Онтологии и OWL

Ф.В.Филиппов, доцент кафедры ИУС
СПбГУТ им. проф. М.А.Бонч-Бруевича

Изучаемые вопросы

1. Онтологии
2. Язык OWL
3. Среды разработки онтологий
4. LOD и FOAF
5. Семантические базы данных

Интересные источники:

http://www.youtube.com/watch?v=OzW3Gc_yA9A - Ontology

http://www.youtube.com/watch?v=0cj8shBSx_k - Semantic technology

http://www.rsdn.ru/article/philosophy/what-is-onto.xml - Онтологии

http://www.daml.org/ontologies - DAML ontology library

http://www.cs.umd.edu/projects/plus/SHOE/onts/index.html - **SHOE** ontology library

http://ka2portal.aifb.uni-karlsruhe.de/ - KA2 ontology library

SHOE (Simple HTML Ontology Extensions) - осуществляет аннотацию Web-страниц семантической информацией

1 Онтологии

Ф.В.Филиппов, доцент кафедры ИУС
СПбГУТ им. проф. М.А.Бонч-Бруевича
9000096@mail.ru

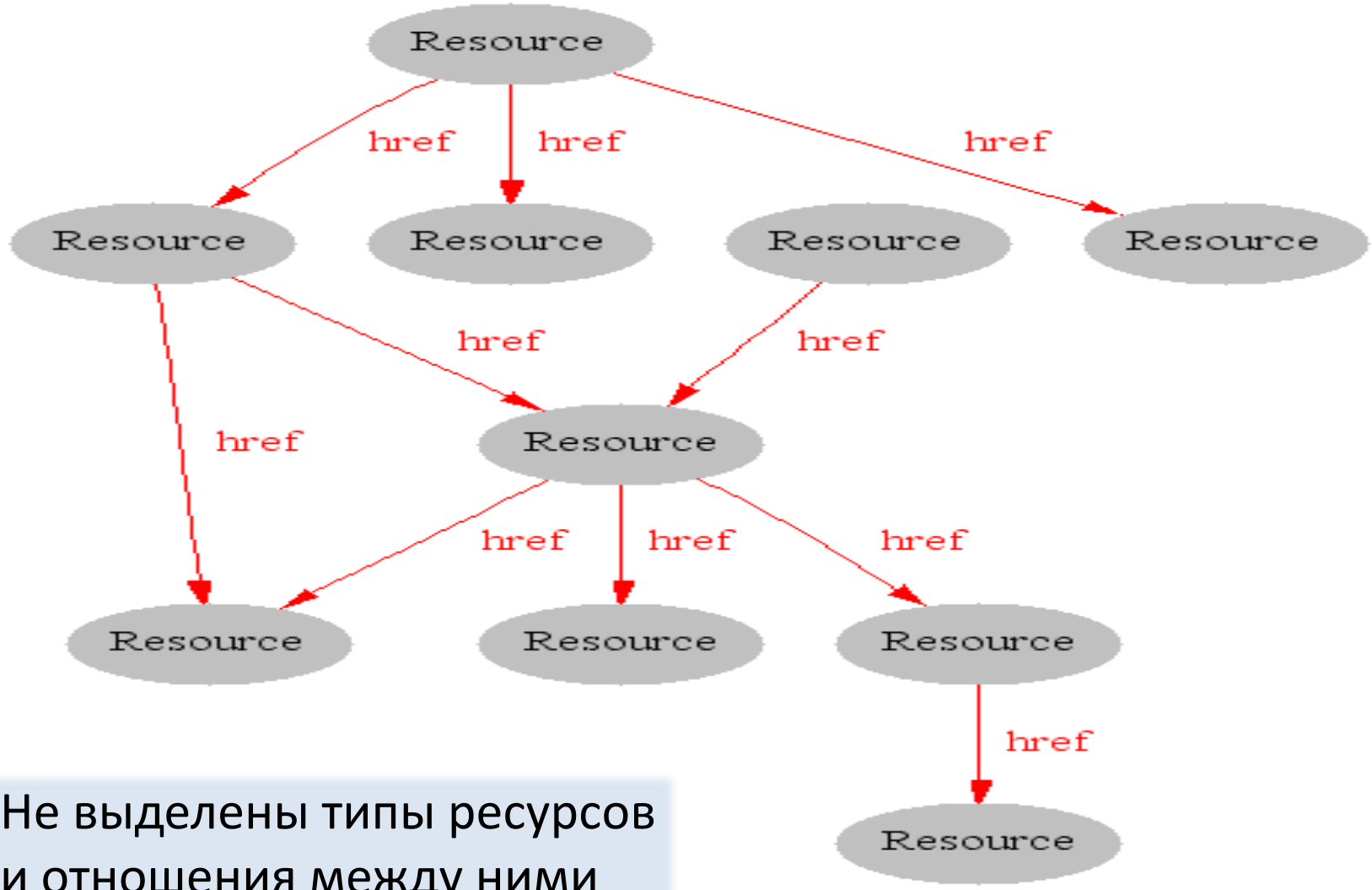
Цель использования онтологий

Semantic Web — часть глобальной концепции современного развития сети Интернет, целью которой является автоматизация интеллектуальных процессов обработки информации и знаний в сети. Именно семантические сети лучше других *соответствуют организации долговременной памяти человека*.

Обработкой и обменом информации, добычей знаний должны заниматься не люди, а интеллектуальные агенты. Для того, чтобы агенты могли взаимодействовать между собой необходимо иметь общее (разделяемое всеми) *формальное представление любого ресурса*. Именно для этой цели в Semantic Web используются ***онтологии*** (область знаний о структуре бытия).

Онтологии позволяют соответствующим программным средствам — интеллектуальным агентам автоматически, без участия человека, определять смысл терминов использованных при описании ресурсов, получать логические следствия, факты, которые не представлены в онтологии буквально, но следуют из ее ***семантики***. Это один из механизмов программных агентов, позволяющий им принимать «разумные» решения и действия.

Синтаксический Web



Не выделены типы ресурсов
и отношения между ними

Семантический Web

Акцент концепции **Semantic Web** делается на работе с *метаданными*, однозначно характеризующими свойства и содержание Web-ресурсов вместо текстового анализа документов (Text Mining). **Semantic Web** разделил средства визуализации (HTML) и средства смыслового содержания (XML → RDF + OWL).

Semantic Web создает общую Web-структуру, из которой можно выделять данные (*факты*) и *метаданные* (*знания над данными*). Такие метаданные можно многократно использовать в самых разнообразных приложениях, не только в Internet. Описание знаний в **Semantic Web** реализуется на языке **RDF** (Resource Description Framework), который определяет термины предметной области, и – языке онтологий **OWL** (Web Ontology Language), который описывает семантическую взаимосвязь ресурсов RDF.

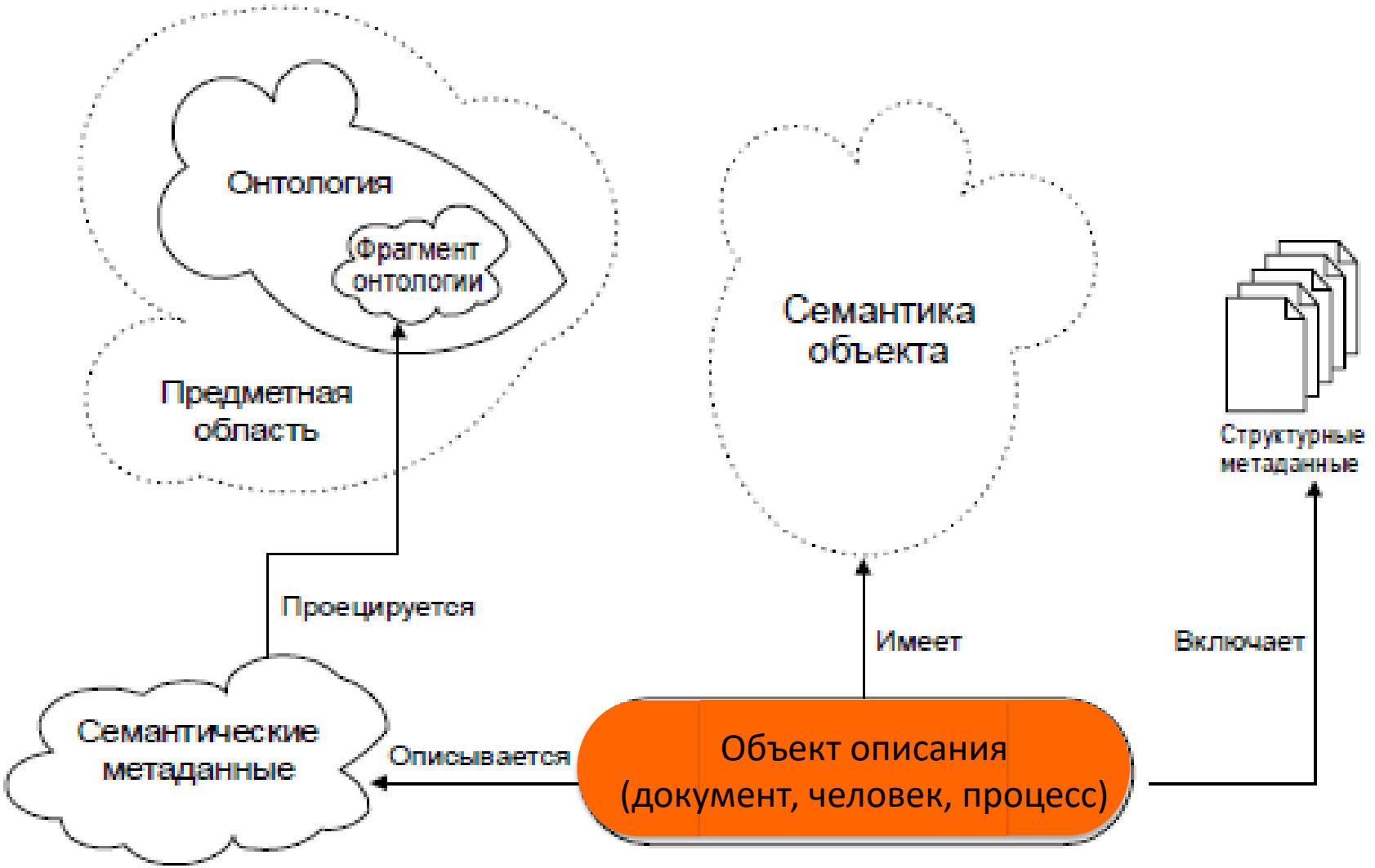
Эти языки используют **XML** (eXtensible Markup Language) для синтаксиса, и – **URI** (Uniform Resource Identifier) для идентификации ресурсов во всем мире.

Что дают онтологии?

Понятия и отношения между ними

- Онтологии обеспечивают компьютерных агентов *словарями терминов* для «понимания» их задач и взаимодействия друг с другом.
- Онтологии обладают *преемственностью*. Можно описывать семантику новых Web-объектов как комбинацию существующих терминов в ранее созданных онтологиях.
- Онтологииут *совместно использоваться* людьми или компьютерными агентами для общего понимания структуры информации.
- Онтологии позволяют сделать допущения в предметной области *явными*, с ними теперь возможно оперировать в математических выражениях.
- Онтологии позволяют отделить *знания в предметной области* от оперативных, временных знаний.
- Онтологии помогают *анализировать и получать новые знания* в предметной области.

Онтология – отражение смысла



Определение онтологии

Онтология — *формальная спецификация разделяемой концептуальной* модели, где

- под «*концептуальной*» моделью подразумевается абстрактная модель предметной области, описывающая систему понятий предметной области,
- под «*разделяемой*» подразумевается согласованное понимание концептуальной модели определенным сообществом (группой людей),
- «*спецификация*» подразумевает описание системы понятий в явном виде,
- «*формальная*» подразумевает, что концептуальная модель является машиночитаемой.

Онтология состоит из классов сущностей (концептов, индивидов) предметной области, свойств этих классов, связей между этими классами и утверждений, построенных из этих классов, их свойств и связей между ними.

Спецификация онтологии на RDF

Идентификатор (id).
Имя (fname).
Фамилия (sname).
Возраст (age).
Пол (sex).

ID	Имя	Фамилия	Возраст	Пол
1	Иван	Брусенко	39	мужской
2	Шамиль	Галимов	37	мужской
3	Татьяна	Волкова	23	женский

Объявление полей:

```
usr:id      rdf:type rdf:Property.
usr:fname   rdf:type rdf:Property.
usr:sname   rdf:type rdf:Property.
usr:age     rdf:type rdf:Property.
usr:sex     rdf:type rdf:Property.
```

Описание строк:

```
usr:user1 rdf:type usr:user.
usr:user2 rdf:type usr:user.
usr:user3 rdf:type usr:user.
```

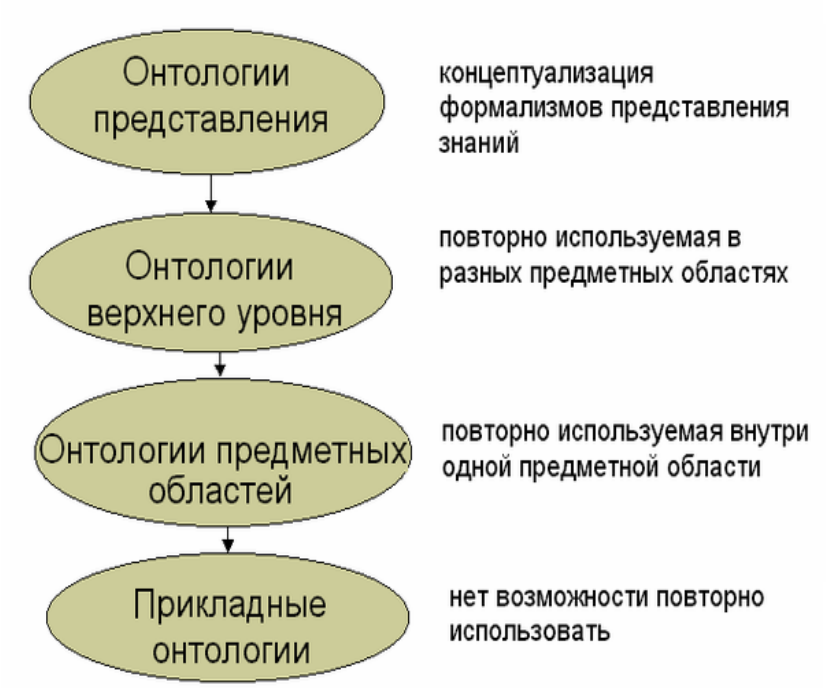
Описание данных:

```
usr:user1 usr:id      1.
usr:user1 usr:fname   Иван.
usr:user1 usr:sname   Брусенко.
usr:user1 usr:age     39.
usr:user1 usr:sex     Мужской.
usr:user2 usr:id      2.
usr:user2 usr:fname   Шамиль.
usr:user2 usr:sname   Галимов.
usr:user2 usr:age     37.
usr:user2 usr:sex     Мужской.
usr:user3 usr:id      3.
usr:user3 usr:fname   Татьяна.
usr:user3 usr:sname   Волкова.
usr:user3 usr:age     23.
usr:user3 usr:sex     Женский.
```

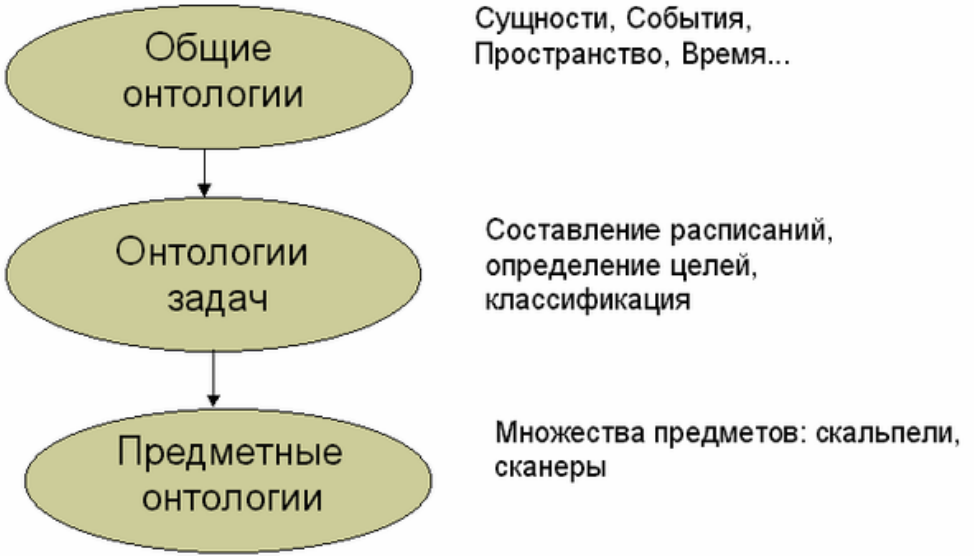
Типы и иерархии онтологий

В области каждой научной дисциплины можно определить онтологии. Уровнем выше можно описать онтологии научных областей, находящихся на стыке отдельных научных дисциплин. Еще выше можно поставить онтологию научной дисциплины вообще.

Классификация по цели создания



Классификация по содержимому



2 Язык OWL

Ф.В.Филиппов, доцент кафедры ИУС
СПбГУТ им. проф. М.А.Бонч-Бруевича
9000096@mail.ru

OWL

- **OWL** (Web Ontology Language) – язык представления онтологий в Web. Фактически это словарь расширяющий набор терминов определенных RDFS.

Три диалекта OWL

OWL Lite (простота)

OWL DL (полнота и разрешимость)

OWL Full (выразительная мощь)

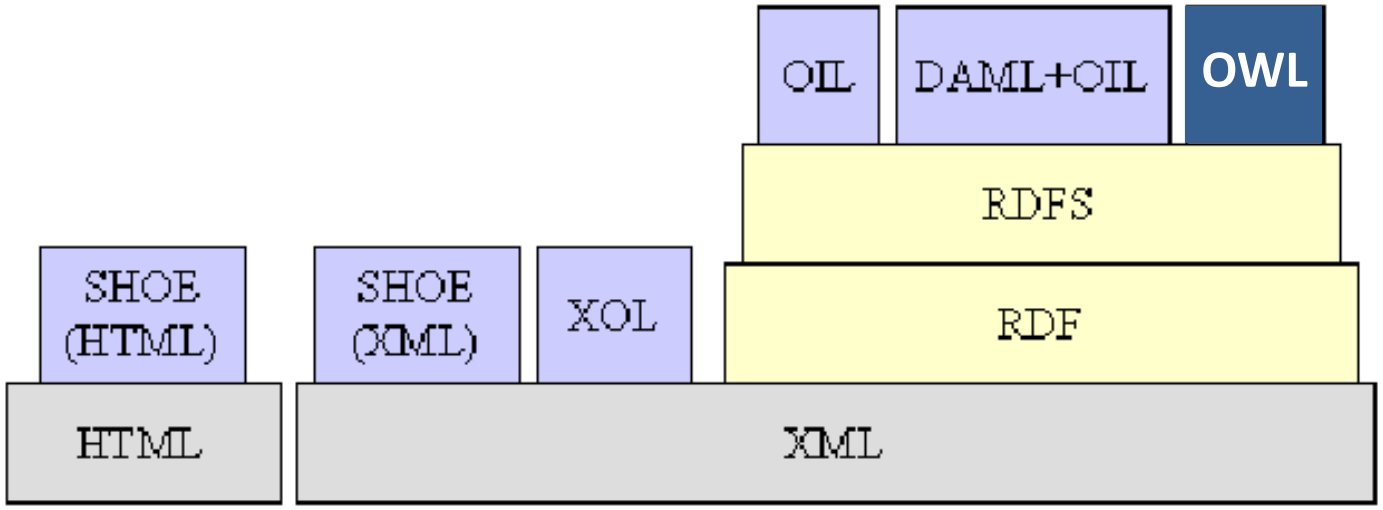
Каждый из этих диалектов (кроме Lite) является расширением предыдущего.

Как следствие:

любая OWL Lite онтология является OWL DL онтологией, а любая OWL DL онтология является OWL Full онтологией.

Исторические предшественники OWL

- **OIL** (Ontology Inference Layer)
- **DAML** (DARPA Agent Markup Language)
- **XOL** (XML-Based Ontology Exchange Language)
- **SHOE** (Simple HTML Ontology Extensions)



OWL является рекомендацией W3C и
объединяет лучшие черты своих предшественников

https://www.w3.org/TR/2004/REC-webont-req-20040210/

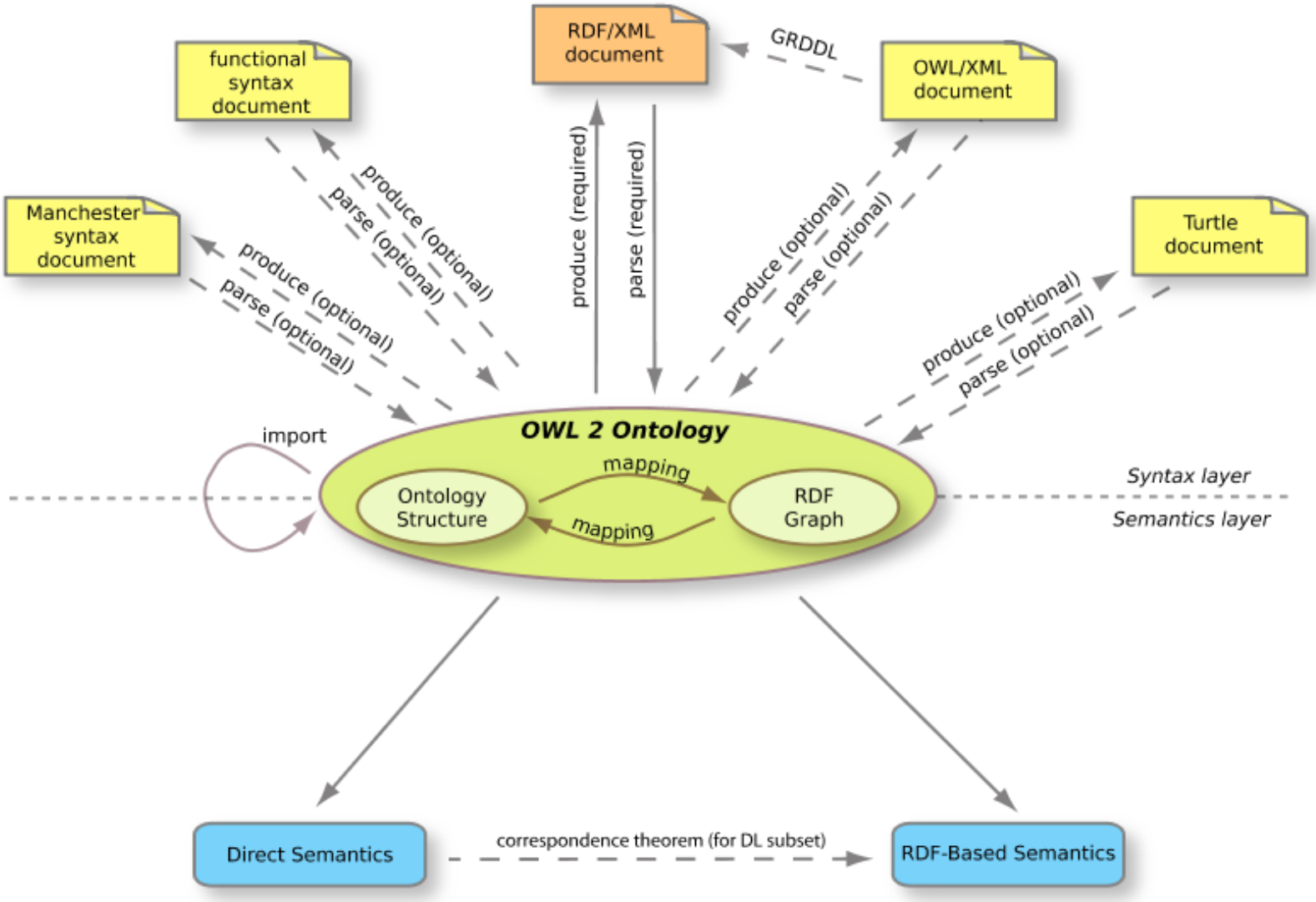
OWL Web Ontology Language
Use Cases and Requirements

W3C Recommendation 10 February 2004

New Version Available: OWL 2 (Document Status Update, 12 November 2009)

The OWL Working Group has produced a W3C Recommendation for a new version of OWL which adds features to this 2004 version, while remaining compatible. Please see OWL 2 Document Overview for an introduction to OWL 2 and a guide to the OWL 2 document set.

https://www.w3.org/TR/owl2-overview/



https://www.w3.org/TR/owl2-overview/

OWL **не является языком программирования**: OWL декларативен, т.е. он описывает “состояние дел” логическим способом.

OWL **не является языком схемы синтаксического соответствия**. В отличие от XML, он не обеспечивает средства для синтаксического структурирования документа.

OWL **не является инструментарием баз данных**. Хотя следует признать, что OWL документы хранят информацию также, как и базы данных. К тому же есть определенная аналогия между хранимой в OWL информацией и содержимым баз данных.

For Users:

OWL 2 Primer provides an approachable introduction to OWL 2.
(БУКВАРЬ)

OWL 2 Quick Reference Guide provides a brief guide to the constructs of OWL 2.
(КРАТКОЕ РУКОВОДСТВО)

https://www.w3.org/TR/2012/REC-owl2-primer-20121211/#Ontology_Management

(БУКВАРЬ)

OWL является языком для выражения онтологий. Он представляет собой набор точных описательных высказываний о какой-то части мира (предметной области). Точные описания удовлетворяют нескольким целям: прежде всего, они предотвращают недопонимания в человеческом общении и они гарантируют, что программное обеспечение ведет себя единым, предсказуемым образом.

Существуют различные **синтаксисы** языка OWL, которые служат различным целям:

- **Functional-Style** проще для спецификации и служит основой для реализации API
- **RDF/XML** является обязательным для поддержки всеми инструментариями
- **Turtle** простое представление основанное на RDF триплетах
- **Manchester** разработан, чтобы было легче понимать не логикам
- **OWL/XML** для определения OWL с помощью XML-схем

Есть инструменты, которые могут переформатировать синтаксисы из одного в другой.

https://www.w3.org/TR/2012/REC-owl2-primer-20121211/#Ontology_Management (БУКВАРЬ)

The buttons below can be used to show or hide the available syntaxes.

Hide Functional-Style Syntax | Hide RDF/XML Syntax | Hide Turtle Syntax | Hide Manchester Syntax | Hide OWL/XML Syntax

Functional-Style Syntax

```
SubClassOf( :Mother :Woman )
```

RDF/XML Syntax

```
<owl:Class rdf:about="Mother">
  <rdfs:subClassOf rdf:resource="Woman"/>
</owl:Class>
```

Turtle Syntax

```
:Mother rdfs:subClassOf :Woman .
```

Manchester Syntax

```
Class: Mother
  SubClassOf: Woman
```

OWL/XML Syntax

```
<SubClassOf>
  <Class IRI="Mother"/>
  <Class IRI="Woman"/>
</SubClassOf>
```

https://www.w3.org/TR/2012/REC-owl2-primer-20121211/#Ontology_Management

(БУКВАРЬ)

Functional-Style Syntax

```
DataPropertyAssertion( :hasAge :John "51"^^xsd:integer )
```

RDF/XML Syntax

```
<Person rdf:about="John">
  <hasAge rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">51</hasAge>
</Person>
```

Turtle Syntax

```
:John  :hasAge  51 .
```

Manchester Syntax

```
Individual: John
  Facts: hasAge "51"^^xsd:integer
```

OWL/XML Syntax

```
<DataPropertyAssertion>
  <DataProperty IRI="hasAge"/>
  <NamedIndividual IRI="John"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#integer">51</Literal>
</DataPropertyAssertion>
```

https://www.w3.org/TR/2012/REC-owl2-primer-20121211/#Ontology_Management (БУКВАРЬ)

Functional-Style Syntax

```
EquivalentClasses(
  :JohnsChildren
  ObjectHasValue( :hasParent :John )
)
```

RDF/XML Syntax

```
<owl:Class rdf:about="JohnsChildren">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasParent"/>
      <owl:hasValue rdf:resource="John"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

Turtle Syntax

```
:JohnsChildren  owl:equivalentClass  [
  rdf:type         owl:Restriction ;
  owl:onProperty   :hasParent ;
  owl:hasValue     :John
] .
```

Manchester Syntax

```
Class: JohnsChildren
  EquivalentTo: hasParent value John
```

OWL/XML Syntax

```
<EquivalentClasses>
  <Class IRI="JohnsChildren"/>
  <ObjectHasValue>
    <ObjectProperty IRI="hasParent"/>
    <NamedIndividual IRI="John"/>
  </ObjectHasValue>
</EquivalentClasses>
```

https://www.w3.org/TR/2012/REC-owl2-primer-20121211/#Ontology_Management

(ВЫЯСНИМ СТРУКТУРУ ОНТОЛОГИИ Manchester Syntax)

13 Appendix: The Complete Sample Ontology

ПРЕФИКСЫ

```
Prefix: : <http://example.com/owl/families/>
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: otherOnt: <http://example.org/otherOntologies/families/>
Ontology: <http://example.com/owl/families>
Import: <http://example.org/otherOntologies/families.owl>
```

СВОЙСТВА ОБЪЕКТОВ

```
ObjectProperty: hasWife
  SubPropertyOf: hasSpouse
  Domain:       Man
  Range:        Woman
ObjectProperty: hasParent
  InverseOf: hasChild
ObjectProperty: hasSpouse
  Characteristics: Symmetric
ObjectProperty: hasChild
  Characteristics: Asymmetric
ObjectProperty: hasRelative
  Characteristics: Reflexive
```

СВОЙСТВА ОБЪЕКТОВ

```
ObjectProperty: parentOf
  Characteristics: Irreflexive
ObjectProperty: hasHusband
  Characteristics: Functional
  Characteristics: InverseFunctional
ObjectProperty: hasAncestor
  Characteristics: Transitive
ObjectProperty: hasGrandparent
  SubPropertyChain: hasParent o hasParent
ObjectProperty: hasUncle
  SubPropertyChain: hasFather o hasBrother
ObjectProperty: hasFather
  SubPropertyOf: hasParent
ObjectProperty: hasBrother
ObjectProperty: hasDaughter
ObjectProperty: hasSon
ObjectProperty: loves
```

НЕСОВМЕСТИМЫЕ и ЭКВИВАЛЕНТНЫЕ СВОЙСТВА

```
DisjointProperties: hasParent, hasSpouse
DisjointProperties: hasSon,    hasDaughter
EquivalentProperties: hasChild, otherOnt:child
EquivalentProperties: hasAge,   otherOnt:age
```

СВОЙСТВА ДАННЫХ

```
DataProperty: hasAge
  Domain: Person
  Range:  xsd:nonNegativeInteger
  Characteristics: Functional
DataProperty: hasSSN
```

ТИПЫ ДАННЫХ

```
Datatype: personAge
  EquivalentTo: integer[<= 0 , >= 150]
Datatype: minorAge
  EquivalentTo: integer[<= 0 , >= 18]
Datatype: majorAge
   EquivalentTo: personAge and not minorAge
Datatype: toddlerAge
   EquivalentTo: { 1, 2 }
Datatype: minorAge
```

КЛАССЫ

```
Class: Woman
  SubClassOf: Person
Class: Mother
  SubClassOf:   Woman
  EquivalentTo: Woman and Parent
Class: Person
  Annotations:  rdfs:comment "Represents the set of all people."
  EquivalentTo: Human
  HasKey: hasSSN
```

КЛАССЫ

```
Class: Parent
  EquivalentTo: hasChild some Person
  EquivalentTo: Mother or Father
Class: ChildlessPerson
  EquivalentTo: Person and not Parent
  SubClassOf:   Person and not (inverse hasParent some owl:Thing)
Class: Grandfather
  SubClassOf: Man and Parent
Class: HappyPerson
  EquivalentTo: hasChild only Happy and hasChild some Happy
Class: JohnsChildren
  EquivalentTo: hasParent value John
Class: NarcisticPerson
  EquivalentTo: loves Self
Class: Orphan
  EquivalentTo: inverse hasChild only Dead
Class: Teenager
 SubClassOf: hasAge some integer[<= 13 , >= 19]
Class: Man
  SubClassOf: Annotations: rdfs:comment "States that every man is a person."
Person
Class: MyBirthdayGuests
  EquivalentTo: { Bill, John, Mary }
```

КЛАССЫ

Class: Father
  SubClassOf: Man and Parent
Class: X
  SubClassOf: Parent and hasChild max 1 and hasChild only Female
  EquivalentTo: {Mary, Bill, Meg} and Female
Class: Adult
Class: Dead
Class: Father
Class: Female
Class: Happy
Class: Human
Class: SocialRole
Class: YoungChild

НЕСОВМЕСТИМЫЕ и ЭКВИВАЛЕНТНЫЕ КЛАССЫ

DisjointClasses: Mother, Father, YoungChild
DisjointClasses: Woman, Man
EquivalentClasses: Adult, otherOnt:Grownup

ИНДИВИДЫ

Individual: Mary
  Types: Person
  Types: Woman
Individual: Jack
  Types: Person and not Parent

ИНДИВИДЫ

```
Individual: John
  Types: Father
  Types: hasChild max 4 Parent
  Types: hasChild min 2 Parent
  Types: hasChild exactly 3 Parent
  Types: hasChild exactly 5
  Facts: hasAge "51"^^xsd:integer
  Facts: hasWife Mary
  DifferentFrom: Bill
Individual: Bill
  Facts: not hasWife     Mary
  Facts: not hasDaughter Susan
Individual: James
  SameAs: Jim
Individual: Jack
  Facts: not hasAge "53"^^xsd:integer
Individual: Father
 Types: SocialRole
Individual: Meg
Individual: Susan
Individual: Jim
Individual: otherOnt:JohnBrown
Individual: otherOnt:MaryBrown
```

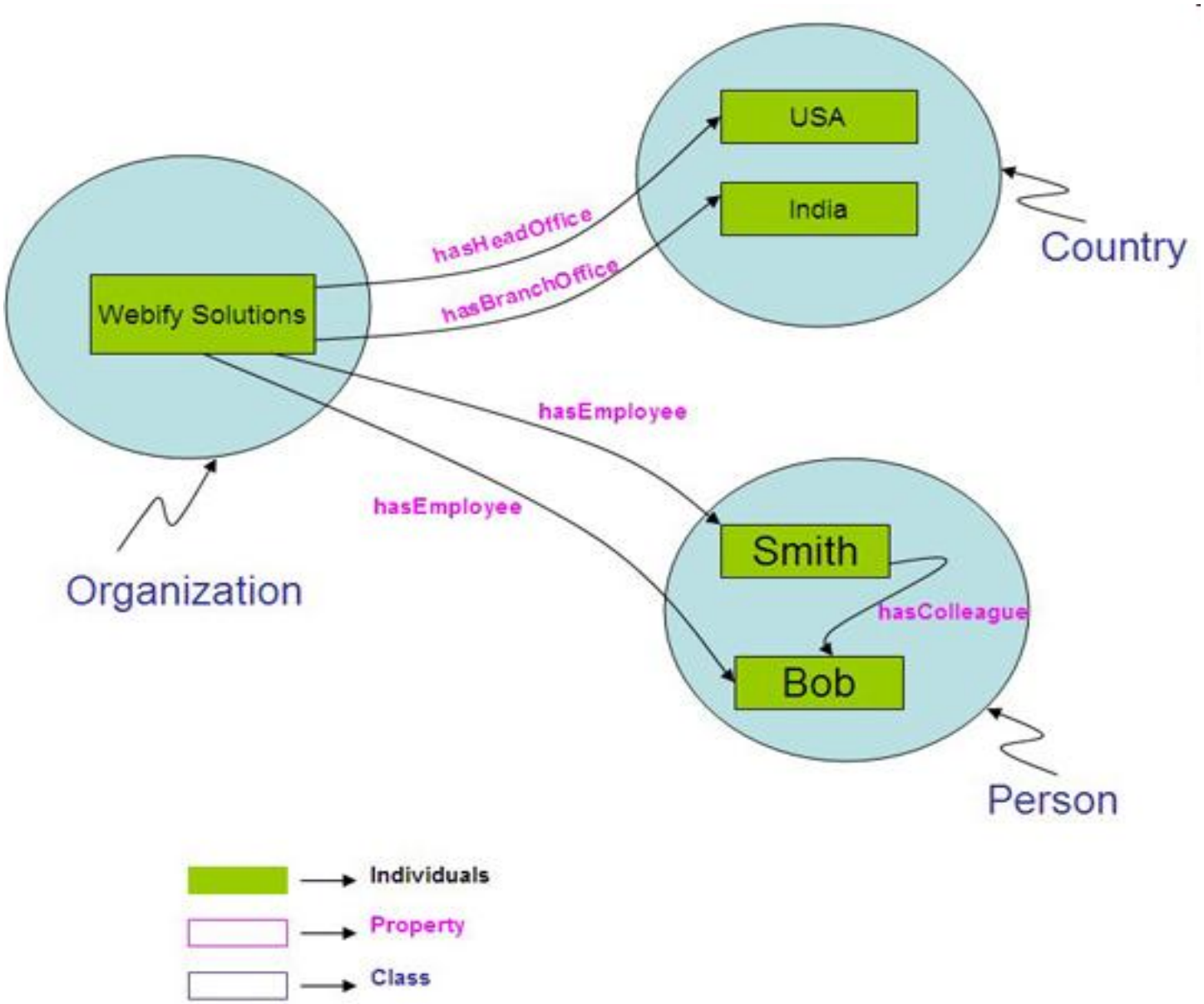
ИДЕНТИЧНЫЕ ИНДИВИДЫ
SameIndividual: John, otherOnt:JohnBrown
SameIndividual: Mary, otherOnt:MaryBrown

https://www.w3.org/TR/2012/REC-owl2-primer-20121211/#Ontology_Management

(Итого: СТРУКТУРА ОНТОЛОГИИ на OWL)

ПРЕФИКСЫ
СВОЙСТВА ОБЪЕКТОВ
НЕСОВМЕСТИМЫЕ и ЭКВИВАЛЕНТНЫЕ СВОЙСТВА
СВОЙСТВА ДАННЫХ
ТИПЫ ДАННЫХ
КЛАССЫ
НЕСОВМЕСТИМЫЕ и ЭКВИВАЛЕНТНЫЕ КЛАССЫ
ИНДИВИДЫ
ИДЕНТИЧНЫЕ ИНДИВИДЫ

Базовые элементы OWL - пример



Некоторые конструкции OWL

Префиксы указывают подмножество языка онтологий (owl, rdf, rdfs) для описания терминов метаданных.

owl: Class (Thing, Nothing)	Класс определяет группу индивидов (сущностей, концептов), которых объединяет наличие некоторых общих свойств. Thing - самый общий класс, Nothing - пустой
rdfs: subClassOf	Иерархии классов создаются путем одного или нескольких утверждений о том, что данный класс - подкласс другого класса.
rdf: Property	Свойства используются, чтобы установить отношения между индивидами или индивидами и значениями данных.
rdfs: subPropertyOf	Иерархии свойств создаются путем одного или нескольких утверждений, что данное свойство - подсвойство одного или нескольких других свойств.
rdfs: domain	Область определения свойства, ограничивает индивидов, к которым это свойство может быть применено.
rdfs: range	Диапазон свойства, ограничивающий индивидов, которые могут выступать в качестве значений этого свойства.

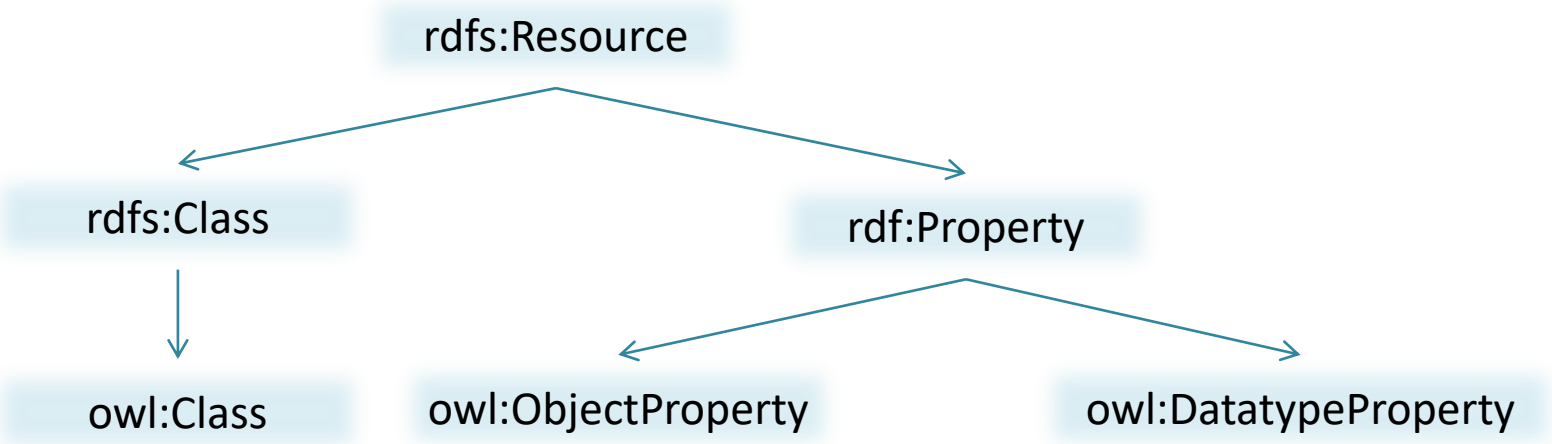
Некоторые конструкции OWL (продолжение…)

owl:equivalentClass	Два класса представляются как эквивалентные. Эквивалентные классы имеют одних и тех же представителей. Равенство используется для создания синонимичных классов.
owl:equivalentProperty	Два свойства представляются как эквивалентные. Эквивалентные свойства связывают одного индивида с одним и тем же набором других индивидов. Равенство используется для создания синонимичных свойств.
owl:ObjectProperty	Свойства, связывающие представителей двух классов.
owl:DatatypeProperty	Свойства, связывающие представителей класса с представителями типов данных.

См. полностью:

http://sherdim.ru/pts/semantic_web/REC-owl-features-20040210_ru.html

Иерархия классов OWL и RDF/RDFS



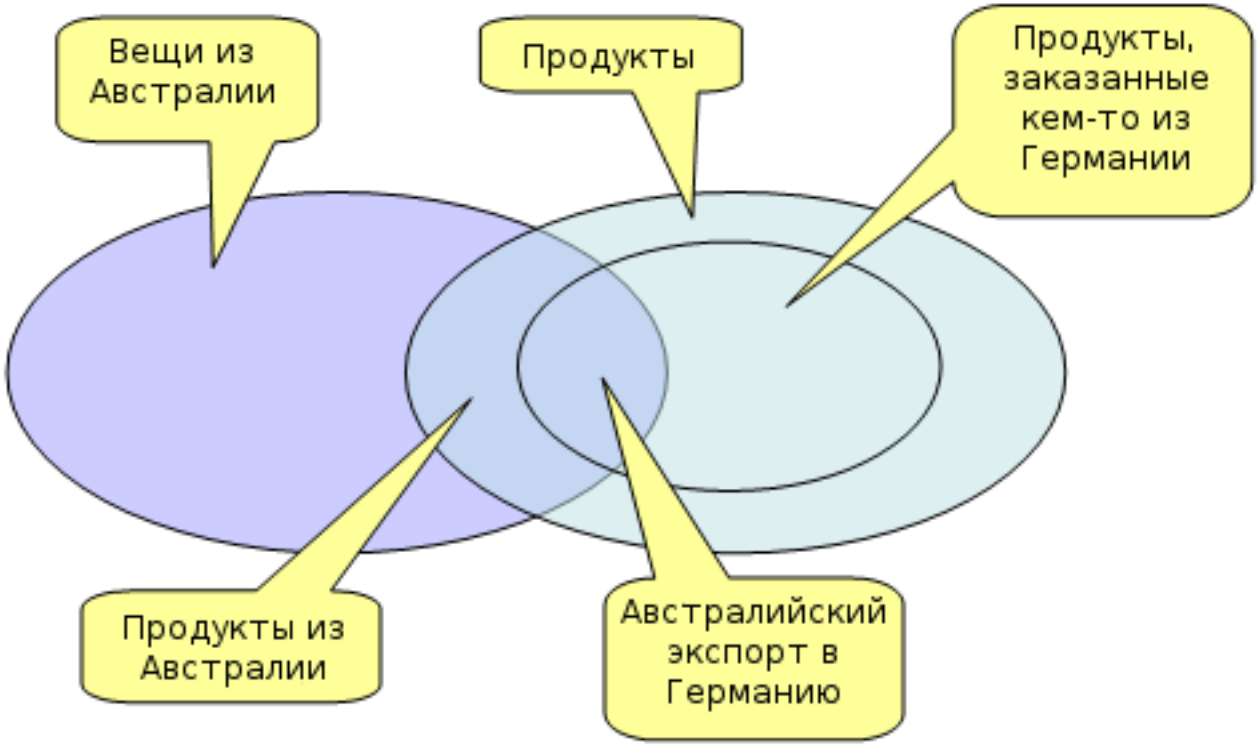
OWL-конструкторы типа owl:Class, owl:DatatypeProperty и owl: ObjectProperty являются специализациями их двойников в RDF/RDFS

Базовые элементы OWL - классы

Класс может быть определен:

1. Идентификатором класса (URI).
2. Перечислением всех экземпляров класса.
3. Ограничением свойства.
4. Пересечением двух и более определений классов.
5. Объединением двух и более определений классов.
6. Дополнением определения класса.

Представление OWL - классов



Классы в OWL могут рассматриваться *как множества сущностей* с общими характеристиками

Конструкции OWL для аксиом

Простейшая аксиома, определяющая *именованный класс*:

- <owl:Class rdf:ID="Human"/>
- Все что постулирует эта аксиома – существование класса с именем “Human”.

Конструкции OWL для определения *сложных аксиом классов*:

- **rdfs:subClassOf** - говорит о том, что экстенсионал одного класса (подкласс) полностью входит в экстенсионал другого (надкласс).
- **owl:equivalentClass** - говорит о том, что экстенсионалы двух классов совпадают.
- **owl:disjointWith** - говорит о том, что экстенсионалы двух классов не пересекаются.

Категории OWL свойств

Две основные категории OWL свойств:

- объектные свойства (owl:ObjectProperty)
  - связывают между собой индивиды
- свойства-значения (owl:DatatypeProperty)
  - связывают индивиды со значениями данных

Оба класса свойств являются подклассами класса rdf:Property

Простейший пример аксиомы свойства:

<owl:ObjectProperty rdf:ID="hasParent"/>

Всё что постулирует эта аксиома – существование некоторого свойства "hasParent" связывающего экземпляры owl:Thing друг с другом.

Конструкции определения свойств

- Конструкции RDF Schema:

  rdfs:subPropertyOf (определяет подсвойство данного свойства)

  rdfs:domain (определяет домен)

  rdfs:range (определяет диапазон)

- Отношения между свойствами:

  owl:equivalentProperty (определяет эквивалентное свойство)

  owl:inverseOf (определяет обратное свойство)

- Логические характеристики свойства:

  owl:SymmetricProperty (определяет свойство как симметричное данному)

  owl:TransitiveProperty (определяет транзитивное свойство)

Свойства свойств

Каждое объектное свойство может иметь следующие основные свойства (свойства у свойства):

• домен (**rdfs:domen**) – классы, у объектов которых имеется это свойство (область возможного применения);

• диапазон (**rdfs:range**) – классы, объекты которых могут быть значениями свойства (область допустимых значений);

• наследование (**rdfs:subPropertyOf**) – свойство, которое является базовым для описываемого свойства.

Можно объявить более чем один домен или диапазон. В этом случае используется пересечение доменов или диапазонов.

Определение индивидов OWL

Индивиды определяются при помощи *аксиом индивидов* (фактов)

Два вида фактов (аксиом):

- (1) Факты о членстве индивидов в классах и о значении свойств индивидов – аксиомы первого вида.
- (2) Факты об идентичности индивидов – аксиомы второго вида.

ФАКТ≡АКСИОМА

индивиды|объекты|сущности|концепты

Пример аксиомы первого вида

```
<Балет rdf:ID="ЛебединоеОзеро">
    <имеетКомпозитора rdf:resource="#Чайковский"/>
</Балет>
```

Данная аксиома постулирует сразу 2 факта:

- (а) существует некоторый индивид класса “Балет” имеющий имя “ЛебединоеОзеро”;
- (б) этот индивид связан свойством “имеетКомпозитора” с индивидом: “Чайковский” (определенным где-то в другом месте).

Первый факт говорит о членстве в классе, второй – о значении свойства индивида.

Аксиомы второго вида

Для описания фактов об идентичности индивидов используются *аксиомы идентичности*

Вспомогательные конструкции OWL:

- *owl:sameAs* постулирует, что две ссылки URI ссылаются на один и тот же индивид.
- *owl:differentFrom* постулирует, что две ссылки URI ссылаются на разные индивиды.
- *owl:AllDifferent* предоставляет средство для определения списка попарно различных индивидов.

Полный список конструкций OWL Lite

Термины RDF Schema:

- Class (Thing, Nothing)
- rdfs:subClassOf
- rdf:Property
- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range
- Individual

(Не)Равенство:

- equivalentClass
- equivalentProperty
- sameAs
- differentFrom
- AllDifferent
- distinctMembers

Характеристики свойств:

- ObjectProperty
- DatatypeProperty
- inverseOf
- TransitiveProperty
- SymmetricProperty
- FunctionalProperty
- InverseFunctionalProperty

Ограничения свойств:

- Restriction
- onProperty
- allValuesFrom
- someValuesFrom

Ограниченная кардинальность:

- minCardinality (только 0 или 1)
- maxCardinality (только 0 или 1)
- cardinality (only 0 or 1)

Информация заголовка:

- Ontology
- imports

Пересечение классов:

- intersectionOf

Типы данных

- xsd datatypes

Версии:

- versionInfo
- priorVersion
- backwardCompatibleWith
- incompatibleWith
- DeprecatedClass
- DeprecatedProperty

Аннотационные свойства:

- rdfs:label
- rdfs:comment
- rdfs:seeAlso
- rdfs:isDefinedBy
- AnnotationProperty
- OntologyProperty

Расширение конструкций OWL Lite до OWL DL и Full

Class Axioms:

- *oneOf, dataRange*
- *disjointWith*
- *equivalentClass*
  (applied to class expressions)
- *rdfs:subClassOf*
  (applied to class expressions)

Boolean Combinations of Class Expressions:

- *unionOf*
- *complementOf*
- *intersectionOf*

Arbitrary Cardinality:

- *minCardinality*
- *maxCardinality*
- *cardinality*

Filler Information:

- *hasValue*

https://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/
(КРАТКОЕ РУКОВОДСТВО)

Стандартные префиксные имена в OWL:

rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
owl:	http://www.w3.org/2002/07/owl#
xsd:	http://www.w3.org/2001/XMLSchema#

Определение классов:

_x rdf:type owl:Class.
_x owl:oneOf (a_1…a_n).

Предопределенные классы

Универсальный класс	owl:Thing
Пустой класс:	owl:Nothing

Булевские соединения и перечисление индивидов

Пересечение:	_x owl:intersectionOf (C_1…C_n).
Объединение:	_x owl:unionOf (C_1…C_n).
Дополнение:	_x owl:complementOf (C).
Перечисление:	_x owl:oneOf (a_1…a_n).

https://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/
(КРАТКОЕ РУКОВОДСТВО)

Определение классов:

```
_:x rdf:type owl:Restriction.
_:x owl:onProperty P.
_:x owl:cardinality n.
```

Ограничение свойств объектов

Все:	_x: owl:allValuesFrom C.
Некоторые:	_x: owl:someValuesFrom C.
Индивидуальное значение:	_x: owl:hasValue a.
Точное количество:	_:x owl:cardinality n.

. . .

```
_:x rdf:type owl:Restriction.
_:x owl:onProperty R.
_:x owl:cardinality n.
```

Ограничение свойств данных

Все:	_x: owl:allValuesFrom D.
Некоторые:	_x: owl:someValuesFrom D.
Значение литерала:	_x: owl:hasValue v.
Точное количество:	_:x owl:cardinality n.

. . .

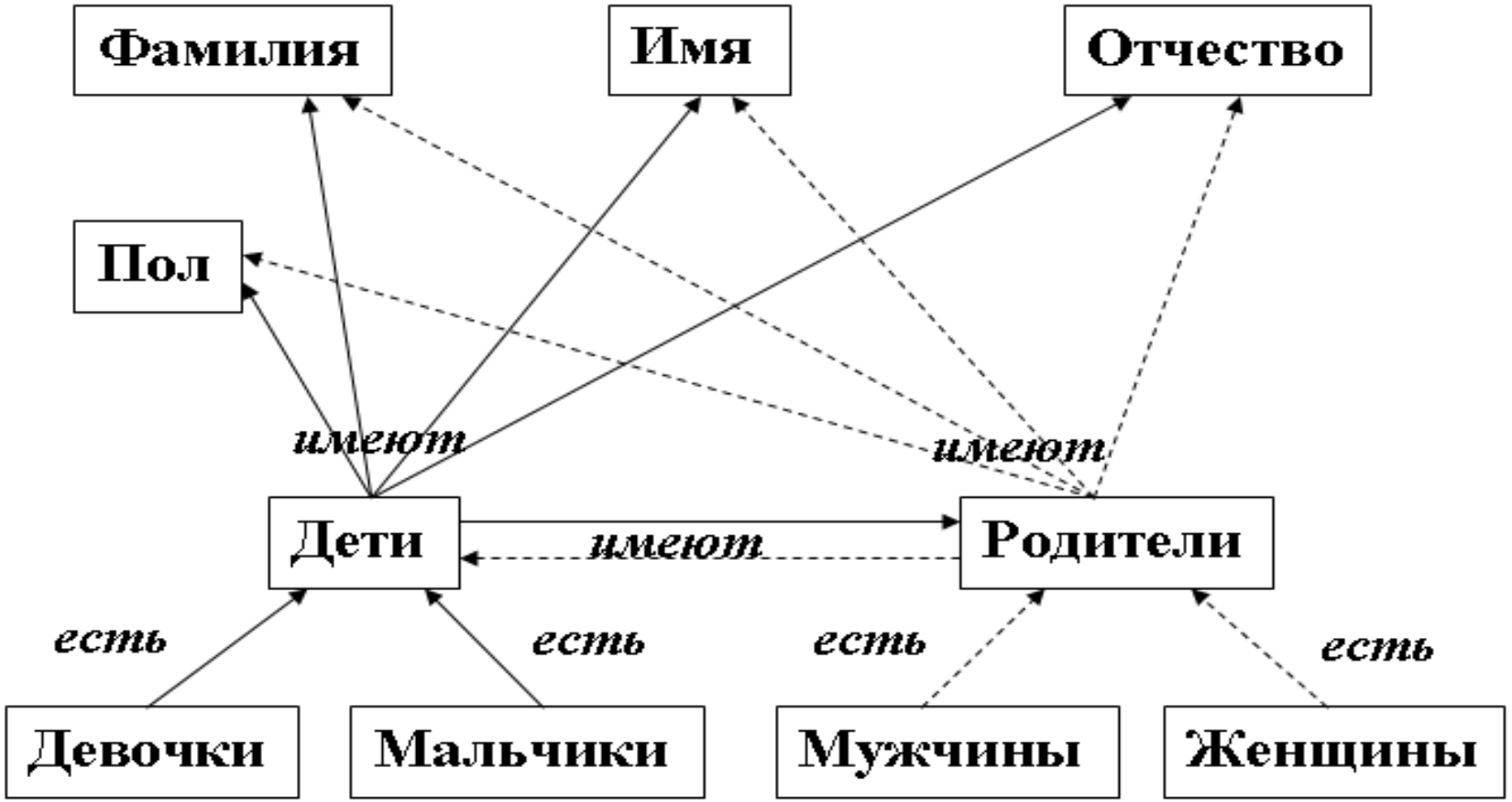
Жизненный цикл онтологии

Построение (creating).
Заполнение (populating).
Проверка (validating).
Использование (deploying).
Поддержка (maintaining).
Развитие (evolving).

3 Среды разработки онтологий

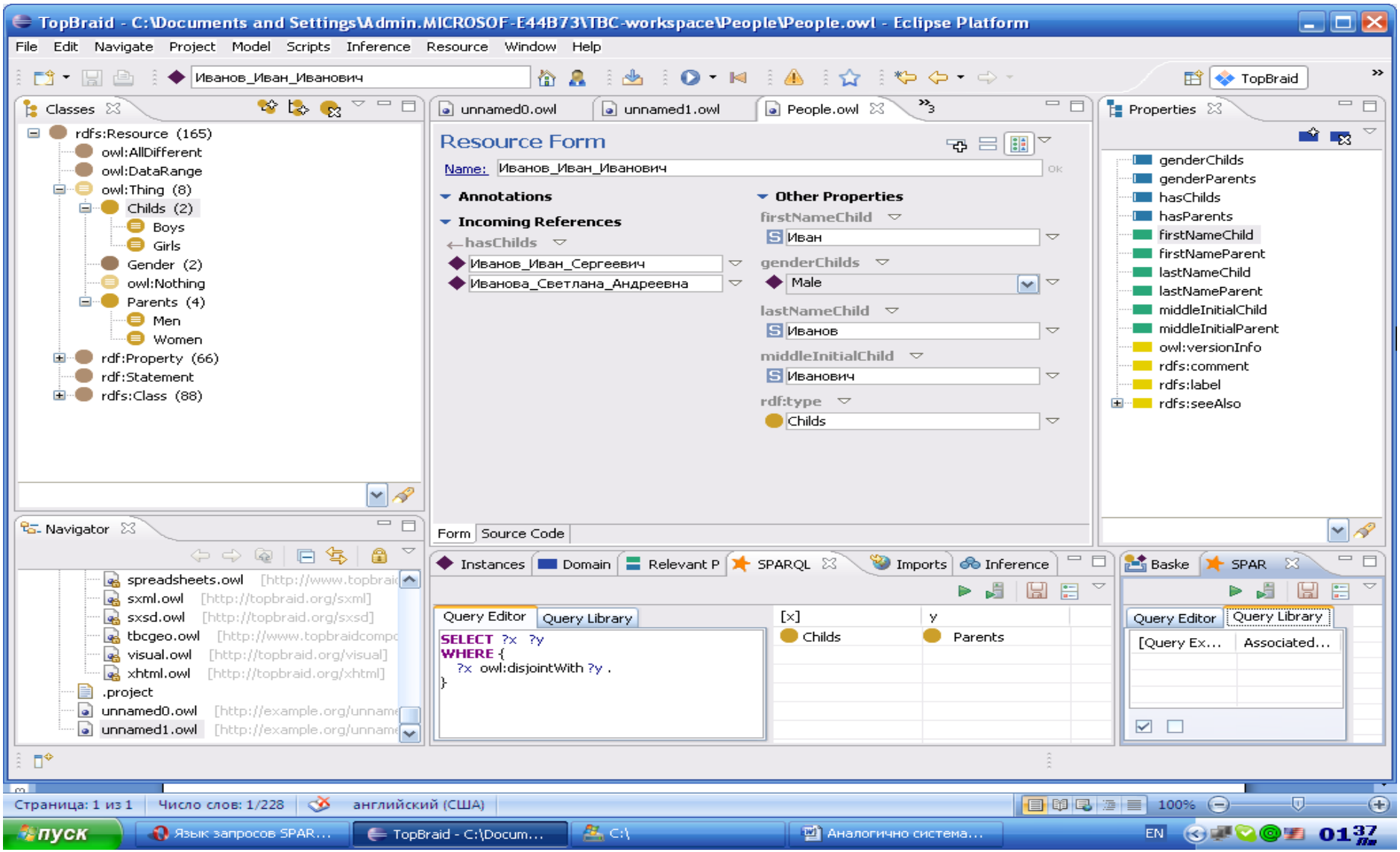
Ф.В.Филиппов, доцент кафедры ИУС
СПбГУТ им. проф. М.А.Бонч-Бруевича
9000096@mail.ru

Пример разработки базы знаний

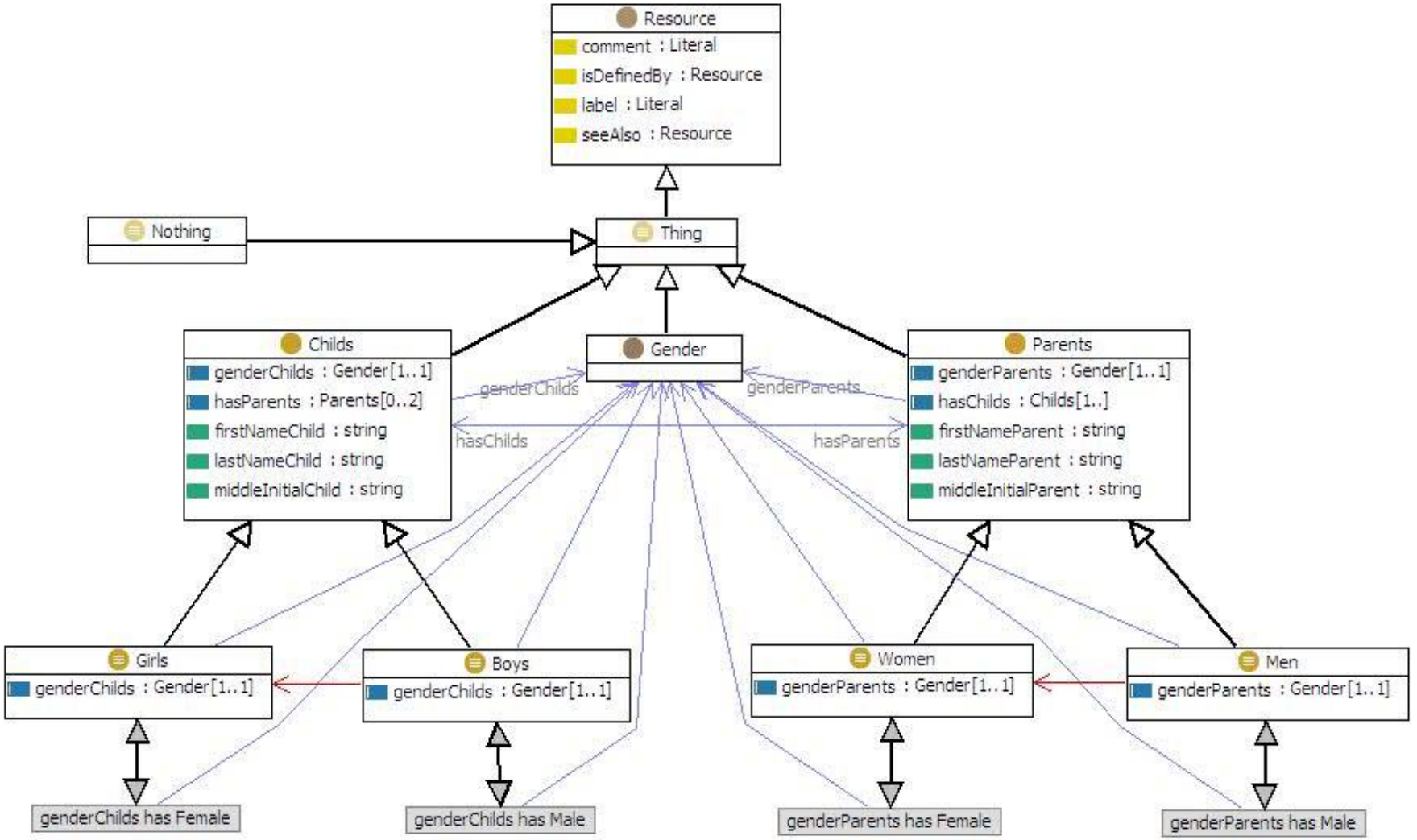


Семантическая сеть

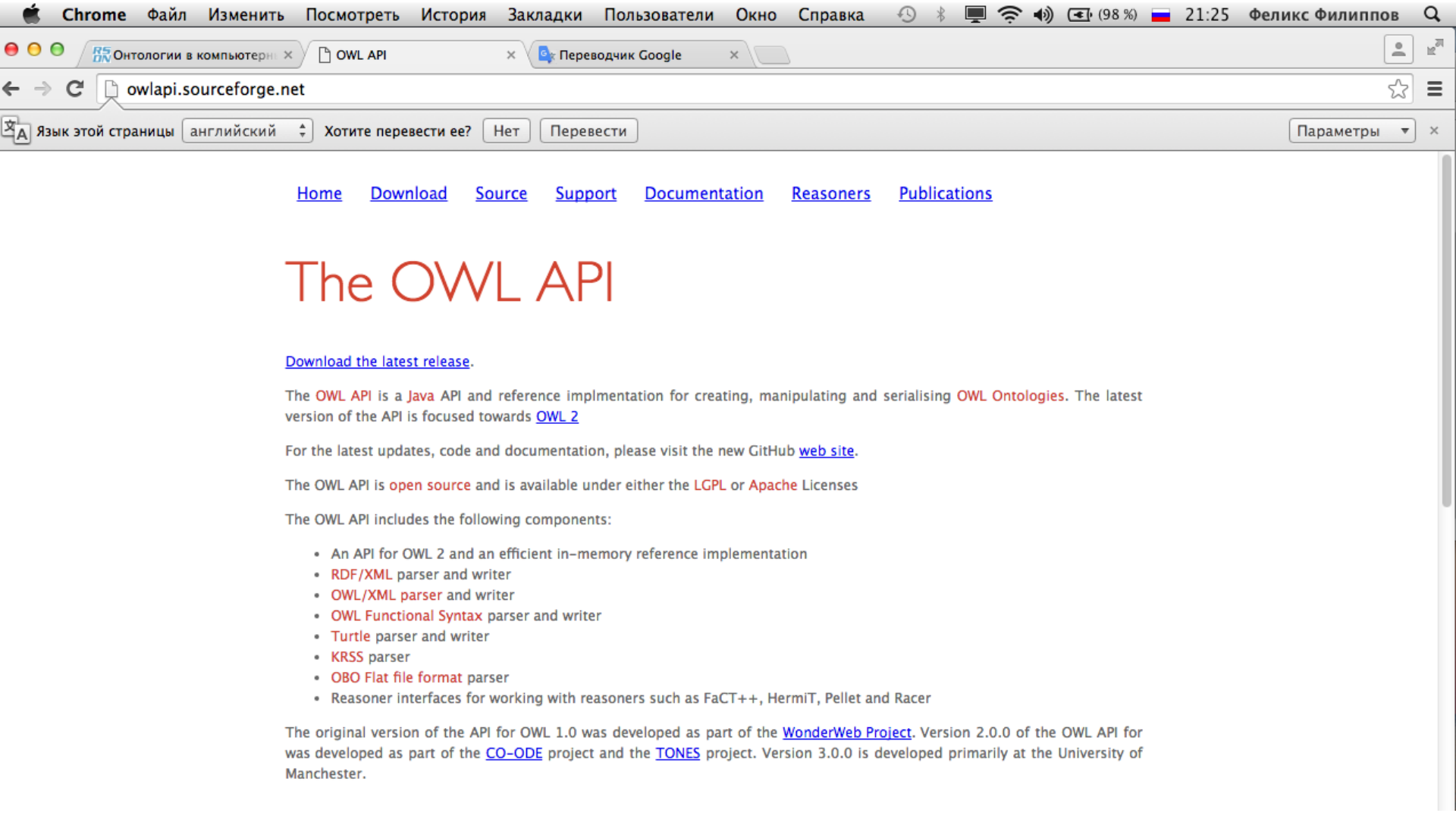
Среда TopBraid формирования БЗ



Визуальное представление онтологий



http://owlapi.sourceforge.net



Create, query and write RDF graphs.

Description

The R package *redland* provides methods to create, query and write information stored in the Resource Description Framework (RDF). This package is implemente
R scripts that provide an R interface (aka "wrapper") to the Redland RDF C libraries. Documentation for the redland R package classes and functions are available f
the standard R help facility, for example, `'help("Node-class")'`, `'?getNodeType'`, etc.

An overview of the redland R package is available with the R command: `'vignette("redland_overview")'`.

The Redland C library functions are described at http://librdf.org/docs/api/index.html.

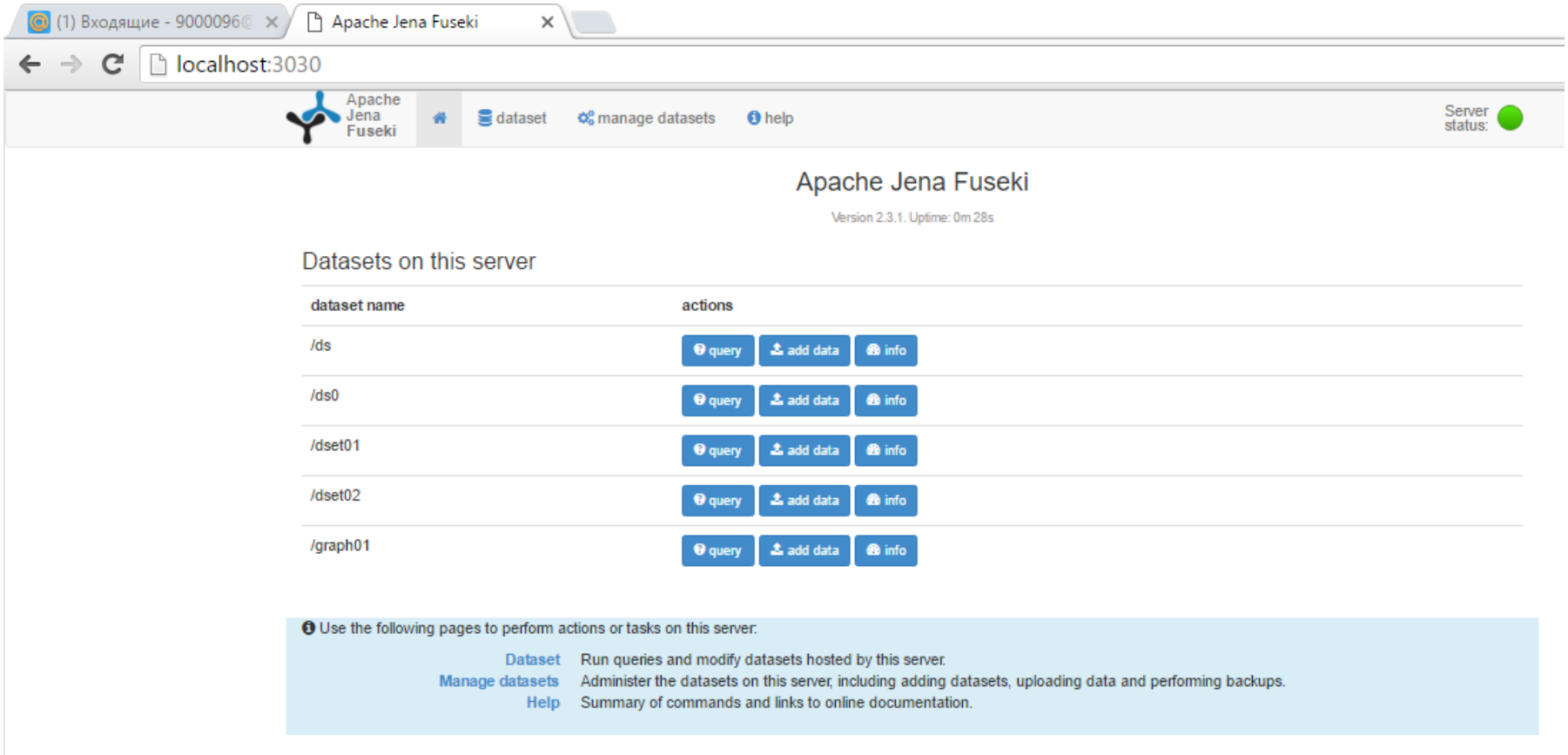
An introduction to RDF can be found at http://www.w3.org/TR/rdf-primer.

Details

The redland R package classes and the corresponding Redland C library types are shown in the following table:

Concept	Redland C type	redland R class	Purpose
Resource / Literal	librdf_node	redland::Node	RDF Model & Syntax nodes
Statement / Triple	librdf_statement	redland::Statement	RDF Model & Syntax arcs (statements, triples) [isa Resource]
Model	librdf_model	redland::Model	Set of Statements usually held in one Storage.
Node	librdf_node	redland::Node	The subject, predicate or object of a Statement
Storage	librdf_storage	redland::Storage	Storage for Models either persistant or in-memory.
Parser	librdf_parser	redland::Parser	Syntaxes parsers delivering Stream of Statements or writing to a Model

Apache Jena Fuseki



Apache Jena Fuseki

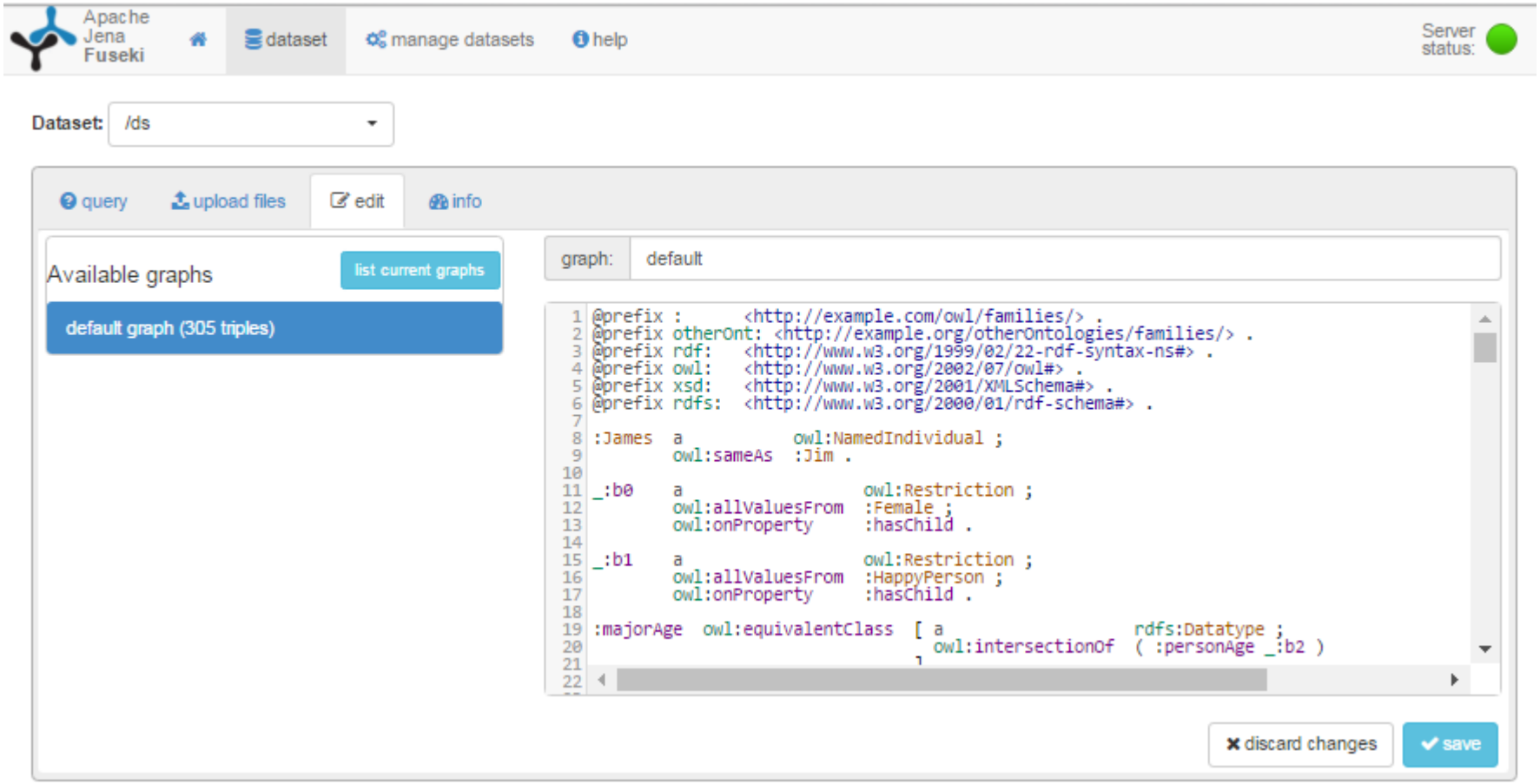
http://jena.apache.org/documentation/ontology/

Метод	Действие
add<property>	Добавляет значение данного свойства ресурсу
set<property>	Удаляет любые существующие значения для данного свойства и устанавливает заданное
list<property>	Возвращает итератор, пробегающий значения указанного свойства
get<property>	Возвращает значение для данного свойства, если ресурс имеет одно свойство. Если нет, то возвращать нуль. Если ресурс имеет более чем одно значение, возвращает произвольное.
has<property>	Возвращает истину, если есть хотя бы одно значение для данного свойства.
remove<property>	Удаляет заданное значение из значений свойства на этом ресурсе. Не имеет никакого эффекта, если ресурс не имеет это значение..

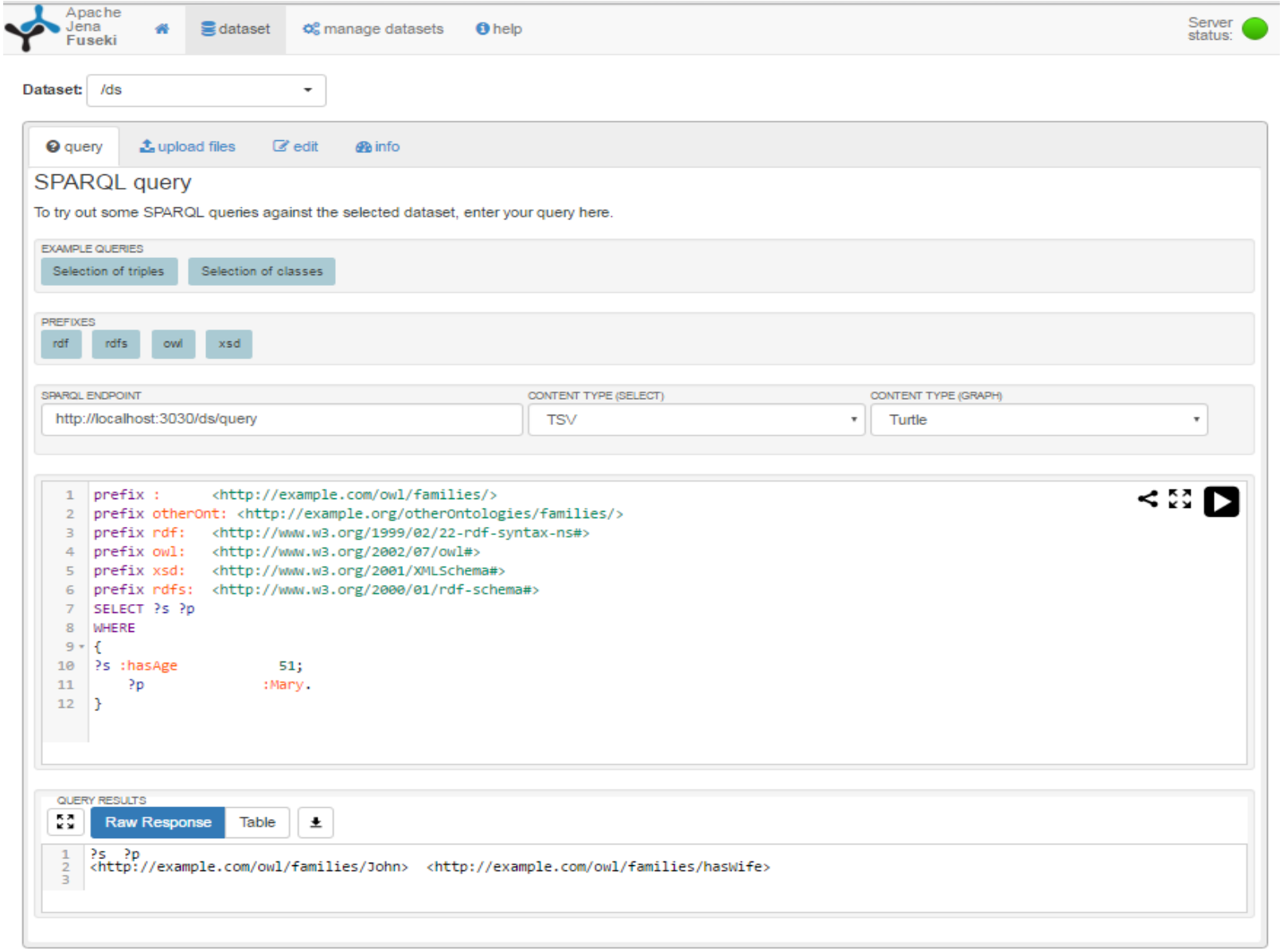
addSameAs(Resource r)
getCardinality(Property p)
setPropertyValue(Property p, RDFNode value)

Apache Jena Fuseki

13 Appendix: The Complete Sample Ontology



Apache Jena Fuseki



```
prefix :       <http://example.com/owl/families/>
prefix otherOnt: <http://example.org/otherOntologies/families/>
prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix owl:    <http://www.w3.org/2002/07/owl#>
prefix xsd:    <http://www.w3.org/2001/XMLSchema#>
prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
SELECT ?s ?p ?o
{
  ?s ?p ?o .
}
```

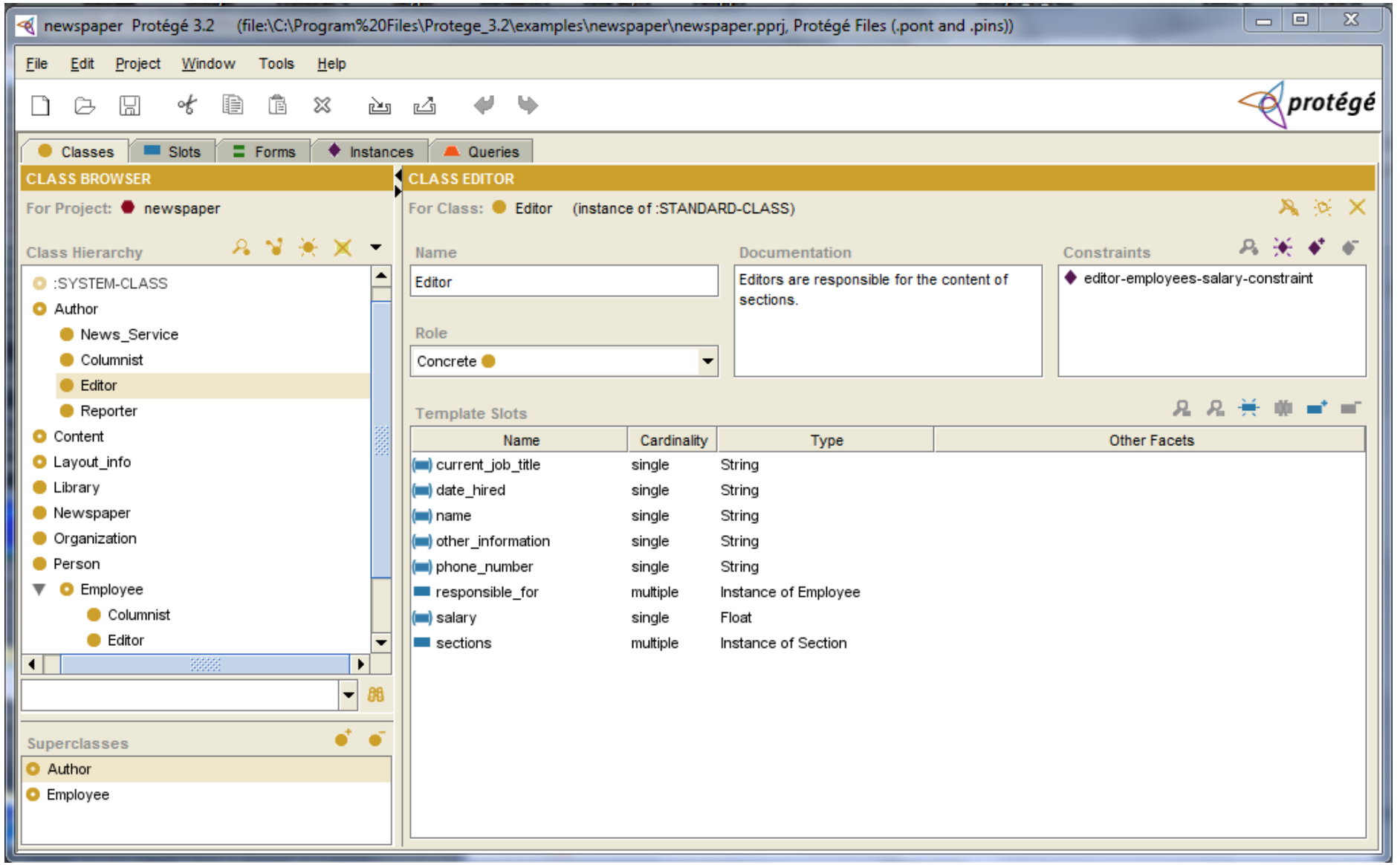
13 Appendix: The Complete Sample Ontology

QUERY RESULTS

Raw Response | Table

```
?s  ?p  ?o
_:B582efe5fX3A153e03fe069X3AX2D7fb1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2000/01/rdf-schema#Datatype>
_:B582efe5fX3A153e03fe069X3AX2D7fb1 <http://www.w3.org/2002/07/owl#intersectionOf>  _:B582efe5fX3A153e03fe069X3AX2D7fb0
<http://example.com/owl/families/James> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2002/07/owl#NamedIndividual>
<http://example.com/owl/families/James> <http://www.w3.org/2002/07/owl#sameAs>  <http://example.com/owl/families/Jim>
_:B582efe5fX3A153e03fe069X3AX2D7faa <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2002/07/owl#Class>
_:B582efe5fX3A153e03fe069X3AX2D7faa <http://www.w3.org/2002/07/owl#intersectionOf>  _:B582efe5fX3A153e03fe069X3AX2D7fa9
_:B582efe5fX3A153e03fe069X3AX2D7ff2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>  <http://example.com/owl/families/Person>
_:B582efe5fX3A153e03fe069X3AX2D7ff2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>  _:B582efe5fX3A153e03fe069X3AX2D7ff0
_:B582efe5fX3A153e03fe069X3AX2D7fc4 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2002/07/owl#Restriction>
_:B582efe5fX3A153e03fe069X3AX2D7fc4 <http://www.w3.org/2002/07/owl#onProperty>  <http://example.com/owl/families/hasChild>
_:B582efe5fX3A153e03fe069X3AX2D7fc4 <http://www.w3.org/2002/07/owl#allValuesFrom>  <http://example.com/owl/families/Female>
_:B582efe5fX3A153e03fe069X3AX2D7feb <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2002/07/owl#Restriction>
_:B582efe5fX3A153e03fe069X3AX2D7feb <http://www.w3.org/2002/07/owl#onProperty>  <http://example.com/owl/families/hasChild>
_:B582efe5fX3A153e03fe069X3AX2D7feb <http://www.w3.org/2002/07/owl#allValuesFrom>  <http://example.com/owl/families/HappyPerson>
_:B582efe5fX3A153e03fe069X3AX2D7fbd <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>  <http://example.com/owl/families/Father>
_:B582efe5fX3A153e03fe069X3AX2D7fbd <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>  _:B582efe5fX3A153e03fe069X3AX2D7fbc
_:B582efe5fX3A153e03fe069X3AX2D7fd7 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>  <http://example.com/owl/families/Man>
_:B582efe5fX3A153e03fe069X3AX2D7fd7 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>  _:B582efe5fX3A153e03fe069X3AX2D7fd6
_:B582efe5fX3A153e03fe069X3AX2D7ffe <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>  <http://example.com/owl/families/hasParent>
_:B582efe5fX3A153e03fe069X3AX2D7ffe <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#nil>
_:B582efe5fX3A153e03fe069X3AX2D7fa9 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>  <http://example.com/owl/families/Person>
_:B582efe5fX3A153e03fe069X3AX2D7fa9 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest>  _:B582efe5fX3A153e03fe069X3AX2D7fa7
<http://example.com/owl/families/majorAge>  <http://www.w3.org/2002/07/owl#equivalentClass>  _:B582efe5fX3A153e03fe069X3AX2D7fb1
<http://example.com/owl/families/Adult> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2002/07/owl#Class>
<http://example.com/owl/families/Adult> <http://www.w3.org/2002/07/owl#equivalentClass>  <http://example.org/otherOntologies/families/Grownup>
_:B582efe5fX3A153e03fe069X3AX2D7fd1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2002/07/owl#Restriction>
_:B582efe5fX3A153e03fe069X3AX2D7fd1 <http://www.w3.org/2002/07/owl#onProperty>  _:B582efe5fX3A153e03fe069X3AX2D7fd0
_:B582efe5fX3A153e03fe069X3AX2D7fd1 <http://www.w3.org/2002/07/owl#someValuesFrom>  <http://www.w3.org/2002/07/owl#Thing>
<http://example.com/owl/families/Mary>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://example.com/owl/families/Person>
<http://example.com/owl/families/Mary>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://example.com/owl/families/Woman>
<http://example.com/owl/families/Mary>  <http://www.w3.org/2002/07/owl#sameAs>  <http://example.org/otherOntologies/families/MaryBrown>
_:B582efe5fX3A153e03fe069X3AX2D7fa3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  <http://www.w3.org/2002/07/owl#Restriction>
_:B582efe5fX3A153e03fe069X3AX2D7fa3 <http://www.w3.org/2002/07/owl#cardinality> "\n          5\n"^^<http://www.w3.org/2001/XMLSchema#nonNegativeInteger>
_:B582efe5fX3A153e03fe069X3AX2D7fa3 <http://www.w3.org/2002/07/owl#onProperty>  <http://example.com/owl/families/hasChild>
_:B582efe5fX3A153e03fe069X3AX2D7fca <http://www.w3.org/1999/02/22-rdf-syntax-ns#first>  <http://example.com/owl/families/Meg>
```

Protégé



Области применения онтологий в Web
(1 – информационный поиск)

Information Retrieval (IR) - деятельность по сбору, организации, поиску, извлечению и распространению информации при помощи компьютерных технологий.

Примерами задач в области информационного поиска являются:

- информационный поиск документов по запросу пользователя;
- автоматическая рубрикация документов по заранее заданному рубрикатору;
- автоматическая кластеризация документов - разбиение на кластеры близких по смы слу документов;
- разработка вопросно-ответных систем - поиск точного фрагмента текста, отвечающ его на вопрос пользователя, а не целого документа;
- автоматическое составление аннотации документа;
- прогнозирование и многое другое.

Многие задачи совпадают с задачами Text Mining, но решаются они другими средствами – на основе семантики, добавленной к данным в виде онтологий.

Области применения онтологий в Web
(2 – интеграция разнородных источников знаний)

Базы данных содержат и способны обрабатывать большие массивы относительно простой информации (при этом доступ возможен только к этим явно введенным данным). В ***базах знаний*** обычно хранится меньший объем информации, но они имеют более сложную структуру, что позволяет использовать возможности ***логического вывода*** и получать такие утверждения, которые не были в явном виде введены.

Могут быть рассмотрены три важные задачи, возникающие при семантическом управлении данными:

- выражение концептуальной модели предметной области (онтологии предметной области) для конкретного источника данных;
- интеграция нескольких баз знаний при помощи объединения их онтологий;
- выражение и выполнение запросов к базам знаний.

Области применения онтологий в Web
(3 – семантический web в корпоративных сетях)

На жестких дисках LAN, MAN в корпорациях имеются огромные массивы разнотипных семантически не структурированных данных.

Традиционные методы обработки информации здесь бессильны.

Характерные семантические задачи в корпоративных сетях:

- поиск в разнотипных файлах (не только HTML), в корпоративных базах данных и системах документооборота,
- группировка тематически близких документов,
- автоматическое реферирование, перевод, выявление ключевых понятий,
- проведение нечеткого поиска,
- системы поддержки принятия управленческих решений.

Области применения онтологий в Web
(4 – базы знаний)

Основная задача при использовании онтологий в корпоративных сетях – это автоматизированное выявление ***знаний*** в массах данных (которые изначально не структурированы, семантически не связанны) с целью их использования в процессе принятия решения.

С этой целью информационные массивы (из Internet, MAN, LAN) преобразуются с семантической обработкой информации в хранилища данных ***Data Warehouse*** или ***базы знаний (данные + онтологии)***. Если такие базы открывают в Internet, то их называют ***порталами знаний***. Полученные базы позволяют значительно повысить:

- интеллектуальный анализ данных,
- глубинный анализ текстов,
- обнаружение новых знаний,
- принятие правильных решений,
- прогноз и тенденции событий…

Такие ИПС реализуются в программно-аппаратных комплексах Google Search Appliance, InfoStream Port и др. Стоимость внедрения таких систем – сотни тысяч долларов

4 LOD и FOAF

Ф.В.Филиппов, доцент кафедры ИУС
СПбГУТ им. проф. М.А.Бонч-Бруевича
9000096@mail.ru

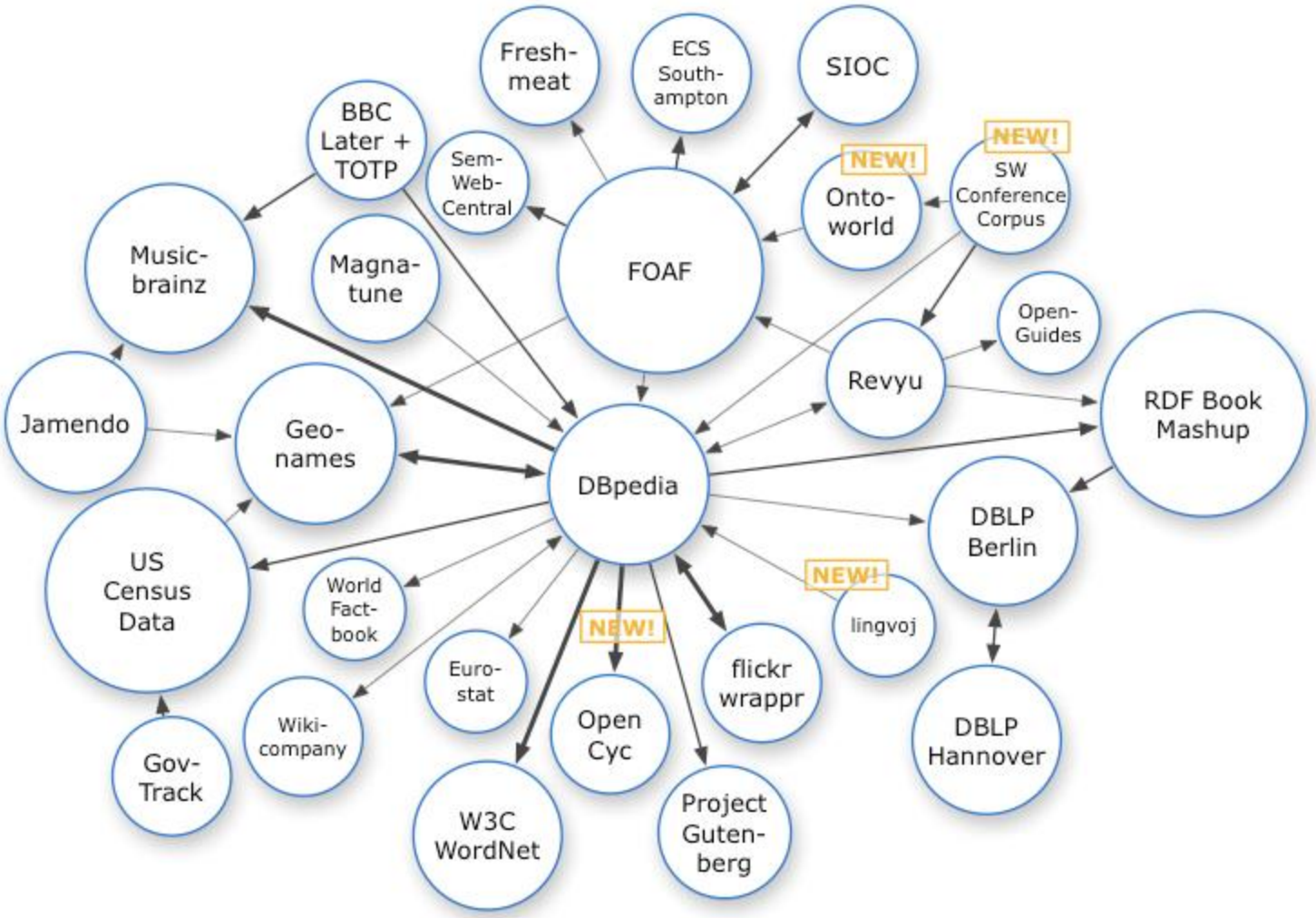
Linked Open Data (LOD)
звездная схема

★	Available on the web (whatever format) but with an open licence, to be Open Data
★★	Available as machine-readable structured data (e.g. excel instead of image scan of a table)
★★★	As (2) plus non-proprietary format (e.g. CSV instead of excel)
★★★★	All the above plus, use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff
★★★★★	All the above, plus: link your data to other people's data to provide context (связь)

Согласно звездной схеме, вы получаете одну звезду, если информация была обнародована вообще, т.е. она имеет открытую лицензию.

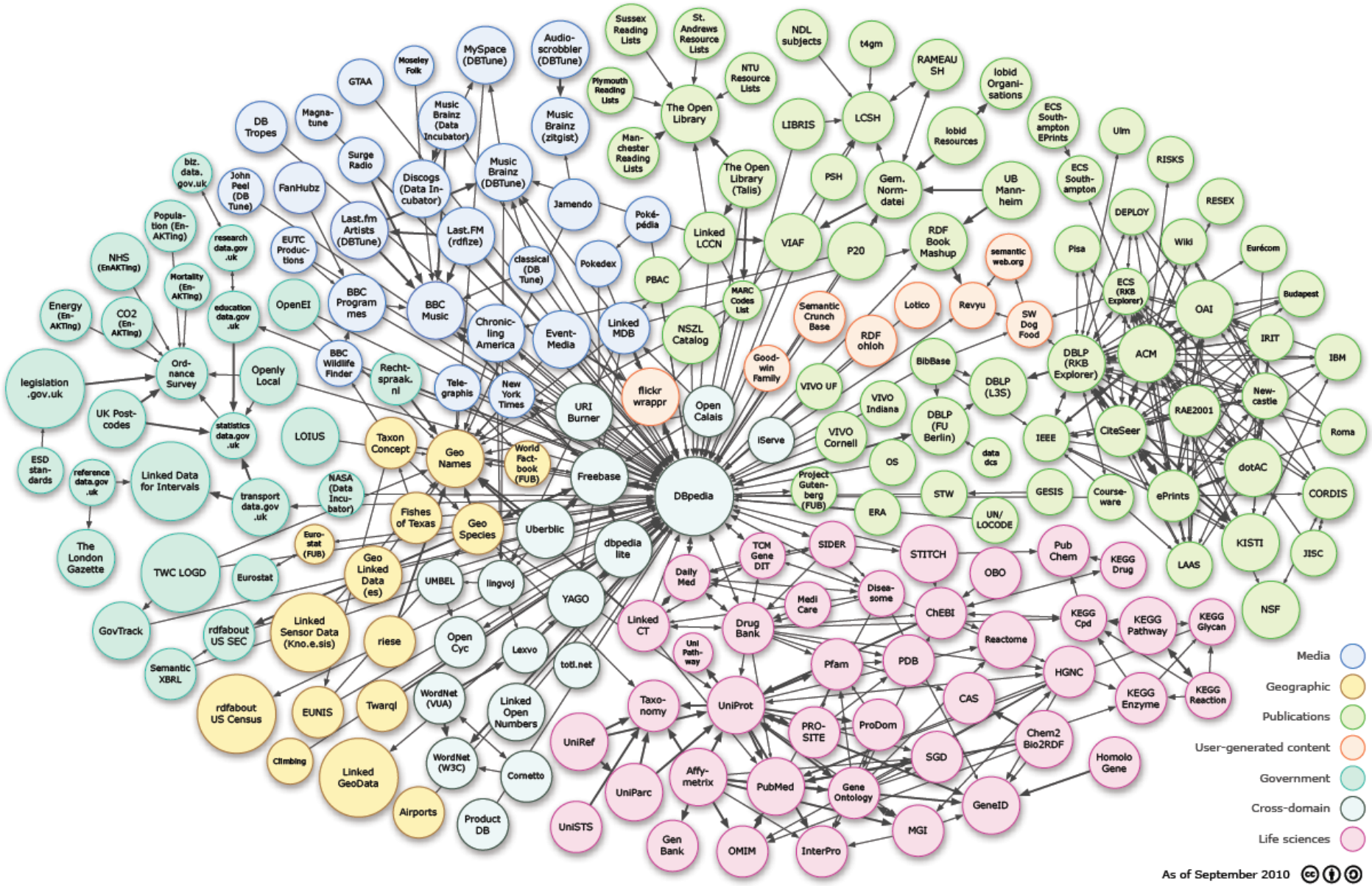
Вы получаете больше звезд, когда ваша информация становится все более доступной и проще для использования.

LOD 2007

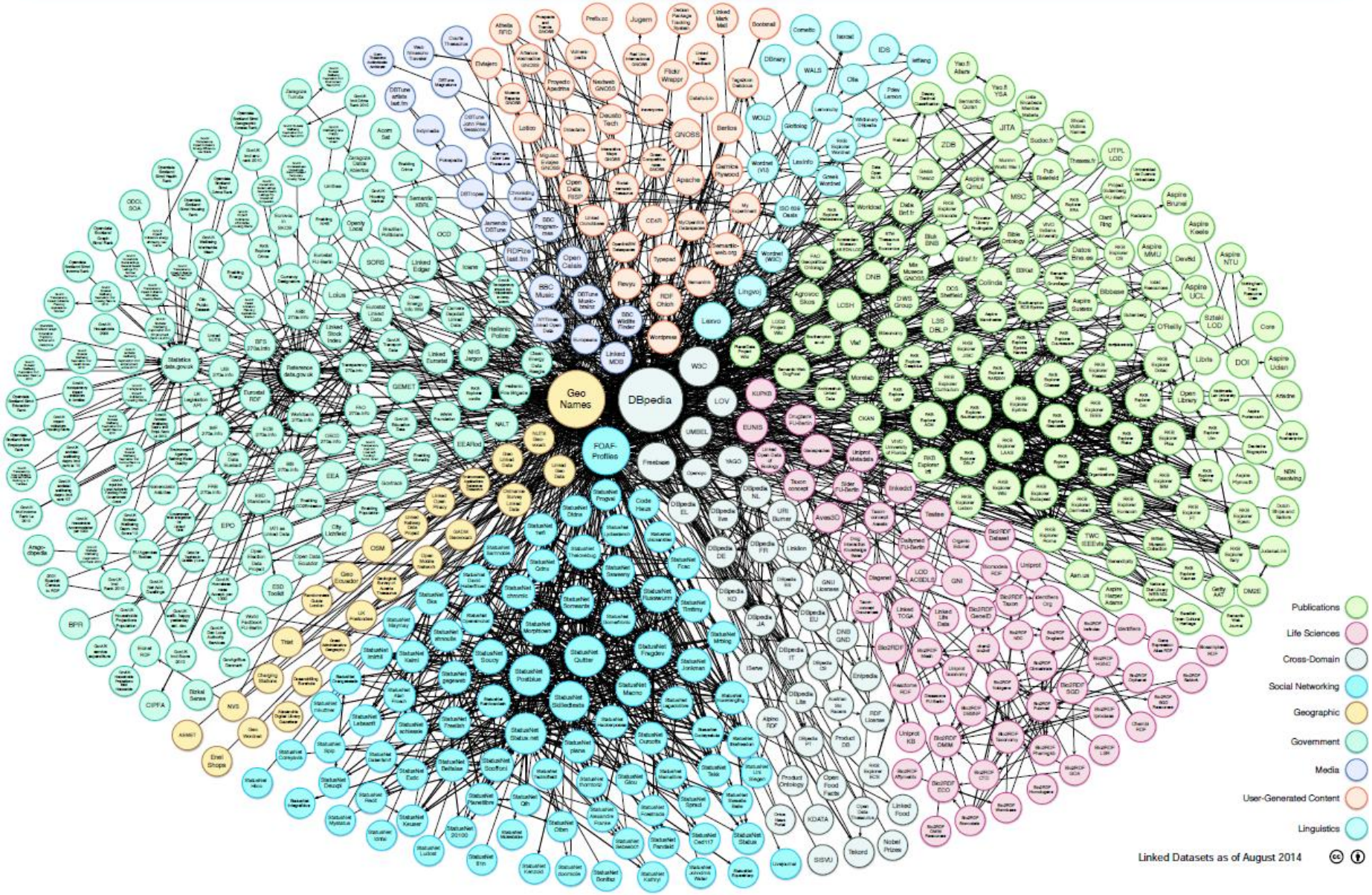


http://lod-cloud.net/

LOD 2010


DBpedia
The Open Library
LCSH
Linked LCCN
VIAF
P20
RDF Book Mashup
Gem. Norm-datei
UB Mann-heim
PSH
LIBRIS
NDL subjects
t4gm
RAMEAU SH
Iobid Organi-sations
Iobid Resources
The Open Library (Talis)
Sussex Reading Lists
St. Andrews Resource Lists
NTU Resource Lists
Plymouth Reading Lists
Man-chester Reading Lists
ECS South-ampton EPrints
ECS South-ampton
Ulm
RISKS
DEPLOY
RESEX
Wiki
Eurécom
Pisa
ECS (RKB Explorer)
OAI
Budapest
IRIT
IBM
New-castle
RAE2001
CiteSeer
Roma
dotAC
CORDIS
ePrints
KISTI
JISC
LAAS
NSF
Course-ware
GESIS
IEEE
DBLP (RKB Explorer)
ACM
DBLP (L3S)
DBLP (FU Berlin)
BibBase
SW Dog Food
semantic web.org
Revyu
Lotico
RDF ohloh
Semantic Crunch Base
Good-win Family
VIVO UF
VIVO Indiana
VIVO Cornell
IServe
NSZL Catalog
MARC Codes List
Linked MDB
PBAC
Poké-pédia
Pokedex
Jamendo
classical (DB Tune)
Music Brainz (zitgist)
Audio-scrobbler (DBTune)
MySpace (DBTune)
Music Brainz (DBTune)
Music Brainz (Data Incubator)
Discogs (Data In-cubator)
Last.FM (rdfize)
Last.fm Artists (DBTune)
Surge Radio
Magna-tune
Moseley Folk
GTAA
DB Tropes
FanHubz
John Peel (DB Tune)
biz. data. gov.uk
Popula-tion (En-AKTing)
NHS (EnAKTing)
Mortality (En-AKTing)
Energy (En-AKTing)
CO2 (En-AKTing)
research data.gov .uk
education data.gov .uk
EUTC Produc-tions
OpenEI
BBC Programmes
BBC Music
BBC Wildlife Finder
Chronic-ling America
Event-Media
Tele-graphis
New York Times
Recht-spraak. nl
URI Burner
flickr wrappr
Open Calais
Freebase
World Fact-book (FUB)
Geo Names
Taxon Concept
LOIUS
Openly Local
Ord-nance Survey
legislation .gov.uk
UK Post-codes
statistics data.gov .uk
ESD stan-dards
reference data.gov .uk
Linked Data for Intervals
transport data.gov .uk
NASA (Data Incu-bator)
Euro-stat (FUB)
Eurostat
The London Gazette
TWC LOGD
GovTrack
Semantic XBRL
rdfabout US SEC
Linked Sensor Data (Kno.e.sis)
rdfabout US Census
EUNIS
Climbing
Linked GeoData
Twarql
riese
Geo Linked Data (es)
Fishes of Texas
Geo Species
UMBEL
lingvoj
YAGO
Uberblic
dbpedia lite
Open Cyc
Lexvo
totl.net
WordNet (VUA)
WordNet (W3C)
Linked Open Numbers
Cornetto
Product DB
Airports
Project Guten-berg (FUB)
OS
STW
ERA
UN/ LOCODE
data dcs
TCM Gene DIT
Daily Med
Linked CT
Uni Path-way
Drug Bank
Medi Care
SIDER
Disea-some
STITCH
ChEBI
OBO
Pub Chem
KEGG Drug
KEGG Cpd
KEGG Pathway
KEGG Glycan
KEGG Enzyme
KEGG Reaction
Reactome
HGNC
CAS
Chem2 Bio2RDF
Homolo Gene
GeneID
MGI
InterPro
OMIM
Gen Bank
UniSTS
UniParc
UniRef
Taxo-nomy
UniProt
Affy-metrix
PubMed
Gene Ontology
PRO-SITE
ProDom
Pfam
PDB
SGD
Media
Geographic
Publications
User-generated content
Government
Cross-domain
Life sciences
As of September 2010

LOD 2014


Geo Names
DBpedia
FOAF-Profiles
Publications
Life Sciences
Cross-Domain
Social Networking
Geographic
Government
Media
User-Generated Content
Linguistics
Linked Datasets as of August 2014


Десятки миллиардов триплетов.
Сотни миллионов связей.

Связывание наборов данных

http://www.slideshare.net/iradche/linked-open-data-16524818

Проиндексированная статья в базе данных AGRIS

Связывание наборов данных

Субъект	Предикат	Объект
Ресурс 1	имеет заголовок	典型草原三种蝗虫种群死亡率的研究
Ресурс 1	имеет автора	Han Jianguo
Ресурс 1	имеет тему	Acrididae (grasshoppers)

Описание ресурса 1 в виде «условных» триплетов

Субъект	Предикат	Объект
agris:CN2009002389	имеет заголовок	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	имеет автора	Han Jianguo
agris:CN2009002389	имеет тему	Acrididae (grasshoppers)

«Триплифируем» описание ресурса 1

Ресурс 1 идентифицирован при помощи URI http://agris.fao.org/resource/CN2009002389.
Сократим его до agris:CN2009002389.

Связывание наборов данных

Субъект	Предикат	Объект
agris:CN2009002389	имеет заголовок	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	имеет автора	agris-author:hanjianguo
agris:CN2009002389	имеет тему	Acrididae (grasshoppers)

Автор идентифицирован при помощи URI http://agris.fao.org/author/hanjianguo.
Сократим его до agris-author:hanjianguo.

Сокращаем идентификатор

Субъект	Предикат	Объект
agris:CN2009002389	имеет заголовок	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	имеет автора	agris-author:hanjianguo
agris-author:hanjianguo	имеет имя	Han Jianguo
agris:CN2009002389	имеет тему	Acrididae (grasshoppers)

Автор agris-author:hanjianguo имеет имя Han Jianguo.

Определяем имя автора

Связывание наборов данных

Субъект	Предикат	Объект
agris:CN2009002389	имеет заголовок	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	имеет автора	agris-author:hanjianguo
agris-author:hanjianguo	имеет имя	Han Jianguo
agris:CN2009002389	имеет тему	agrovoc:c_4416

Тема идентифицируется при помощи URI http://aims.fao.org/aos/agrovoc/c_4416. Сократим ее до agrovoc:c_4416.

Сокращаем идентификатор темы

Субъект	Предикат	Объект
agris:CN2009002389	имеет заголовок	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	имеет автора	agris-author:hanjianguo
agris-author:hanjianguo	имеет имя	Han Jianguo
agris:CN2009002389	имеет тему	agrovoc:c_4416
agrovoc:c_4416	имеет пометку	Acrididae (en)
agrovoc:c_4416	имеет пометку	蝗科 (zh)

Описание http://aims.fao.org/aos/agrovoc/c_4416 в AGROVOC Concept Scheme говорит нам о том, как этот концепт на английском и китайском языках.

Идентифицируем язык

Связывание наборов данных

Субъект	Предикат	Объект
agris:CN2009002389	dct:title	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	имеет автора	agris-author:hanjianguo
agris-author:hanjianguo	имеет имя	Han Jianguo
agris:CN2009002389	имеет тему	agrovoc:c_4416
agrovoc:c_4416	имеет пометку	Acrididae (en)
agrovoc:c_4416	имеет пометку	蝗科 (zh)

Вводим префикс dct:

Элемент Dublin Core "Title" (заголовок) идентифицирован при помощи URI http://purl.org/dc/terms/title.
Сократим его до dct:title.

Субъект	Предикат	Объект
agris:CN2009002389	dct:title	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	dct:creator	agris-author:hanjianguo
agris-author:hanjianguo	foaf:name	Han Jianguo
agris:CN2009002389	dct:subject	agrovoc:c_4416
agrovoc:c_4416	имеет пометку	Acrididae (en)
agrovoc:c_4416	имеет пометку	蝗科 (zh)

Вводим другие префиксы

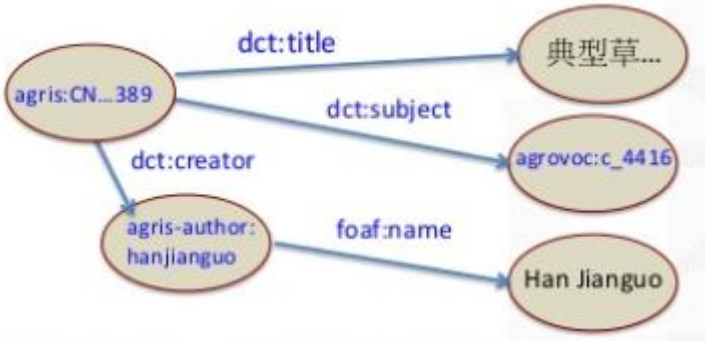
Повторяем то же самое для автора (author, creator) и темы (topic, subject).
Имя определено в словаре FOAF.

Связывание наборов данных

Субъект	Предикат	Объект
agris:CN2009002389	dct:title	典型草原三种蝗虫种群死亡率的研究
agris:CN2009002389	dct:creator	agris-author:hanjianguo
agris-author:hanjianguo	foaf:name	Han Jianguo
agris:CN2009002389	dct:subject	agrovoc:c_4416
agrovoc:c_4416	skos:prefLabel	Acrididae (en)
agrovoc:c_4416	skos:prefLabel	蝗科 (zh)

Свойство для пометок (preferred label) концепта определено в словаре Simple Knowledge Organization System (SKOS).

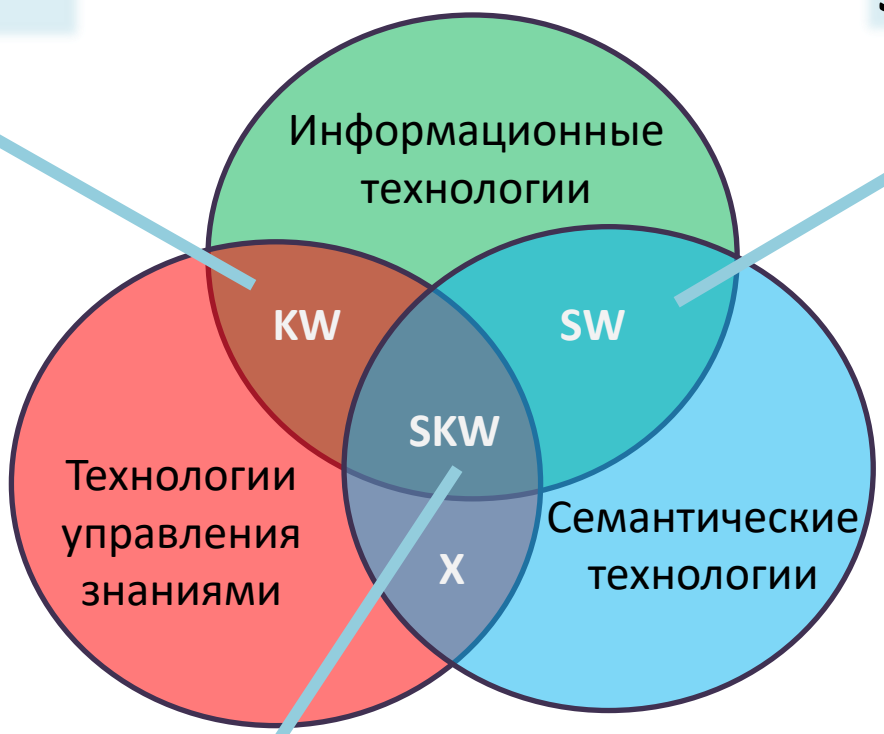
Полностью формализованные триплеты



Классификации Web порталов

Knowlege **W**eb

Semantic **W**eb



Тузовский А.Ф., Чириков С.В., Ямпольский В.З. Системы управления знаниями (методы и технологии) / Под общ. ред. В.З. Ямпольского. – Томск: Изд-во НТЛ, 2005. – 260 с.

Knowledge **S**emantic **W**eb

Онтология FOAF (пример SKW портала)

- **FOAF** (friend of a friend) – проект для создания сети машиночитаемых домашних страниц, описывающих людей, связи между ними, то, что они создают и чем занимаются.

- **FOAF** – онтология описания человека / личности

- Основное назначение – повысить мобильность пользователей социальных сетей

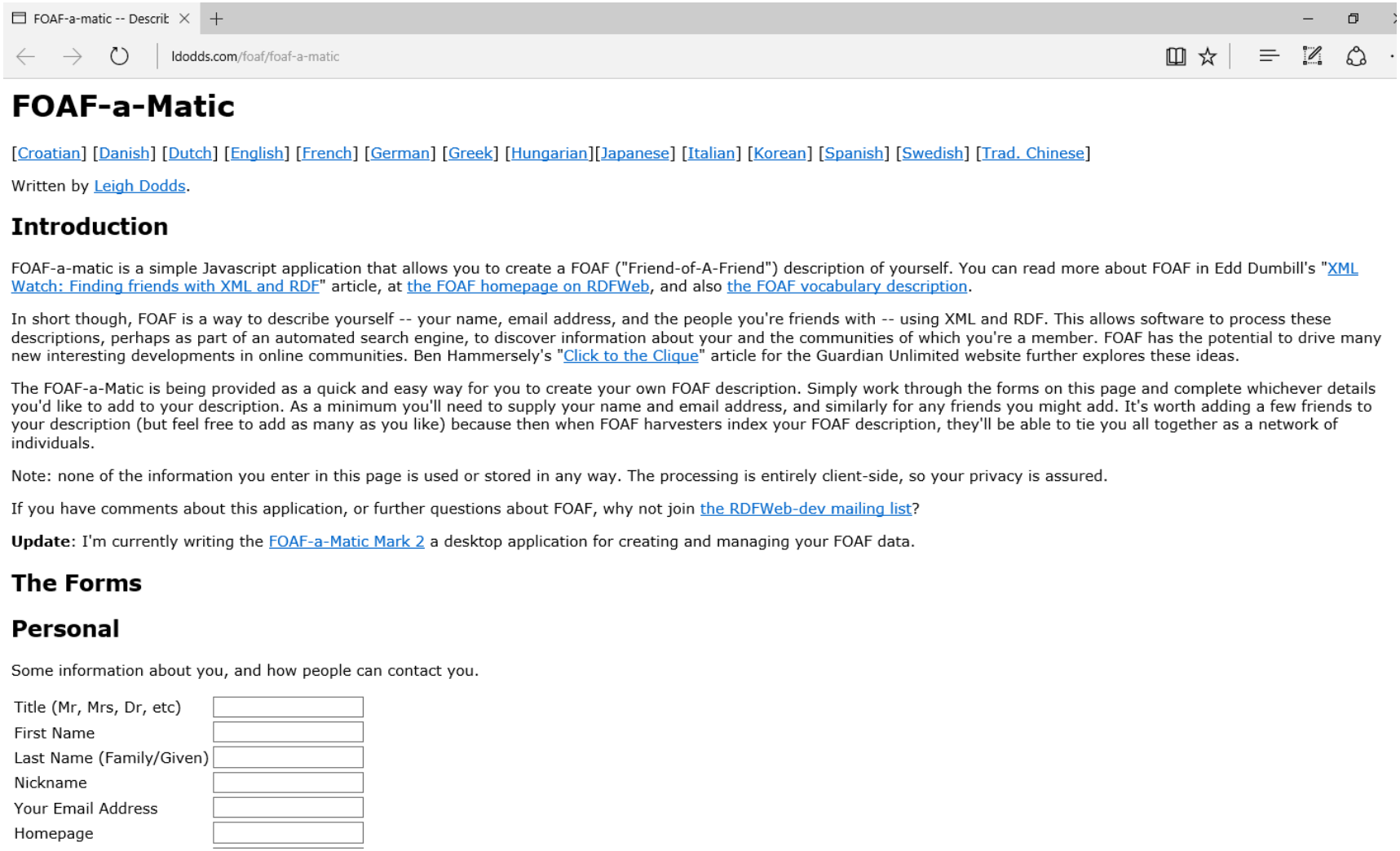
- Основа – RDF и XML

FOAF пример

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#«
xmlns:foaf="http://xmlns.com/foaf/0.1/«
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
        <foaf:person>
                <foaf:name>Иван Иванов</foaf:name>
                <foaf:mbox rdf:resource="mailto:ivanov@mail.ru" />
                <foaf:homepage rdf:resource=" www.ivanovpage.com/" />
                <foaf:nick>IvanIvanov</foaf:nick>
        </foaf:person>
</rdf:RDF>
```

FOAF-a-matic

Генерация FOAF-онтологий: http://www.ldodds.com/foaf/foaf-a-matic

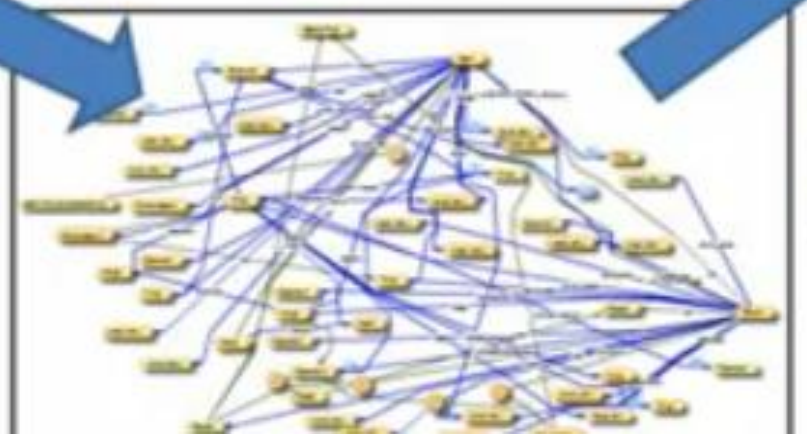


US , Ontology 101

ONTOLOGY

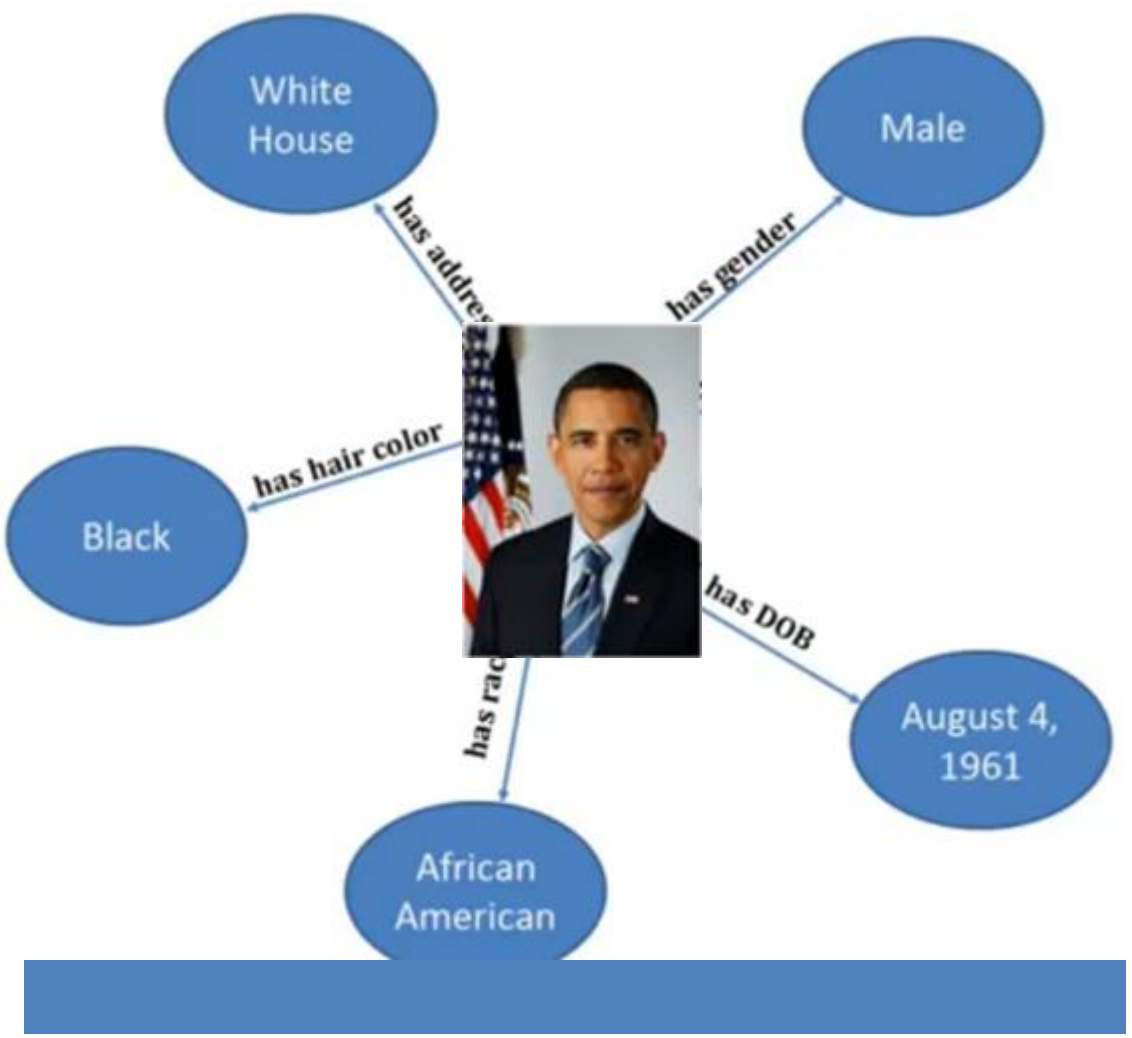
A DATA MODEL THAT REPRESENTS KNOWLEDGE AS A SET OF CONCEPTS WITHIN A DOMAIN AND THE RELATIONSHIPS BETWEEN THESE CONCEPTS

POLICIES
RULES
RELATIONSHIPS
DEFINITIONS
PROCESSES

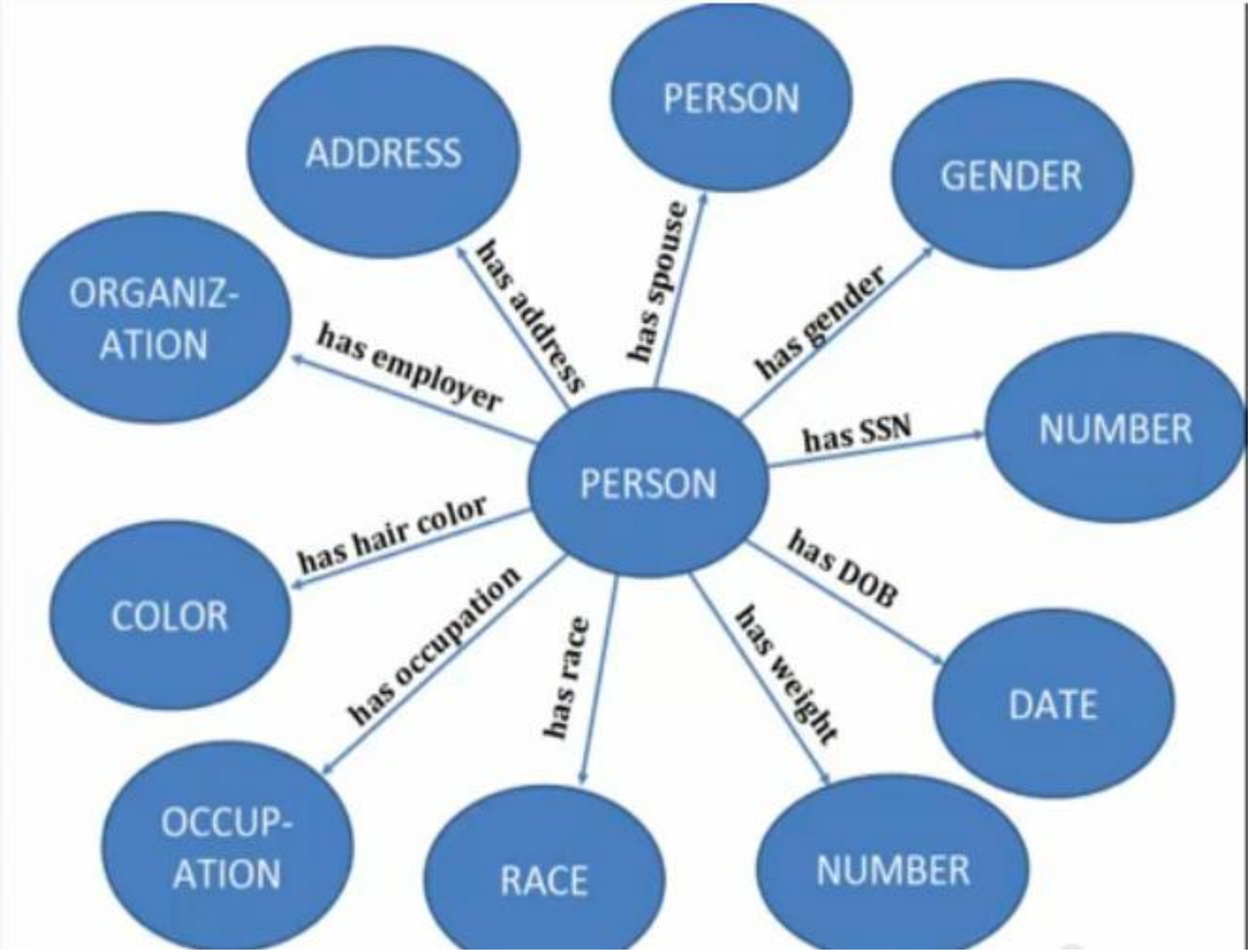


http://www.youtube.com/watch?feature=endscreen&v=jfUPLuPL3Ho&NR=1 - большое число видеопрезентаций

Who is mystery person?



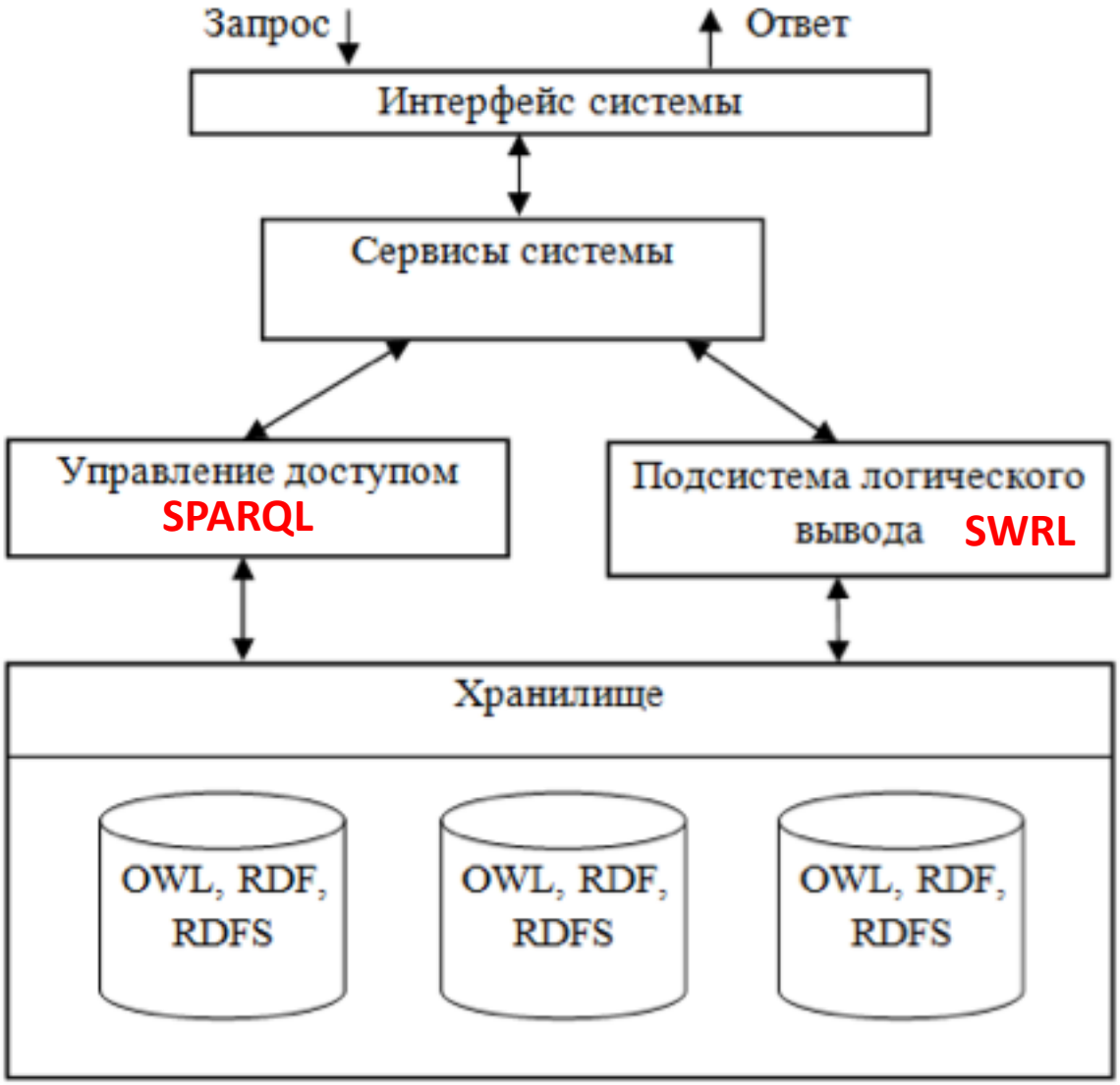
ONTOLOGIES are easily extensible



5 Семантические базы данных

Ф.В.Филиппов, доцент кафедры ИУС
СПбГУТ им. проф. М.А.Бонч-Бруевича
9000096@mail.ru

Semantic Data Base



$SDB = \{O, M, R\}$

O – онтология,

M – семантические метаданные,

R – множество логических правил.

$R = \{R_1, R_2\}$

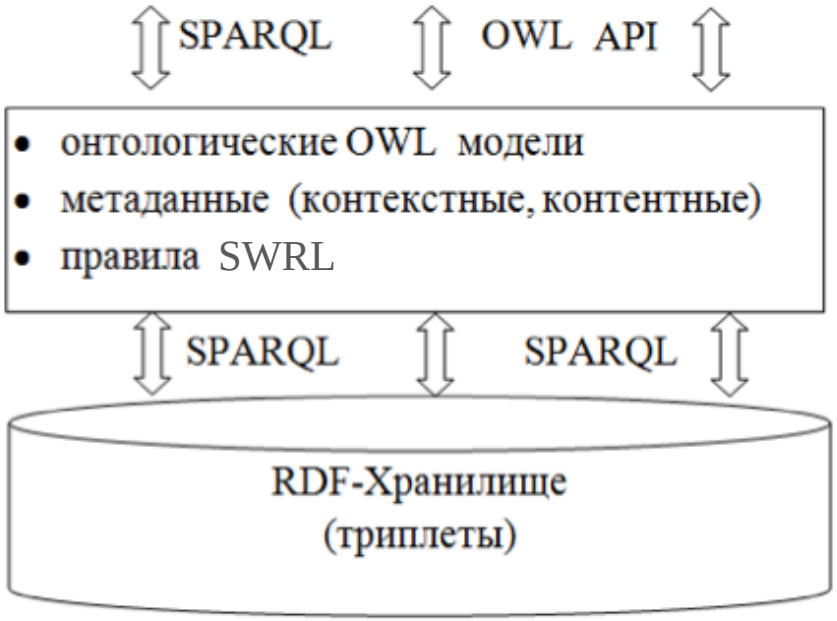
R_1 – множество правил в OWL,

R_2 – пользовательские правила.

Semantic Web Rule Language (SWRL)

Язык SWRL (*язык описания правил в семантическом вебе*) позволяет включать правила в описание онтологий OWL.

Правила SWRL не содержат конкретных объектов, а только ссылаются на них → можно применять одно и то же правило к нескольким группам объектов.



Примеры SWRL-правил

SWRL-правило: определение «быстрого компьютера»

FastComputer(?c) <- Computer(?c) ^ hasCPU(?c,?cpu) ^
hasSpeed(?cpu,?sp) ^ HighSpeed(?sp)

SWRL-правило: определение «дяди»

hasUncle(?x1,?x3) <- hasBrother(?x2,?x3) ^ hasParent(?x1,?x2)

http://www.daml.org/rules/proposal/

Предикаты в SWRL-правилах

Варианты предикатов, которые могут использоваться в правилах SWRL:

1. Утверждения о принадлежности к классу: Man(?x) -> Person (?x)
2. Утверждения о существовании связи: hasWife(Jack, ?x)
3. Утверждения о значении свойства-литерала:
   hasAge(?x , ?y) ∧ swrlb:greaterThanOrEqual(?y, 16)
4. Условие раздельности двух индивидуальных объектов:
   differentFrom (?x, ?y)
5. Условие совпадения двух объектов: sameAs(?x, ?y)
6. Условие о принадлежности значения переменной определенному типу данных: xsd:int (?x)
7. Встроенные условия (built-ins), например в пункте 3:
   swrlb:greaterThanOrEqual. Имеется большой набор таких функций, а также возможность определять собственные.

Системы управления СБД (СУ СБД)

- организация хранения RDF-данных;
- предоставление программного интерфейса для извлечения информации из хранимых RDF-данных посредством языка структурированных запросов SPARQL или специального интерфейса программирования приложений (application programming interface – API);
- поддержка функций администрирования хранимых данных: добавление, удаление, модификация и распределение прав доступа.

В настоящее время существует много различных СУ СБД, такие, как **Redland**, FreeBase, Sesame, Oracle 11g Release, **Virtuoso Universal Server**, **Apache Jena Fuseki**

- поддерживается возможность выполнения логических выводов для OWL-онтологий
- универсальный пользовательский интерфейс;
- поддержка хранилищ Quad-based (данные хранятся в виде квадов – кортежей из четырёх элементов <граф, субъект, предикат, объект>);

Переходим

к

SPARQL