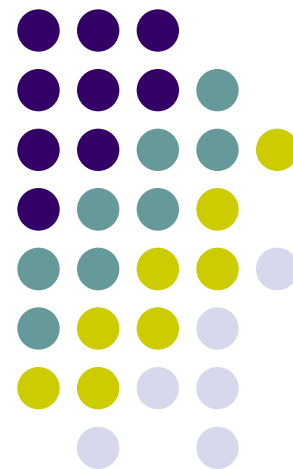


**ПРЕДСТАВЛЕНИЕ
ЗНАНИЙ:**

**ФОРМАЛЬНЫЕ ЯЗЫКИ
для
Semantic Web**

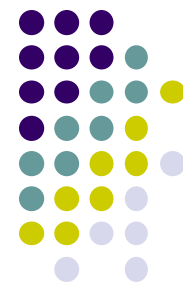


СОДЕРЖАНИЕ



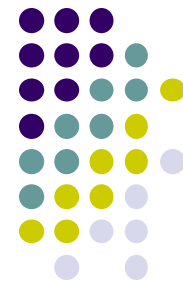
1. Концепции *Semantic Web* и *Linked Data*
2. Понятие *URI*
3. Форматы *RDF*
4. Средства *RDFS* : словари
5. Язык запросов *SPARQL*
6. Язык описания онтологий *OWL*
7. Заключение

ПОИСК И СЕМАНТИЧЕСКИЙ ВЕБ



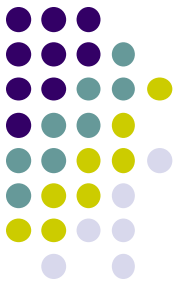
- Развитие Интернета:
 - Web-1 – 1989 г. – веб документов
 - Web-2 – 2000 гг. – структурированный и социализированный веб
 - Web-3 – 2020г.? – семантический веб
- Компьютерная программа не способна, загрузив произвольный документ (веб-страница, файл) понять его содержание
- Целью проекта *Semantic Web* является создание средств:
 - снабжающих ресурсы сети Интернет **метаинформацией**
 - позволяющих пользователю осуществлять **семантическую навигацию и поиск** внутри аннотированных ресурсов
 - обеспечивающих поиск и интеграцию информации программными агентами
- В идеальном варианте, все информационные ресурсы в Интернете должны описываться на двух языках: для человека и для программных агентов

Semantic Web и *Linked Data*



- *Semantic Web* – расширение *Web*, в рамках которого информация представляется в форматах, обеспечивающих семантический поиск на базе аннотированных (метаинформацией) ресурсов Интернет
- *Semantic Web* позволит пользователю быстро получать ответы на сложные запросы
- «Онтологии – ведущая парадигма структурирования информационного контента»
- Основа *Semantic Web* – концепция *Linked Data*, 2006 г., Тим Бернес-Ли
 - Представление данных в структурированном виде, пригодном для машинной обработки (семантические сети и др.)
 - Набор приемов опубликования и соединения структурированных данных
 - Основные понятия: *HTTP URI*, *RDF*, *SPARQL*

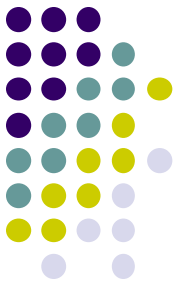
Web и Linked Data: СРАВНЕНИЕ



<i>World Wide Web</i>	<i>Linked Data</i>
Данные для людей - документы в формате <i>HTML</i>	Данные для машин - в формате <i>RDF</i>
Документы в сети именуются с помощью <i>URL</i>	Произвольные сущности именуются с помощью <i>HTTP URI</i>

- *HTTP* – *Hyper Text Transfer Protocol*
(протокол передачи данных по сети)

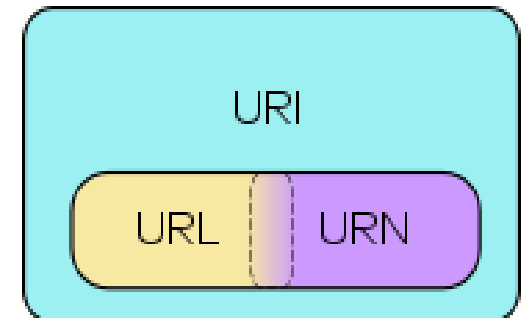
ПОНЯТИЯ *URI*, *URL* и *URN*



URI (*Uniform Resource Identifier*) – символьная строка, позволяющая идентифицировать какую-либо сущность

- Понятие возникло вместе с понятием *URL* (WWW,1989)
- Способы именования произвольных сущностей:
 - *URL* – включает местонахождение сущности
 - *URN* – идентифицирует в некотором пространстве имен, но не указывает местонахождение сущности (например, ISBN:0-395-36341-1 идентифицирует книгу)
 - *URI* – это либо *URL*, либо *URN*
- Примеры:

<http://www.w3c.org/People/Berners-Lee/>
<mailto:timbl@w3.org>



ПОНЯТИЕ *HTTP URI*



HTTP URI – *HTTP Uniform Resource Identifier*

- *HTTP URI* используются для именования сущностей в *Linked Data*, позволяют получать информацию по протоколу *HTTP*
- Для извлечения информации из разыменованного *URI* сущности используются стандарты *RDF* и *SPARQL*

HTTP URI

<http://ru.wikipedia.org>

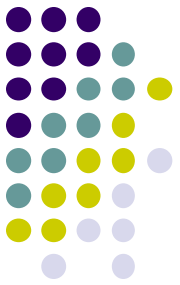
URL

<ftp://myname:mypass@myhost.com>

URI

<urn:issn:1535-3613>

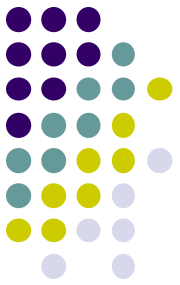
HTTP URI И СУЩНОСТИ



HTTP URI позволяет ссылаться на произвольные
сущности, в том числе понятия
(онтологии, модели ПО)

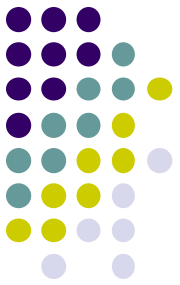
- **FOAF** (*Friend of a Friend*) –
проект по созданию модели ПЗ для машинно-
читаемых домашних страниц и социальных сетей
- Примеры используемых понятий:
 - Person – <http://xmlns.com/foaf/0.1/Person>
 - name – <http://xmlns.com/foaf/0.1/name>
 - knows – <http://xmlns.com/foaf/0.1/knows>

ПРИНЦИПЫ *Linked Data*



- Для именования сущностей и понятий используются *HTTP URI*
- Для получения данных применяется протокол *HTTP*
- По данному *URI* предоставляется информация (в удобном формате) о сущности, на которую указывает *URI*
- В описания сущностей включаются ссылки на другие сущности, чтобы сделать знания/данные связанными
- Язык *RDF* применяется для описания сущностей и их связей (отношений)

ЯЗЫК *RDF*



RDF : Resource Description Framework

- Схема, формат представления данных и метаданных
- Стандарт (де факто) для семантического веба
- Возможные метаданные (данные о данных)
 - Метаданные файлов: расширение, размер...
 - Заголовки баз данных
 - Онтологии
- Запись на ***RDF*** – множество ***триплетов*** (троек) вида
<subject, predicate, object>
 - Субъект (*subject*) – *URI* или переменная (*Blank node*)
 - Предикат (*predicate*) – *URI*
 - Объект (*object*) – *URI* , *Blank node* или константа (*литерал*)
- Для описания данных в ***RDF*** используются **словари** (онтологии) классов и отношений (связей, свойств)

ГРАФ ДАННЫХ *RDF*

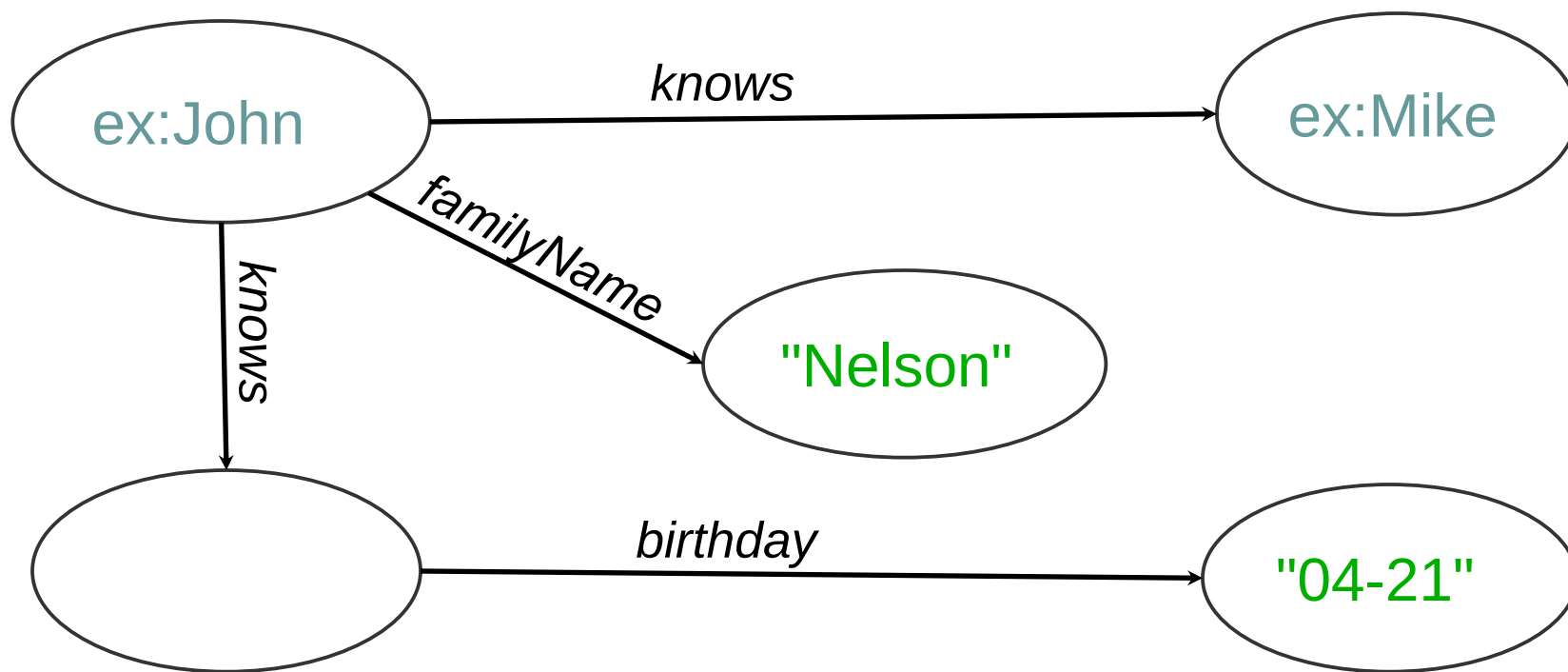


Запись на *RDF* — множество триплетов — может быть представлена графически, в виде семантической сети:

Джон знает Майка.

Фамилия Джона — Нельсон.

Джон знает человека, который родился 21 апреля.



ФОРМАТЫ *RDF*



Несколько форматов (синтаксисов) записи триплетов

- *XML* – используется для хранения и передачи данных по сети

Джона знает человека, который родился 21 апреля:

```
<foaf:Person
  rdf:about="http://example.org/Person#John">
  <foaf:knows>
    <foaf:Person foaf:birthday="04-21"/>
  </foaf:knows>
</foaf:Person>
```

- *N-Triple* – последовательно записываются три HTTP URI, заключенные в угловые скобки, в конце – точка

Джон знает Майка:

```
<http://example.org/Person#John>
  <http://xmlns.com/foaf/0.1/knows>
    <http://example.org/Person#Mike> .
```

ФОРМАТЫ *RDF: NOTATION 3*



- *Notation 3* – используются префиксы, определяемые заранее (*ex*, *foaf*, др.):

@prefix *ex*: <http://example.org/Person#> .

@prefix *foaf*: <http://xmlns.com/foaf/0.1/> .

- В результате их использования – более короткая запись, например:

Джон знает Майка

ex:John foaf:knows ex:Mike .

- Если вершина неизвестна, используется знак ?

Джон знает человека, который родился 21 апреля

ex:John foaf:knows ? .

? foaf:birthday "04-21" .

NOTATION 3: СОКРАЩЕНИЕ ЗАПИСИ



- Несколько троек с одним и тем же пустым узлом можно записать в квадратных скобках, сам пустой узел указывать в этом случае не нужно:

Джон знает человека, который родился 21 апреля

[ex:John foaf:knows ; foaf:birthday "04-21" .]

- Несколько троек с одним и тем же субъектом можно записать через точку с запятой, указав субъект только один раз:

Джон знает Майка, фамилия Джона – Нельсон

ex:John foaf:knows ex:Mike ;
foaf:familyName "Nelson" .

- Если тройки с одним субъектом и предикатом, то объекты можно записать через запятую:

ex:Alice foaf:knows ex:Mike , ex:John .

RDF-СЛОВАРЬ FOAF



FOAF – *Friend of a Friend*
(онтология без аксиом)

Классы:

- Agent
- Person
- Project
- Organization
- Group
- Document
- Image
- ...

Свойства:

- name
- age
- title
- birthday
- familyName
- givenName
- member
- ...

Отношения:

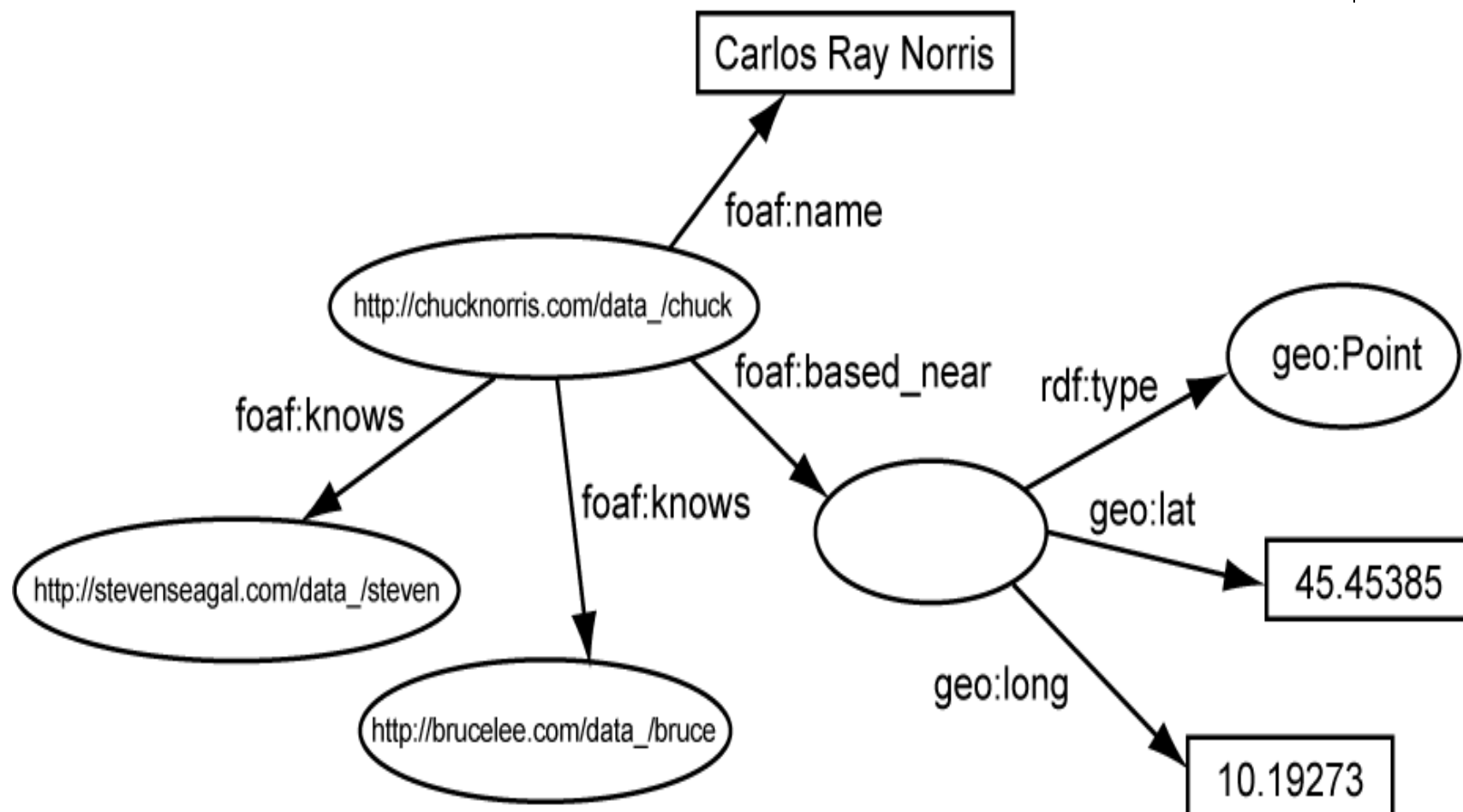
- knows
- based_near
- made
- depicts
- isPrimaryTopicOf
- currentProject
- workplaceHomepage
- ...

Каждому понятию соответствует определенный *HTTP URI* :
<http://xmlns.com/foaf/0.1/knows>

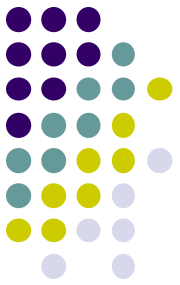
ПРИМЕР *RDF*-ГРАФА



Вершинами графа являются субъекты и объекты,
ребра – свойства и отношения

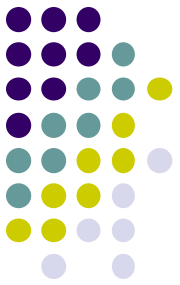


RDF-СЛОВАРИ



- *FOAF* (*Friend of a Friend*)
- *vCard* (*visiting Card*) - для описания физических и юридических лиц, а также фактов, кто с кем знаком
- *GoodRelations* - для описания товаров в магазинах
- *SIOC* (*Semantically-Interlinked Online Communities*) - для описания информации из блогов и форумов
- *SKOS* (*Simple Knowledge Organization System*) для описания тезаурусов и таксономий
- *DOAP* (*Description of a Project*) - для описания open-source проектов
- *Dublin Core* - простой набор элементов метаданных

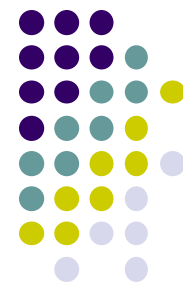
СОЗДАНИЕ НОВЫХ СЛОВАРЕЙ



RDFS : *RDF Scheme*,
RDF Vocabulary Definition Language — язык
описания *RDF*-словарей

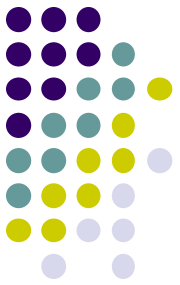
- ***RDFS*** — расширение *RDF*, позволяющее описывать простые (легкие) **онтологии**: классификации с некоторыми дополнительными отношениями
- ***OWL*** : *Web Ontology Language* — язык описания веб-онтологий, позволяет описывать более сложные онтологии

RDFS: ОПИСАНИЕ ОНТОЛОГИЙ



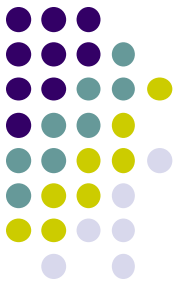
- Для описания используются компоненты:
 - `rdfs:Class`
 - отношения `rdf:type` и `rdfs:subClassOf`
- Пример описания классов:
`ex:MotorVehicle rdf:type rdfs:Class .`
`ex:PassVehicle rdf:type rdfs:Class .`
- Пример взаимосвязи (вложения) классов:
`ex:PassVehicle rdfs:subClassOf ex:MotorVehicle`
(*Пассажирский автомобиль является механическим транспортным средством*)
- ❖ Используются стандартные префиксы:
`@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.`
`@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .`

RDFS: ОПИСАНИЕ ОТНОШЕНИЙ



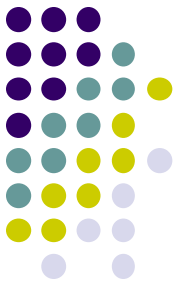
- Для описания используются:
 - класс `rdf:Property`, свойство `rdfs:subPropertyOf`
 - отношение `rdfs:domain` –
накладывает ограничение на субъект
 - отношение `rdfs:range` –
накладывает ограничение на объект
- Пример описания отношения `hasMother` с накладыванием ограничений на субъект и объект:
`ex:hasMother rdf:type rdf:Property .`
`ex:hasMother rdfs:domain ex:Person .`
`ex:hasMother rdfs:range ex:Female .`
`exstaff:John ex:hasMother exstaff:Jessica .`

RDF: ЛОГИЧЕСКИЙ ВЫВОД



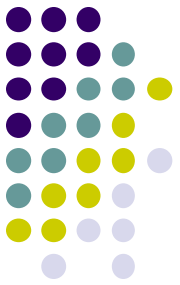
- *RDF*-данные можно интерпретировать с помощью описанных онтологий и получать логические заключения
- Описание отношения **hasMother** :
ex:hasMother rdf:type rdf:Property .
ex:hasMother rdfs:domain ex:Person .
ex:hasMother rdfs:range ex:Female .
- ❖ Известно: *У Джона есть мать Джессика*
exstaff:John ex:hasMother exstaff:Jessica .
- ❖ Значит, *Джон – человек, а Джессика – женщина*:
exstaff:John rdf:type ex:Person .
exstaff:Jessica rdf:type ex:Female .

ЯЗЫК SPARQL



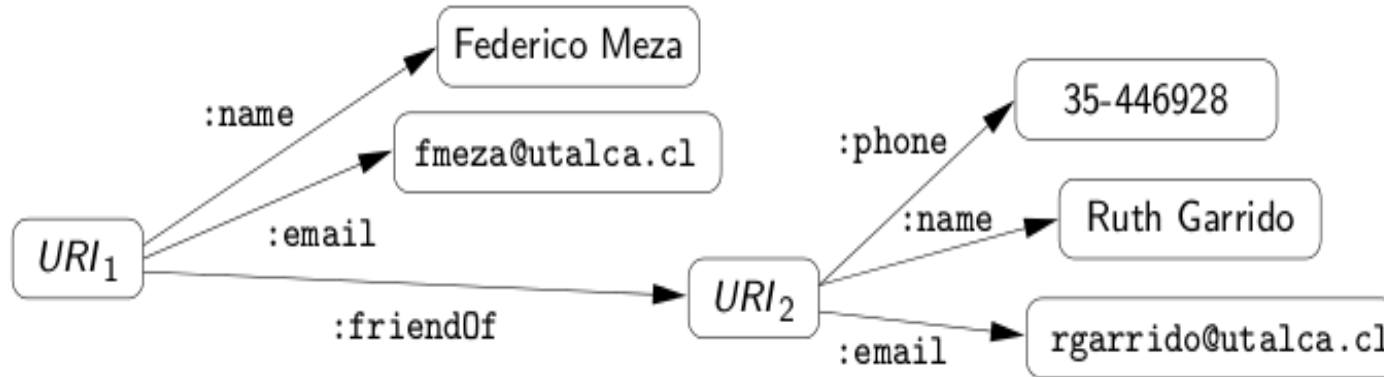
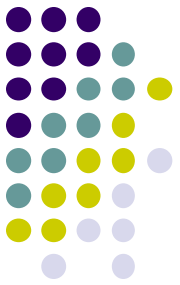
- *SPARQL Protocol and RDF Query Language* – язык запросов к данным, представленным в виде *RDF*, а также протокол для передачи этих запросов и ответов
- Запрос – шаблон графа + условие фильтрации
- *Шаблон графа* включает шаблонные триплеты
- *Шаблонный триплет*, пример: `?X :name "John" .`
- Переменные начинаются либо с `?`, либо с `$`
- Суперпозиции шаблонных триплетов:
 - `{ P1 P2 }` `\\` перечисление триплетов (&)
 - `{ P1 } UNION { P2 }` `\\` дизъюнкция
 - `{ P1 OPTIONAL { P2 } }` `\\` отношения из P2 не обязательны
 - `{ P1 FILTER ( R ) }` `\\` фильтрация выводимых данных
- Возможны префиксы:
`PREFIX foaf: <http://xmlns.com/foaf/0.1/>`

SPARQL: ВИДЫ ЗАПРОСОВ



- ***SELECT*** – возвращает всевозможные значения переменных в запросе
- ***CONSTRUCT*** – строит *RDF*-граф, подставляя значения переменных
- ***ASK*** – булевский запрос, удовлетворяет ли *RDF*-граф заданному шаблону
- ***DESCRIBE*** – возвращает структуру соответствующего *RDF*-хранилища (набора триплетов)

ПРИМЕР ЗАПРОСА *SELECT*



SELECT ?N ?E
WHERE

множество переменных

{

шаблон графа

 ?X :name ?N ;

 :email ?E .

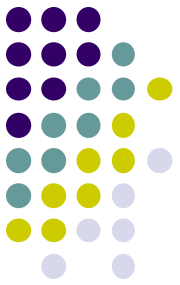
 ?X :friendOf ?Y .

 ?Y :name "Ruth Garrido" .

}

Знак ; означает продолжение использования того же субъекта

ВЫРАЗИТЕЛЬНОСТЬ *SPARQL*



Выразительность языка:

- Группировка
- Опциональные части
- Вложенность
- Объединение шаблонов
- Фильтрация

Развитие:

Регулярные выражения

Семантика языка:
отображения
(*mappings*)

```
SELECT ?X1 ?X2 ...
{{{ P1 .
    P2
    OPTIONAL { P5 } }

    { P3 .
        P4
        OPTIONAL { P7
            OPTIONAL { P8 } } }
    }
UNION
{ P9
    FILTER ( R ) }}
```

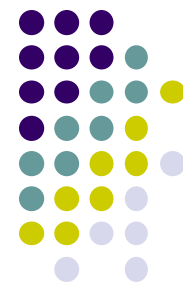
ЯЗЫК OWL



Онтологии – средство структурирования информационного контента

- **OWL** : *Web Ontology Language* - язык описания онтологий для семантического веба
- Две функции :
 - описательная: представление знаний (онтологии)
 - логическая: получение выводов из знаний)
- Модель знаний, ключевые понятия:
 - классы
 - свойства
 - экземпляры (индивиды)
 - аксиомы

OWL: КЛАССЫ и СВОЙСТВА



- Два predefined класса:

- `owl:Thing` (сущность),
- `owl:Nothing` (ничто);

Каждый класс является

родительским для `owl:Nothing` и
дочерним для `owl:Thing`

- Пример объявления класса (XML):

```
<owl:Class rdf:ID="Human"/>
```

- Отношение Класс-Подкласс (Род-Вид, наследование):

`rdfs:subClassOf`

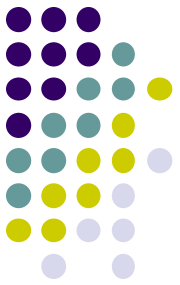
- Два класса свойств `owl:Property`:

- Связывание объектов-индивидов друг с другом:

`owl:ObjectProperty`

- Свойства типов данных: связывают индивидов с значениями типов данных: `owl:DatatypeProperty`

OWL: АКСИОМЫ КЛАССОВ



- Объявления классов:
 - `owl:Class` – объявление класса
 - `owl:SubClassOf` – объявление подкласса
 - `owl:equivalentClass` – эквивалентные классы
 - `owl:disjointWith` – классы не пересекаются
- Ограничения на значения:
 - `owl:allValuesFrom` – все значения свойства – экземпляры указанного класса, если в качестве субъекта выступает данный класс ($\forall$)
 - `owl:someValuesFrom` – существует ($\exists$)
 - `owl:hasValue` – имеет значение
- Ограничения на мощность:
 - `owl:maxCardinality` – максимальное значение
 - `owl:minCardinality` – минимальное значение
 - `owl:cardinality` – значение мощности задается явно

OWL: ПРИМЕР ОПИСАНИЯ КЛАССА



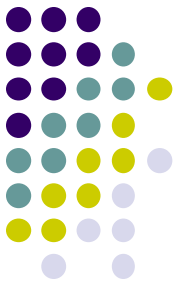
Описание класса «Футбольная команда»:

```
:ФутбольнаяКоманда rdf:type owl:Class ;  
:ФутбольнаяКоманда rdfs:subClassOf ? ;  
? rdf:type owl:Restriction ;  
? owl:onProperty :Игрок ;  
? owl:cardinality 11 .
```

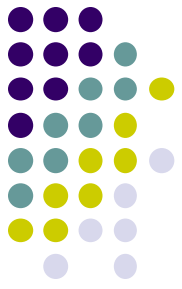
owl:Restriction – класс OWL, позволяющий задавать ограничения:

- **owl:onProperty** задает ограничение свойства
- **owl:cardinality** – ограничение на мощность класса: количество значений для заданного свойства

OWL: АКСИОМЫ СВОЙСТВ



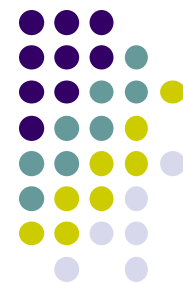
- Соотношение данного свойства с другим:
`owl:equivalentProperty` – эквивалентные свойства
`owl:inverseOf` – первое свойство является обратным для второго свойства
- Глобальные ограничения на мощность свойств:
`owl:FunctionalProperty` – функциональное:
уникальность субъектов во всех тройках
`owl:InverseFunctionalProperty` – уникальность
объектов во всех тройках
- Логические характеристики свойств:
`owl:TransitiveProperty` – транзитивность
`owl:SymmetricProperty` – симметричность



OWL: СИНТАКСИС И СЕМАНТИКА

- 5 официальных форматов синтаксиса (в том числе *RDF-based*)
- Две официальные семантики (согласованы):
 - *Direct*-семантика:
 - Основана на логике первого порядка
 - Прямые соответствия:
 - Классы $\rightarrow$ множества
 - Свойства $\rightarrow$ отношения
 - Элементы $\rightarrow$ константы и переменные проблемной области
 - *RDF-based* семантика

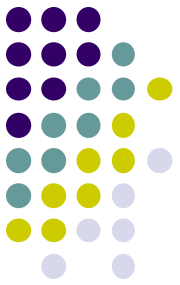
OWL: ДИАЛЕКТЫ



Три версии языка *OWL*:

- *OWL Full* – язык *RDFS* с расширенным синтаксисом
 - нет различий между классами и экземплярами классов: (Например, *Спаниель* – имя класса и экземпляра)
 - не различаются объектные свойства и свойства типов
 - неразрешимость логического вывода
- *OWL DL* – описание на основе Дескриптивных Логик
 - различает классы, типы данных, свойства типов данных, объектные свойства и индивиды
 - запрещено использовать почти все элементы *RDFS*
- *OWL Lite* – для проектирования простых таксономий с некоторыми дополнительными отношениями
 - сохранение всех ограничений *OWL DL* + дополнительные
 - запрещается использовать ограничения *owl:hasValue* и *owl:disjointWith*

OWL Lite: ПРОФИЛИ



- **OWL EL**

- полиномиальные алгоритмы для всех стандартных задач (классификации и проверки существования экземпляров)
- применяется для очень больших онтологий

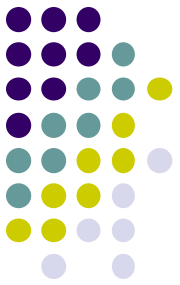
- **OWL QL**

- допускает конъюнктивные запросы с использованием технологии реляционной базы данных
- применяется при относительно легких онтологиях, с большим количеством индивидов

- **OWL RL**

- применяется для легких онтологий с большим количеством индивидов, когда необходимо работать непосредственно на данных в форме *RDF*-триплета

ЗАКЛЮЧЕНИЕ



- Последнее десятилетие:
быстрое развитие проекта *Semantic Web*,
рассмотренных формальных языков,
изучение их семантики и логическое обоснование,
поиск диалектов с приемлемой вычисл. сложностью
- Проект *Semantic Web* аккумулирует результаты,
полученные в рамках всех известных моделей ПЗ
(рассмотренных в курсе)
- *Linking Open Data project* :
цель – развитие концепции *Linked Data*
 - выявление наборов данных, доступных под открытыми лицензиями
 - конвертирование этих данных в формат RDF
 - публикация в *Web*, согласно принципам *Linked Data*

ДОМАШНЕЕ ЗАДАНИЕ № 9



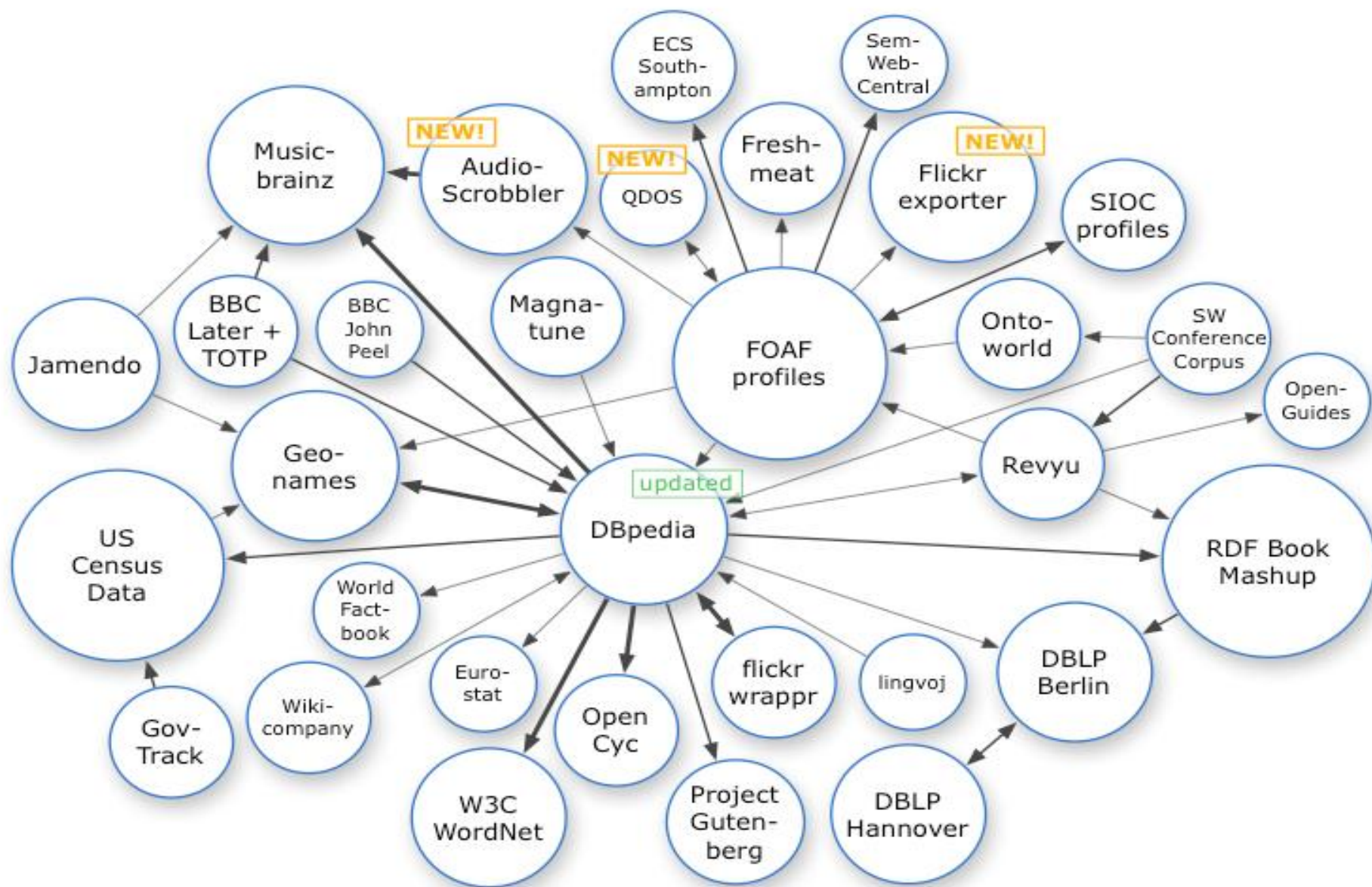
Сформулировать основные отличия средств объектно-ориентированного программирования от фреймовой модели представления знаний.

Ограничение :

- не более 5 пунктов отличий
(а реально может быть и меньше)
- не более 1500 знаков текста (без пробелов)
(≈ 3-5 баллов)

- Контрольный срок сдачи задания – 12 мая
- Сдавать задание в письменном/распечатанном виде

Linked Data 2008



Linked Data 2011

