Matteo Palmonari

matteo.palmonari@disco.unimib.it

Reasoning Agents:

Ontologies and Reasoning for RDF Knowledge Graphs RDFS

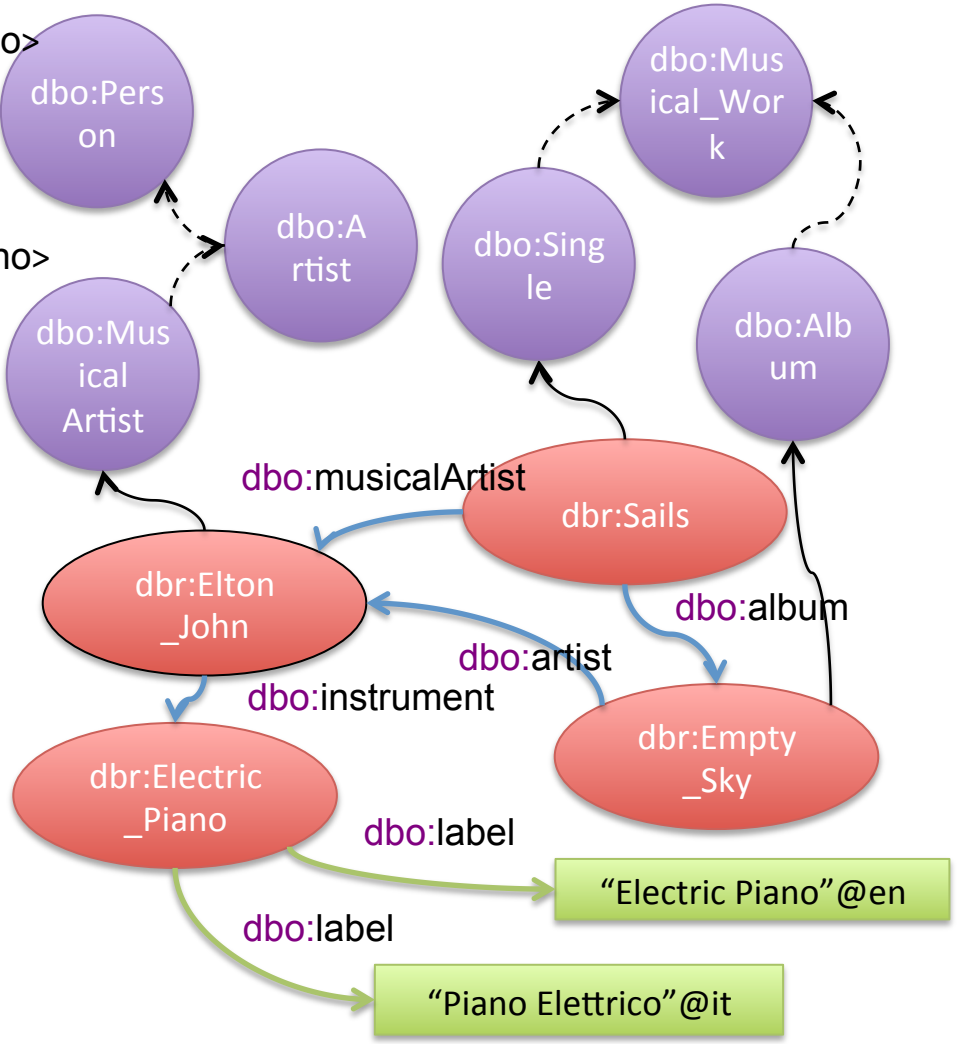
ITIS Lab – Innovative Technologies for Interaction and Services

Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano-Bicocca

RDF: URIfied Graphs and Triples

<dbr:Electric_Piano, rdfs:label, “Electric Piano”@en>
<dbr:Electric_Piano, rdfs:label, “Piano Elettrico”@it>
<dbr:Elton_John, dbo:instrument, dbr:Electric Piano>
<dbr:Empty_Sky, dbo:artist, dbr:Elton_John>
<dbr:Sails, dbo:musicalArtist, dbr:Elton_John>
<dbr:Sails, dbo:album, dbr:Empty_Sky>
<dbr:Elton_John, dbo:instrument, dbr:Electric_Piano>
<dbr:Elton_John, rdf:type, dbr:Musical_Artist>
<dbr:Sails, rdf:type, dbo:Single>
<dbr:Empty_Sky, rdf:type, dbo:Album>
<dbo:Musical_Artist, rdfs:subClassOf, dbo:Artist>
<dbo:Artist, rdfs:subClassOf, dbo:Person>
<dbo:Album, rdfs:subClassOf, dbo:Musical_Work>
<dbo:Single, rdfs:subClassOf, dbo:Musical_Work>

dbr: for http://dbpedia.org/resource/
e.g.: http://dbpedia.org/resource/Elton_John
dbo: for http://dbpedia.org/ontology/
rdfs for https://www.w3.org/2000/01/rdf-schema#
rdf for http://www.w3.org/1999/02/22-rdf-syntax-ns#



Vocabularies

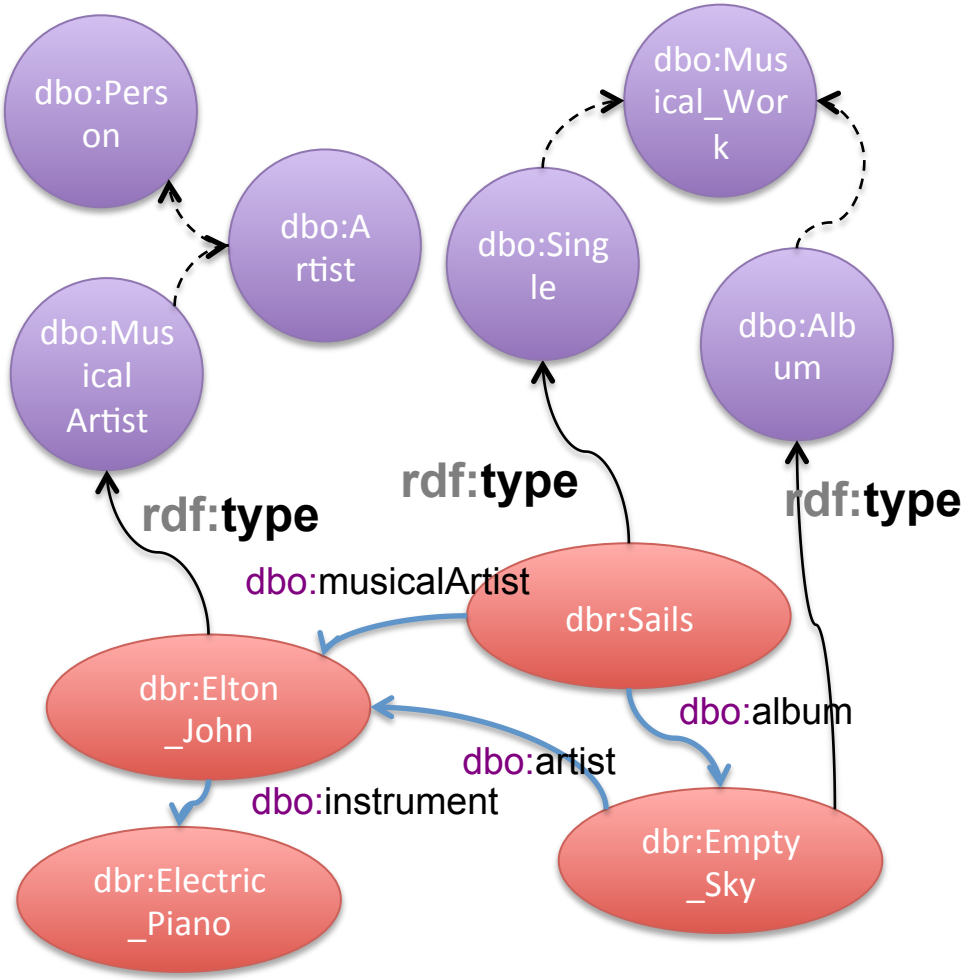
Schemas

Ontologies

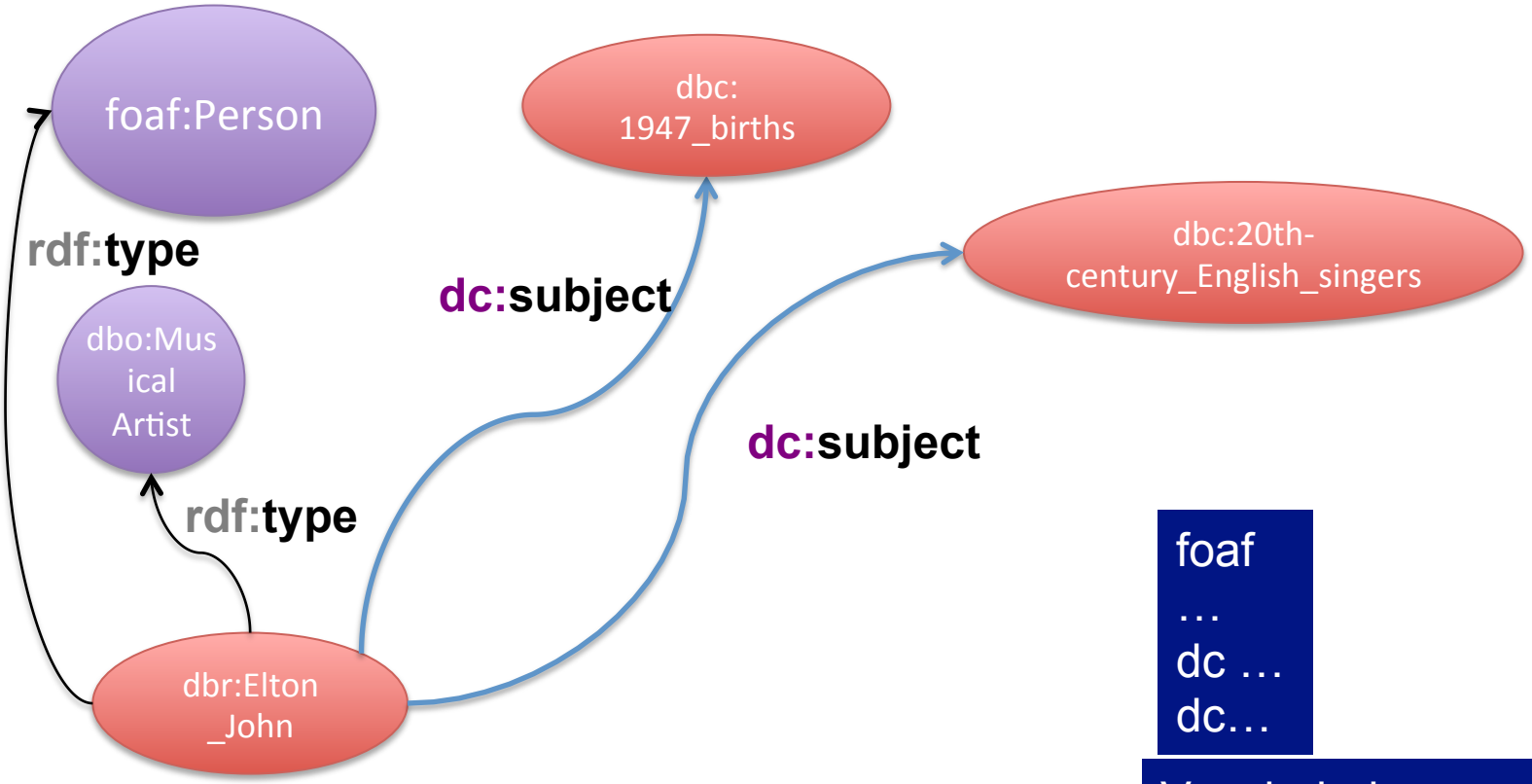
Categories by **rdf:type** assertions

The resource in the *subject* belongs to the **category** specified by the *object*

dbr:Elton_John **rdf:type** dbr:Musical_Artist .
dbr:Sails rdf:type dbo:Single .
dbr:Empty_Sky rdf:type dbo:Album .



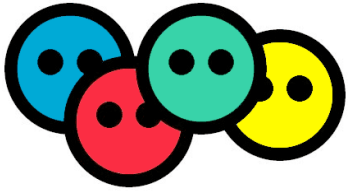
Shared Vocabularies with RDF



foaf
…
dc …
dc…

Vocabularies:
- Properties
- Categories
…with shared meaning

RDF in Action: FOAF



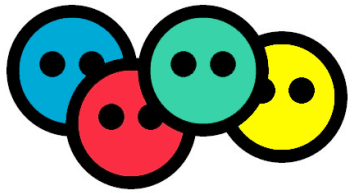
The *Friend of a Friend* (FOAF) project is creating a Web of machine-readable pages describing people, the links between them and the things they create and do

```
@prefix foaf: <http://purl.org/dc/elements/1.1/> .
<foaf:Person rdf:about="#danbri" xmlns:foaf="http://xmlns.com/foaf/0.1/">
<foaf:name>Dan Brickley</foaf:name>
<foaf:homepage rdf:resource="http://danbri.org/" />
<foaf:openid rdf:resource="http://danbri.org/" />
<foaf:img rdf:resource="/images/me.jpg" />
</foaf:Person>
```

example of FOAF
metadata describing Dan Brickley

http://www.foaf-project.org/
http://xmlns.com/foaf/spec/

RDF in Action: FOAF



The *Friend of a Friend* (FOAF) project is creating a Web of machine-readable pages describing people, the links between them and the things they create and do

```
@prefix foaf: <http://purl.org/dc/elements/1.1/> .
<foaf:Person rdf:about="#danbri" xmlns:foaf="http://xmlns.com/foaf/0.1/">
<foaf:name>Dan Brickley</foaf:name>
<foaf:homepage rdf:resource="http://danbri.org/" />
<foaf:openid rdf:resource="http://danbri.org/" />
<foaf:img rdf:resource="/images/me.jpg" />
</foaf:Person>
```

example of FOAF metadata describing Dan Brickley

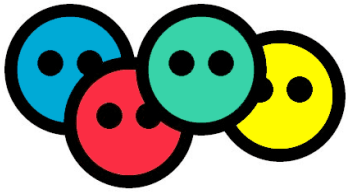
FOAF at a glance

An a-z index of FOAF terms, by class (categories or types) and by property.

Classes: | Agent | Document | Group | Image | LabelProperty | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | account | accountName | accountServiceHomepage | age | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | familyName | family_name | firstName | fundedBy | geekcode | gender | givenName | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | lastName | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | skypeID | status | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |

FOAF
Primitives



FOAF terms, grouped in broad categories.

FOAF Basics

- Agent
- Person
- name
- nick
- title
- homepage
- mbox
- mbox_sha1sum
- img
- depiction (depicts)
- surname
- familyName
- givenName
- firstName
- lastName

Personal Info

- weblog
- knows
- interest
- currentProject
- pastProject
- plan
- based_near
- age
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- topic_interest
- publications
- geekcode
- myersBriggs
- dnaChecksum

Online Accounts / IM

- OnlineAccount
- OnlineChatAccount
- OnlineEcommerceAccount
- OnlineGamingAccount
- account
- accountServiceHomepage
- accountName
- icqChatID
- msnChatID
- aimChatID
- jabberID
- yahooChatID
- skypeID

Projects and Groups

- Project
- Organization
- Group
- member
- membershipClass

Documents and Images

- Document
- Image
- PersonalProfileDocument
- topic (page)
- primaryTopic (primaryTopicOf)
- tipjar
- sha1
- made (maker)
- thumbnail
- logo

Try the "FOAF generator" Facebook application

RDF in Action: DC

The Dublin Core (DC) metadata element set is a **standard** for **cross-domain** information resource description.

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
urn:pp:20070426 dc:creator "Andrea Bonomi" .
urn:pp:20070426 dc:subject "RDF" .
urn:pp:20070426 dc:description “An introduction to RDF” .
urn:pp:20070426 dc:date "26 Apr 2007" .
urn:pp:20070426 dc:dateCopyrighted "13 Apr 2007" .
urn:pp:20070426 cc:License cc:by-sa-1.0 .
```

example of CC metadata for this presentation

The most common metadata elements used by Dublin Core are:

* title (the name given the resource)
* creator (the person or organization responsible for the content)
* subject (the topic covered)
* description (a textual outline of the content)
* publisher (those responsible for making the resource available)
* contributor (those who added to the content)
* date (when the resource was made available)
* type (a category for the content)
* format (how the resource is presented)
* identifier (numerical identifier for the content such as a URL)
* source (where the content originally derived from)
* language (in what language the content is written)
* relation (how the content relates to other resources for instance, if it is a chapter in a book)
* coverage (where the resource is physically located)
* rights (a link to a copyright notice)

Dublin Core Metadata Initiative®

http://dublincore.org/

Describing Data

Vocabularies

A set of well-known vocabularies has evolved in the Semantic Web community. **Some** of them are:

Vocabulary	Description	Classes and Relationships
Friend-of-a-Friend **(FOAF)**	Vocabulary for describing people.	foaf:Person, foaf:Agent, foaf:name, foaf:knows, foaf:member
Dublin Core **(DC)**	Defines general metadata attributes.	dc:FileFormat, dc:MediaType, dc:creator, dc:description
Semantically-Interlinked Online Communities **(SIOC)**	Vocabulary for representing online communities.	sioc:Community, sioc:Forum, sioc:Post, sioc:follows, sioc:topic
Music Ontology **(MO)**	Provides terms for describing artists, albums and tracks.	mo:MusicArtist, mo:MusicGroup, mo:Signal, mo:member, mo:record
Simple Knowledge Organization System **(SKOS)**	Vocabulary for representing taxonomies and loosely structured knowledge.	skos:Concept, skos:inScheme, skos:definition, skos:example

Linked Data Principles

EUCLID
EdUcational Curriculum for the usage of LInked Data

4. Include links to other URIs, so that users can **discover** more things.

There are several ways to reuse URIs:

- direct **reuse**
- (OWL) **sameAs**
- (RDFS) **seeAlso**

} Instance Level

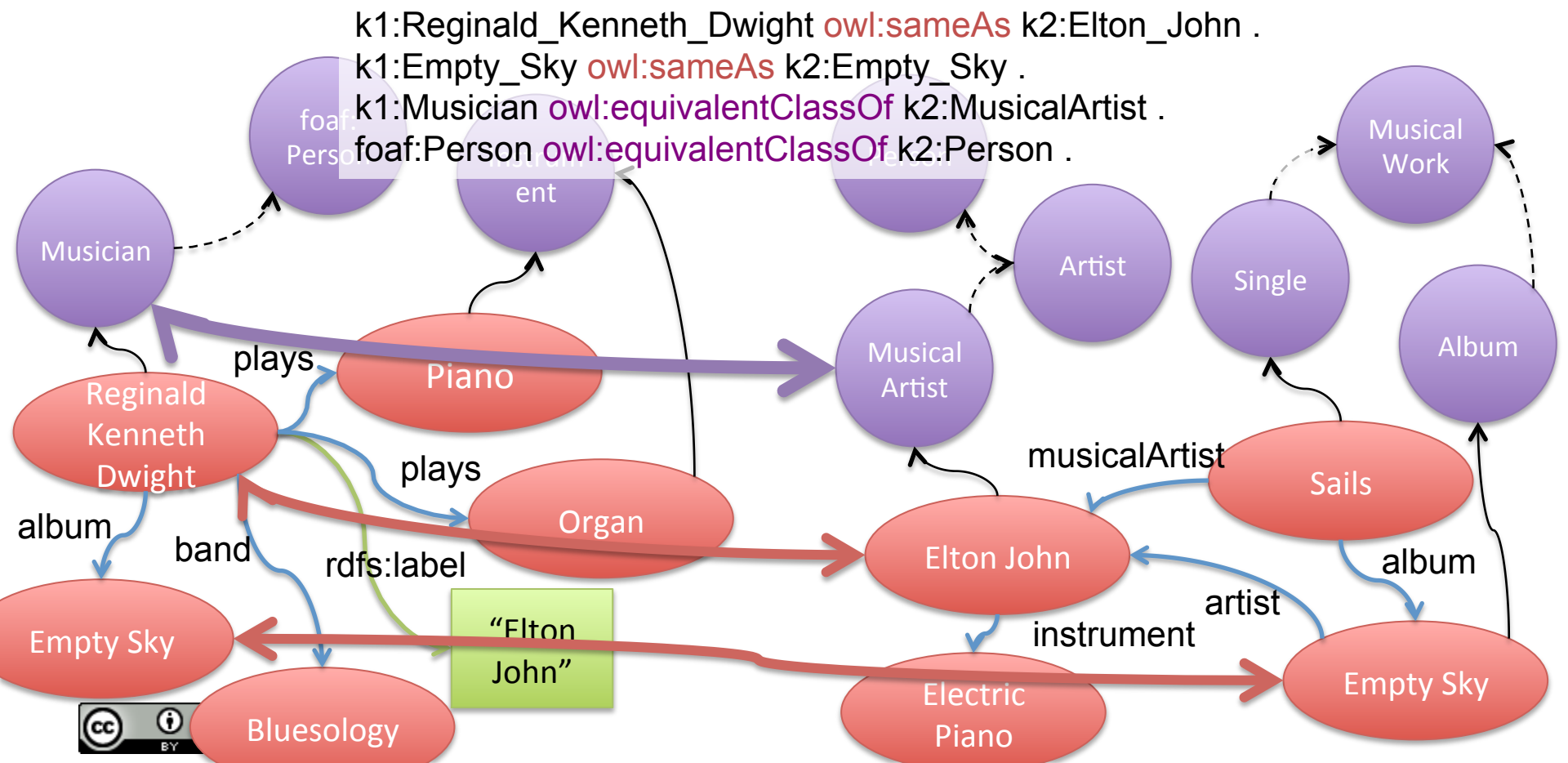
- direct **reuse** of class/property
- (RDFS) **sub**-class/-property
- (OWL) **equivalent** class/property
- SKOS **broad match**

} Schema Level

Reuse URIs and links between resources

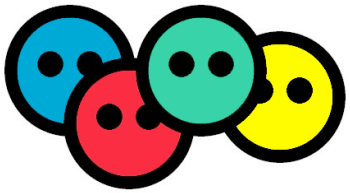
EUCLID
EdUcational Curriculum for the usage of LInked Data

- **Map** two KGs so as to make one connected KG



FOAF Primitives

FOAF terms, grouped in broad categories.



FOAF Basics

- Agent
- Person
- name
- nick
- title
- homepage
- mbox
- mbox_sha1sum
- img
- depiction (depicts)
- surname
- familyName
- givenName
- firstName
- lastName

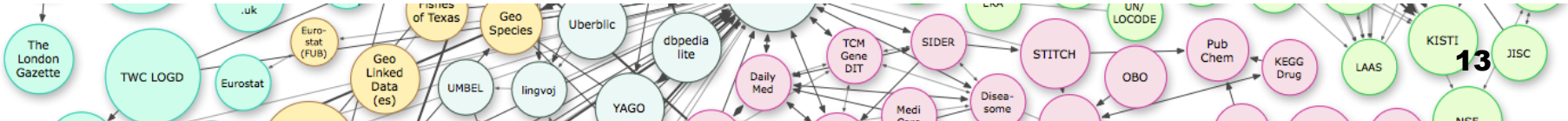
Personal Info

- weblog
- knows
- interest
- currentProject
- pastProject
- plan
- based_near
- age
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- topic_interest
- publications
- geekcode
- myersBriggs
- dnaChecksum

Online Accounts / IM

- OnlineAccount
- OnlineChatAccount
- OnlineEcommerceAccount
- OnlineGamingAccount
- account
- accountServiceHomepage
- accountName
- icqChatID
- msnChatID
- aimChatID
- jabberID
- yahooChatID
- skypeID

What is a Person?



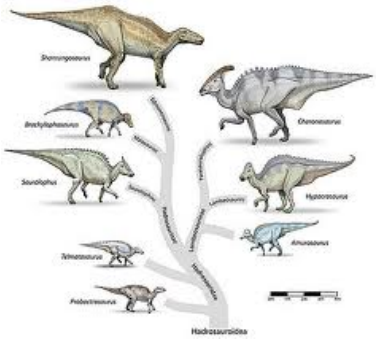
Different types of ontologies

Tesauri – Lexical Ontologies NLP

A thesaurus defines a dictionary for a language, specifying for some linguistic entities (e.g. nouns, verbs, adjectives), the **words** that are part of the language (e.g. "tree") and a fixed number of **relations between words** (e.g. synonyms, hypernyms, etc.). **Concepts** are defined **as sets of words** that can have similar meaning in some contexts.

Classification schemes

Taxonomies



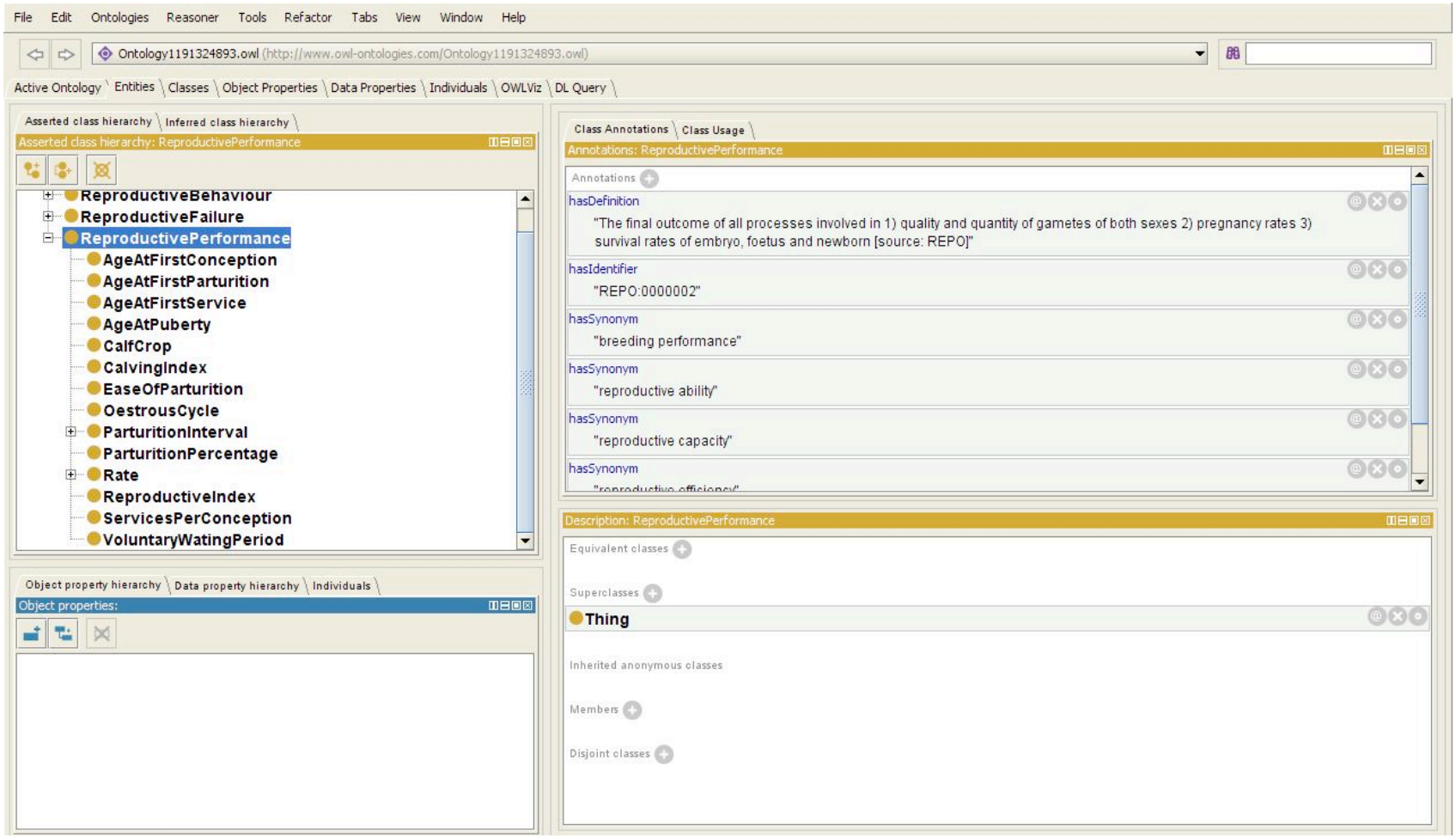
Classification structure for objects of a given domain. Usually **hierarchies** with tree-like structures.

Axiomatic Ontologies

Schemas / models

Formal specifications of a conceptualization with a **language *L***, a set of **logical axioms** in *L*, and a **formal semantics** that univocally specifies their meaning. Usually axiomatic ontologies organize knowledge by defining *individuals*, *classes*, *relations*, their properties and logical languages that characterize them

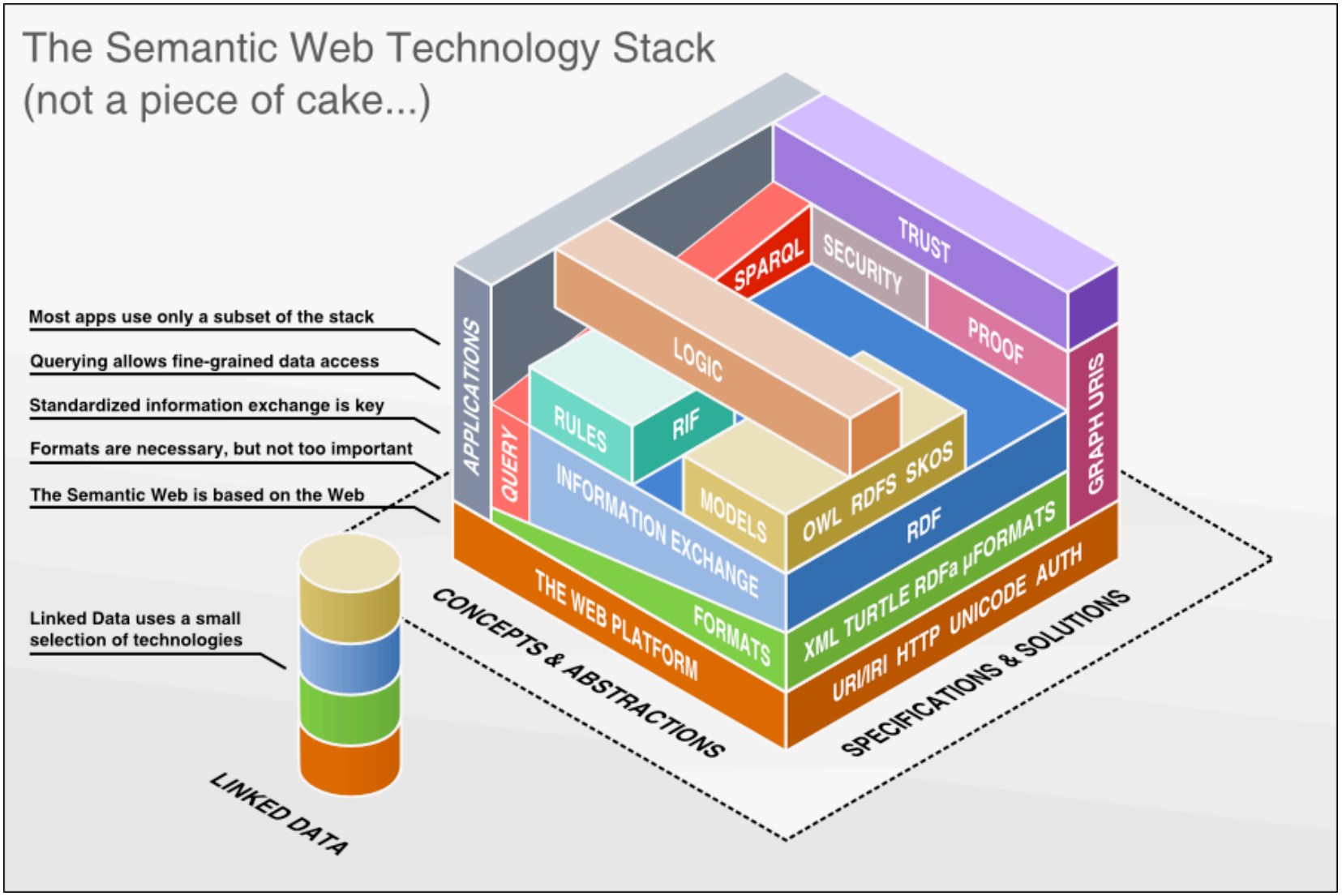
Costruire un ontologia con Protegé



Open source / free tool available at: http://protege.stanford.edu/

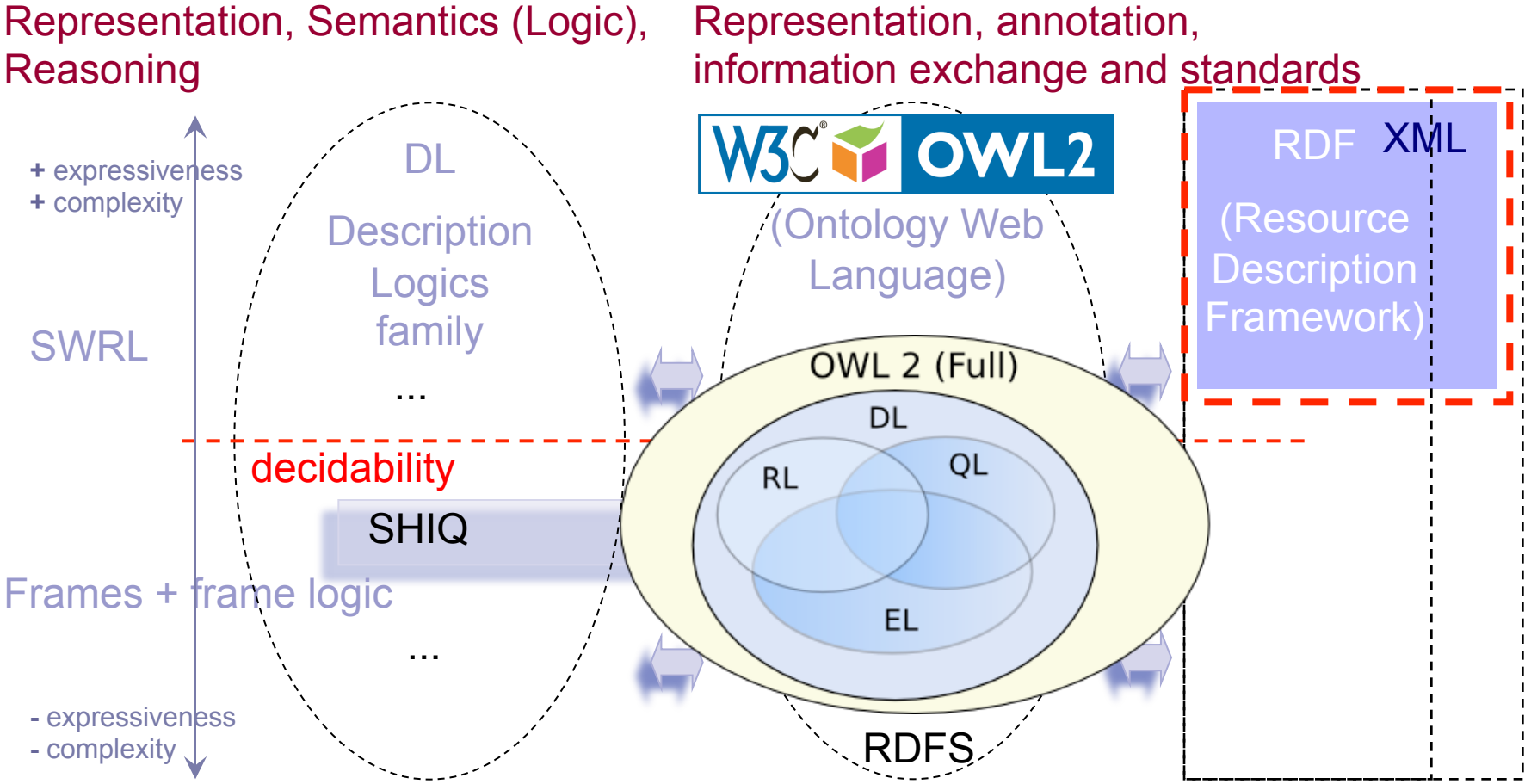
RDFS and OWL Reasoning

Semantic Web Stack Revised



http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake

Ontologies as "Semantic Technologies" (revised)



Ontologies: Editing, Reasoning, Navigation, Query

RDF Schema

RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent **relationships between resources**.

RDF however, provides no mechanisms for describing these properties, nor does it provide any mechanisms for describing the relationships between these properties and other resources. That is the role of the RDF vocabulary description language, RDF Schema. RDF Schema defines **classes** and **properties** that may be used to describe **classes**, **properties** and other **resources**.

RDF Vocabulary Description Language 1.0: RDF Schema

rdfs:Resource

All things described by RDF are called **resources**, and are instances of the class **rdfs:Resource**.

rdfs:Class

Classes are **groups of resources**.
The members of a class are known as **instances**.
Classes are **themselves resource.**The **rdfs:subClassOf** property may be used to state that one class is a subclass of another

rdfs:Literal

The class **rdfs:Literal** is the class of literal values such as **strings** and **integers**. Property values such as textual strings are RDF literals.

rdf:Property

rdf:Property is the class of RDF properties

RDFS taxonomies

- A taxonomy can be represented as a light-weight axiomatic ontology.
- Class hiearchies
  - **rdfs:class** and **rdfs:subClassOf**
- Property hierarchies
  - **rdfs:property** and **rdfs:subPropertyOf**
- Domain and Range restrictions on properties
  - **rdfs:domain** and **rdfs:range**
- RDFS taxonomies can exploit very basic reasoning tasks (IS-A, membership to classes, domain/range checking)

RDF-S semantics (a part of it)

if	then
`x rdfs:subClassOf y .` `a rdf:type x .`	`a rdf:type y .`
`x rdfs:subClassOf y .` `y rdfs:subClassOf z .`	`x rdfs:subClassOf z .`
`x a y .` `a rdfs:subPropertyOf b .`	`x b y .`
`a rdfs:subPropertyOf b .` `b rdfs:subPropertyOf c .`	`a rdfs:subPropertyOf c .`
`x a y .` `a rdfs:domain z .`	`x rdf:type z .`
`x a u .` `a rdfs:range z .`	`u rdf:type z .`

Read out more in RDF Semantics http://www.w3.org/TR/rdf-mt/

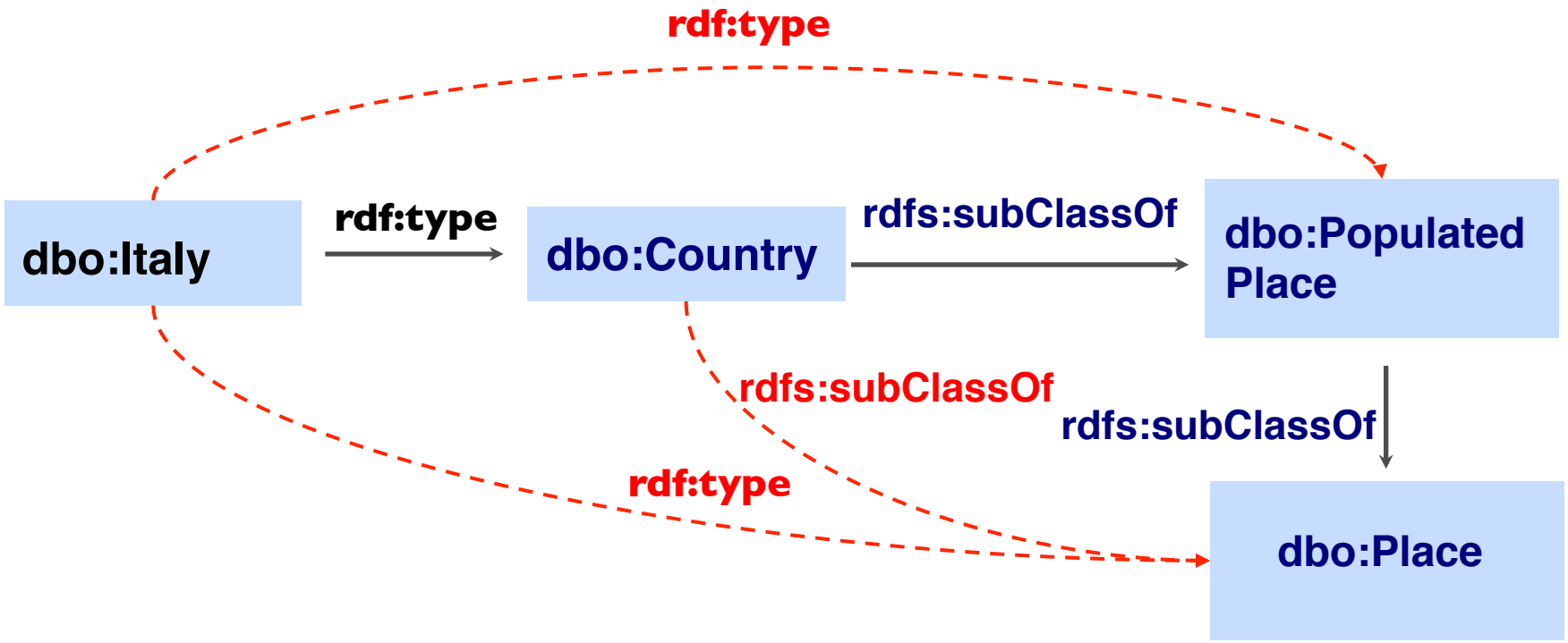
RDFS Rule List

EUCLID
EdUcational Curriculum for the usage of LInked Data

- Contains 13 entailment rules denominated rdfs*i* for inference over RDFS definitions*:
  - rdfs:Literal (rdfs1, rdfs13)
  - rdfs:domain (rdfs2), rdfs:range (rdfs3)
  - rdfs:Resource (rdfs4a, rdfs4, rdfs8)
  - rdfs:subPropertyOf (rdfs5, rdfs6, rdfs7, rdfs12)
  - rdfs:Class (rdfs8, rdfs10)
  - rdfs:subClassOf (rdfs9, rdfs10, rdfs11)
  - rdfs:ContainerMembershipProperty (rdfs12)
  - rdfs:Datatype (rdfs13)

* Source: http://www.w3.org/TR/rdf-mt/#RDFSRules

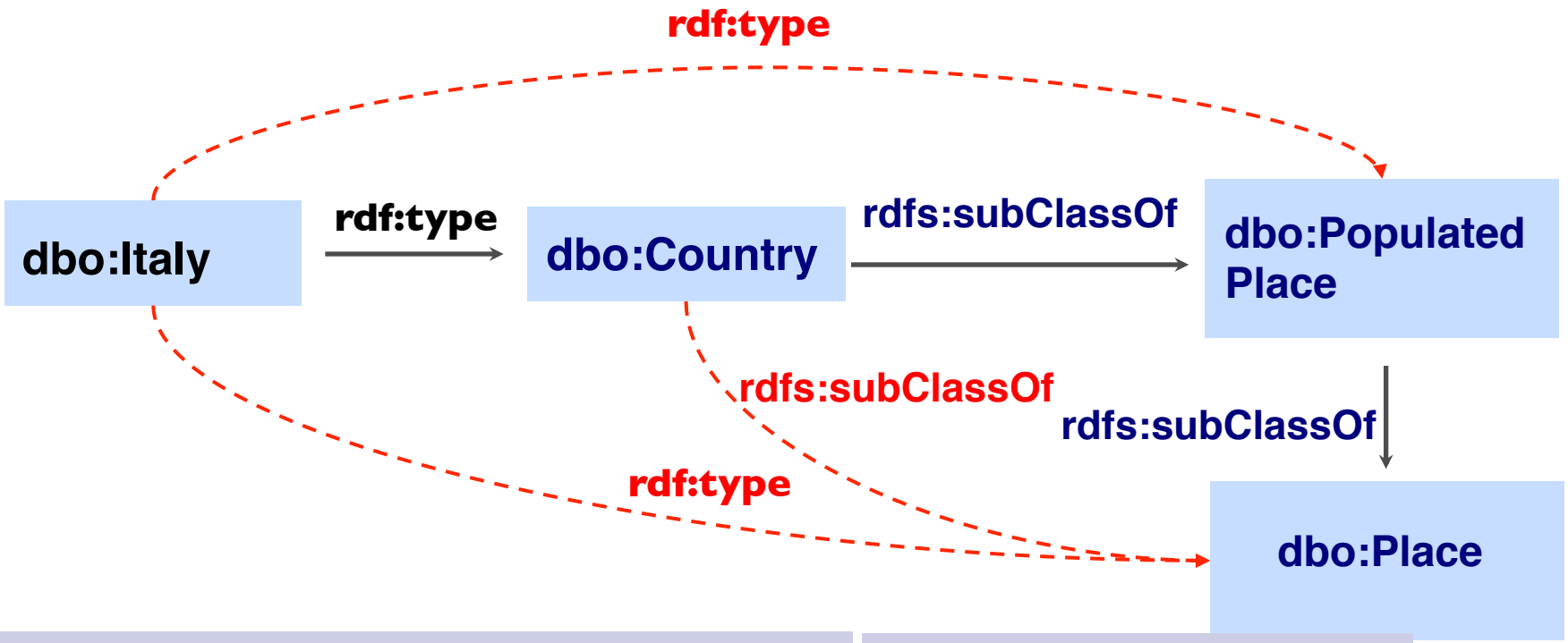
Minimal Reasoning with RDFS



If a Country is a Populated Place **and** a Populated Place is a Place,
then a Country is also a Place

If Italy is a Country **and**
a Country is a Populated Place **and**
a Populated Place is a Place,
then Italy is a also a Populated Place and a Place

Minimal Reasoning with RDFS



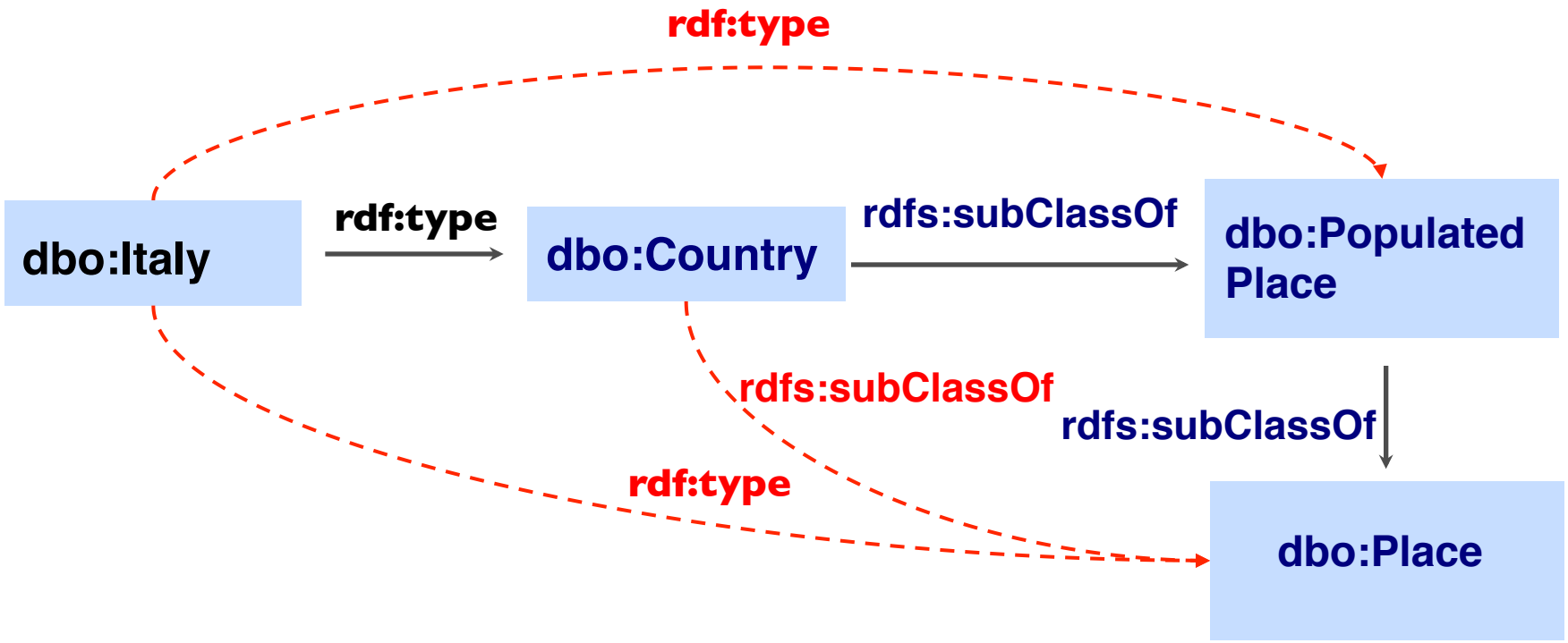
If C is *subclass of* D **and** D is *subclass of* E, **then** a C *is subclass of* E

If x is *of type* C **and** C is *subclass of* D **then** x is also *of type* D

In addition…
C is *subclass of* C

If C is *subclass of* D **and** D is *subclass of* C **then** a C *is equivalent to* D

Minimal Reasoning with RDFS

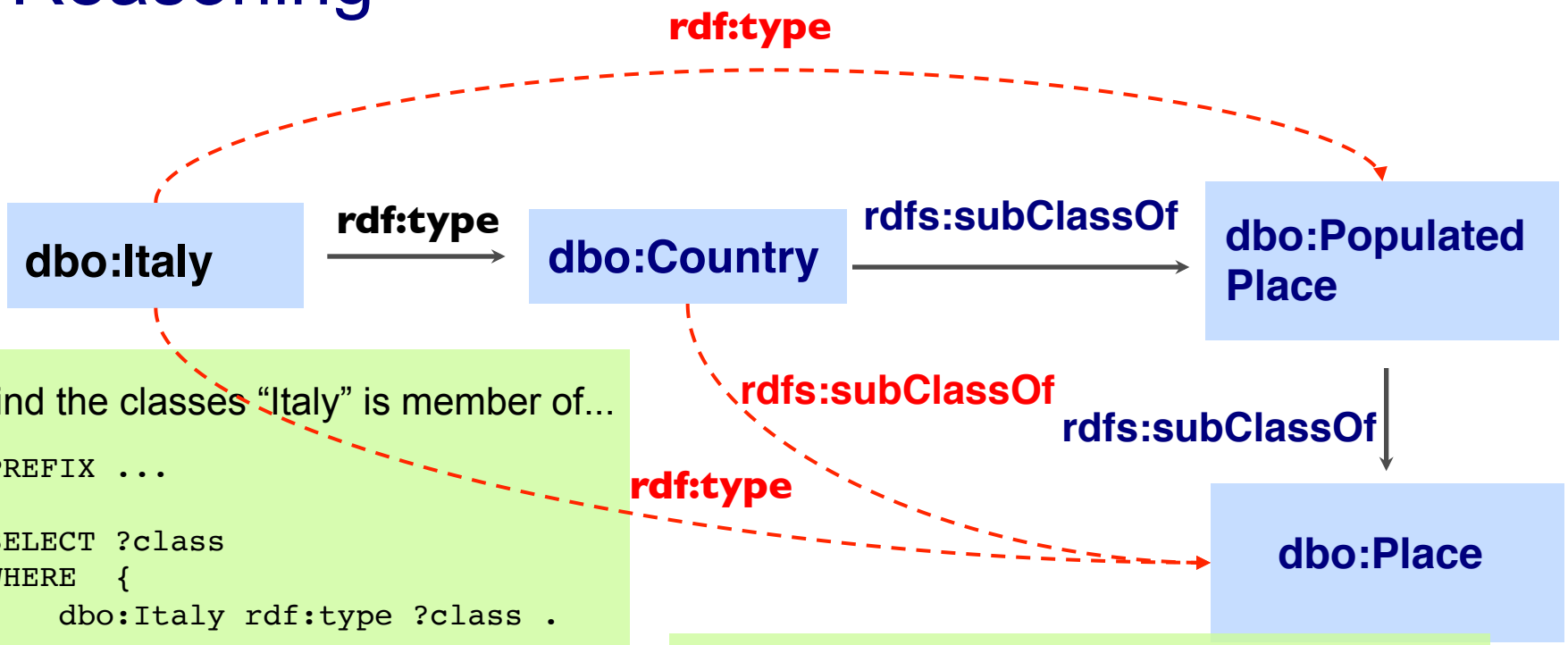


- *Subclass of* relation is **transitive**, **antisymmetric** and **reflexive**
- An entity is subclass of every superclass of its declared class

<u>In addition… (identity and equivalence)</u>

- **If** x = y **then** everything that is true for x is also true for y
- **If** C is equivalent to D **then** everything that is true for C is also true for D

(Pure) SPARQL Query Answering vs Reasoning



Find the classes "Italy" is member of...

```
PREFIX ...

SELECT ?class
WHERE  {
    dbo:Italy rdf:type ?class .
}
```

dbp:Country

dbo:Populated_Place? NO

dbp:Place?

Find the superclasses of "Country"

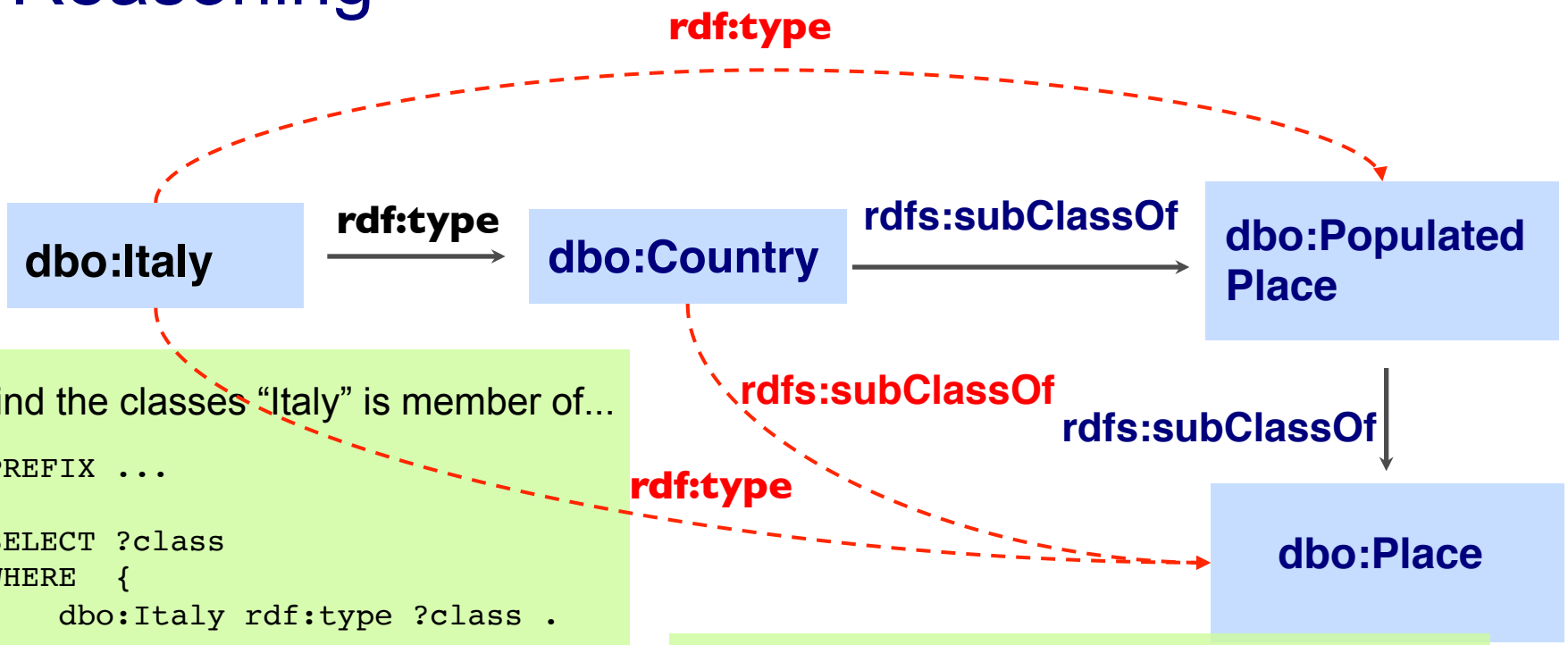
```
PREFIX ...

SELECT ?Class
WHERE  {
    dbo:Country rdfs:subClassOf ?Class .
}
```

dbo:Populated_Place

dbo:Place ? NO

(Pure) SPARQL Query Answering vs Reasoning



Find the classes "Italy" is member of...

```
PREFIX ...

SELECT ?class
WHERE  {
    dbo:Italy rdf:type ?class .
}
```

dbp:Country

dbo:Populated_Place

dbp:Place

Find the superclasses of "Country"

```
PREFIX ...

SELECT ?Class
WHERE  {
    dbo:Country rdfs:subClassOf ?Class .
}
```

dbo:Populated_Place

dbo:Place

SPARQL 1.1: Entailment Regimes

EUCLID
EdUcational Curriculum for the usage of LInked Data

- SPARQL 1.0 was defined only for simple entailment (pattern matching)

- SPARQL 1.1 is extended with entailment regimes other than simple entailment:
  - RDF entailment
  - **RDFS entailment**
  - D-Entailment
  - OWL RL entailment
  - OWL Full entailment
  - OWL 2 DL, EL, and QL entailment
  - RIF entailment

Source: http://www.w3.org/TR/rdf-mt/#RDFSRules

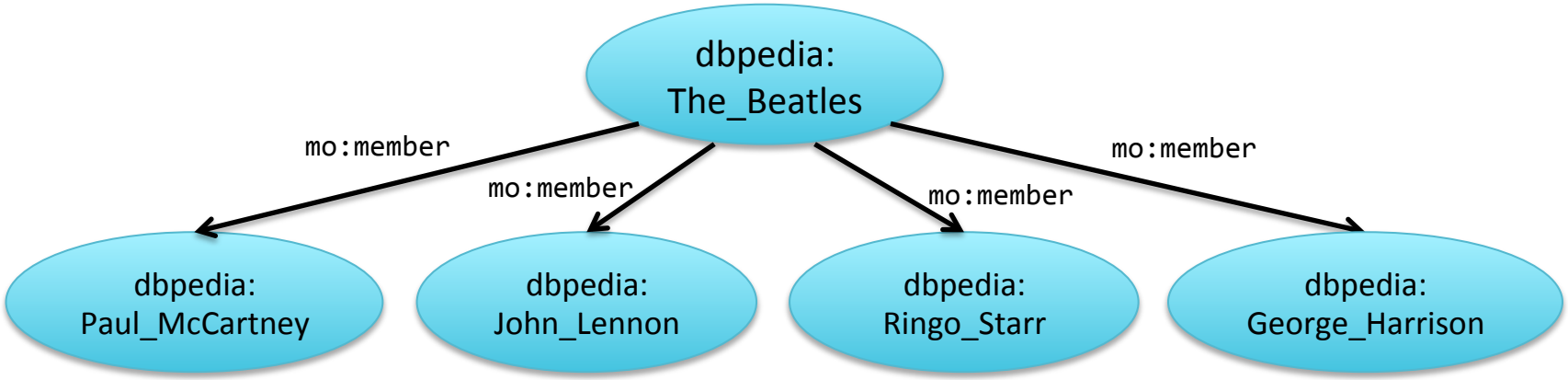
RDFS Entailment Regimes

EUCLID
EdUcational Curriculum for the usage of LInked Data

- Contains 13 entailment rules denominated rdfs*i* for inference over RDFS definitions*:
  - rdfs:Literal (rdfs1, rdfs13)
  - rdfs:domain (rdfs2), rdfs:range (rdfs3)
  - rdfs:Resource (rdfs4a, rdfs4, rdfs8)
  - rdfs:subPropertyOf (rdfs5, rdfs6, rdfs7, rdfs12)
  - rdfs:Class (rdfs8, rdfs10)
  - rdfs:subClassOf (rdfs9, rdfs10, rdfs11)
  - rdfs:ContainerMembershipProperty (rdfs12)
  - rdfs:Datatype (rdfs13)

* Source: http://www.w3.org/TR/rdf-mt/#RDFSRules

rdfs2 – rdfs:domain



Schema:

```
mo:member rdfs:domain
            mo:MusicGroup .
```

Query:

```
SELECT ?x WHERE {
 ?x a mo:MusicGroup.}
```

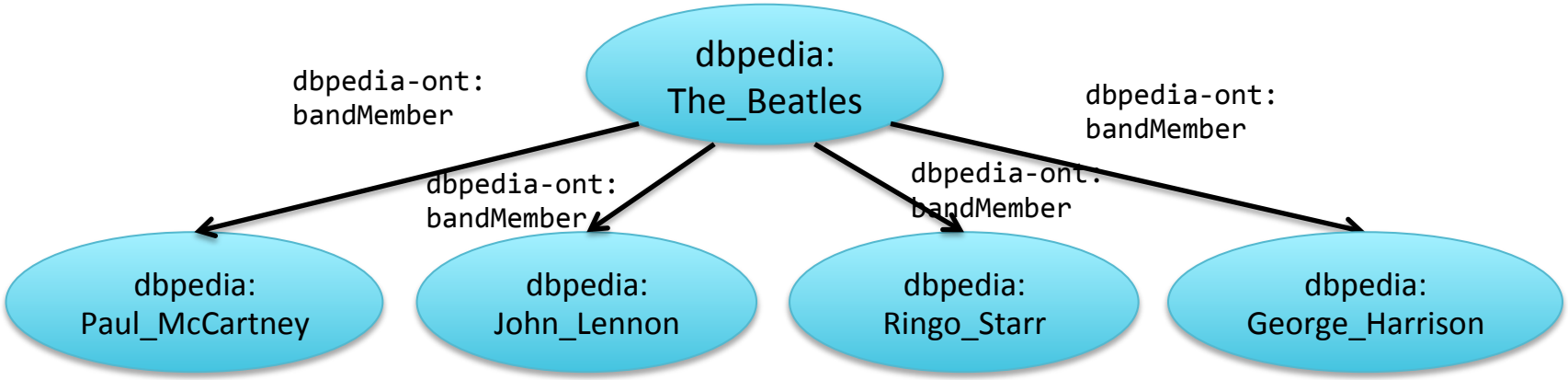
Result set:

?x

Result set with inference:

?x
dbpedia:The_Beatles …

rdfs3 – rdfs:range



Schema:

```
mo:member rdfs:range
          foaf:Agent .
```

Query:

```
SELECT ?x WHERE {
 ?x a foaf:Agent.}
```

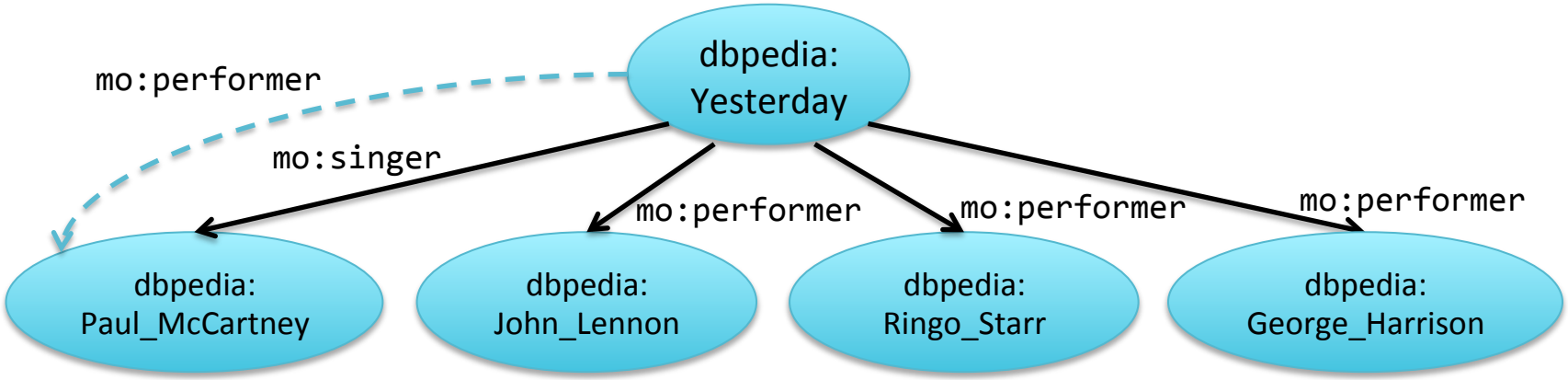
Result set:

?x

Result set with inference:

?x
dbpedia:Paul_McCartney
dbpedia:John_Lennon
dbpedia:Ringo_Starr
dbpedia:George_Harrison …

rdfs7 – rdfs:subPropertyOf



Schema:

```
mo:singer rdfs:subPropertyOf
mo:performer .
```

Query:

```
SELECT ?x WHERE {
 dbpedia:Yesterday mo:performer ?x.}
```

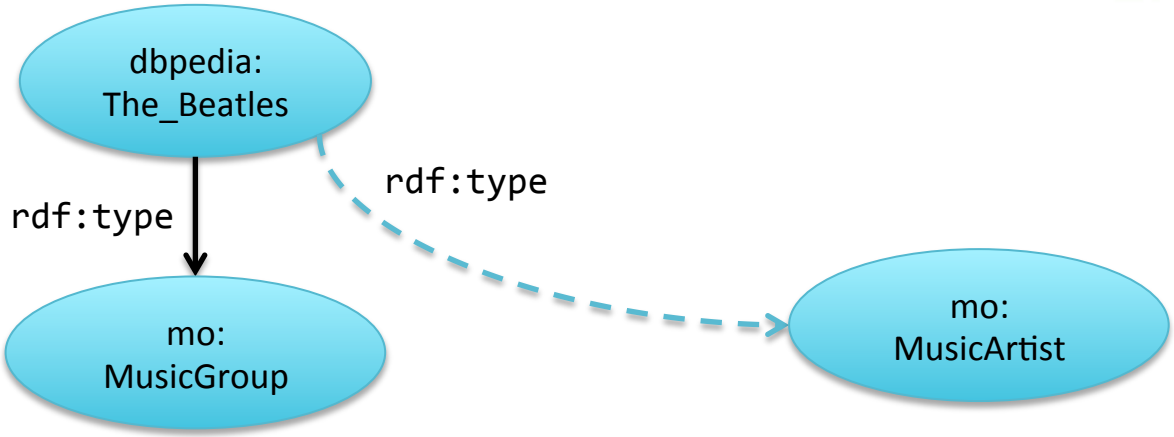
Result set:

?x
dbpedia:John_Lennon
dbpedia:Ringo_Starr
dbpedia:George_Harrison

Result set with inference:

?x
dbpedia:John_Lennon
dbpedia:Ringo_Starr
dbpedia:George_Harrison
dbpedia:Paul_McCartney

rdfs9 – rdfs:subClassOf



Schema:

```
mo:MusicGroup rdfs:subClassOf
mo:MusicArtist .
```

Query:

```
SELECT ?x WHERE {
 ?x a mo:MusicArtist.}
```

Result set:

?x

Result set with inference:

?x
dbpedia:The_Beatles …

Inference from Schema

EUCLID
EdUcational Curriculum for the usage of LInked Data

- Knowledge encoded in the schema leads to infer new facts

Schema:

```
mo:MusicGroup rdfs:subClassOf mo:MusicArtist .
```

Inferred facts:

```
mo:MusicGroup  a rdfs:Class .
mo:MusicArtist a rdfs:Class .
```

- This is also captured in the set of **axiomatic triples,** which provide basic meaning for all the vocabulary terms

```
rdfs:subClassOf rdfs:domain rdfs:Class .
rdfs:subClassOf rdfs:range  rdfs:Class .
```

RDFS:
Lack of Consistency Check

EUCLID
EdUcational Curriculum for the usage of LInked Data

- It is possible to infer facts that seem incorrect facts, but RDFS cannot prevent this:

Schema:

```
mo:member rdfs:domain mo:MusicGroup ;
          rdfs:range  foaf:Agent .
```

Existing facts:

```
:PaulMcCartney a :SoloMusicArtist ;
               :member :TheBeatles .
```

Inferred facts:

```
:PaulMcCartney a :MusicGroup .
```

No contradiction!:
The mis-modeling is not diagnosed

RDFS: Inference Limitations

EUCLID
EdUcational Curriculum for the usage of LInked Data

- We might wish further inferences, but these are **beyond** the entailment rules implemented by **RDFS**

Schema:
```
foaf:knows rdfs:domain foaf:Person ;
           rdfs:range  foaf:Person .
foaf:made  rdfs:domain foaf:Agent .
```

Existing fact:
```
:PaulMcCartney foaf:made :Yesterday ;
               foaf:knows :RingoStarr .
```

Inferred facts:
```
:PaulMcCartney a foaf:Agent ;
               a foaf:Person .
:RingoStarr a foaf:Person .
```

NOT inferred:
```
:Yesterday dc:creator :PaulMcCartney.
:RingoStarr foaf:knows :PaulMcCartney .
```

Cannot model with RDFS that if 'x makes y' implies that 'the creator of y is x'

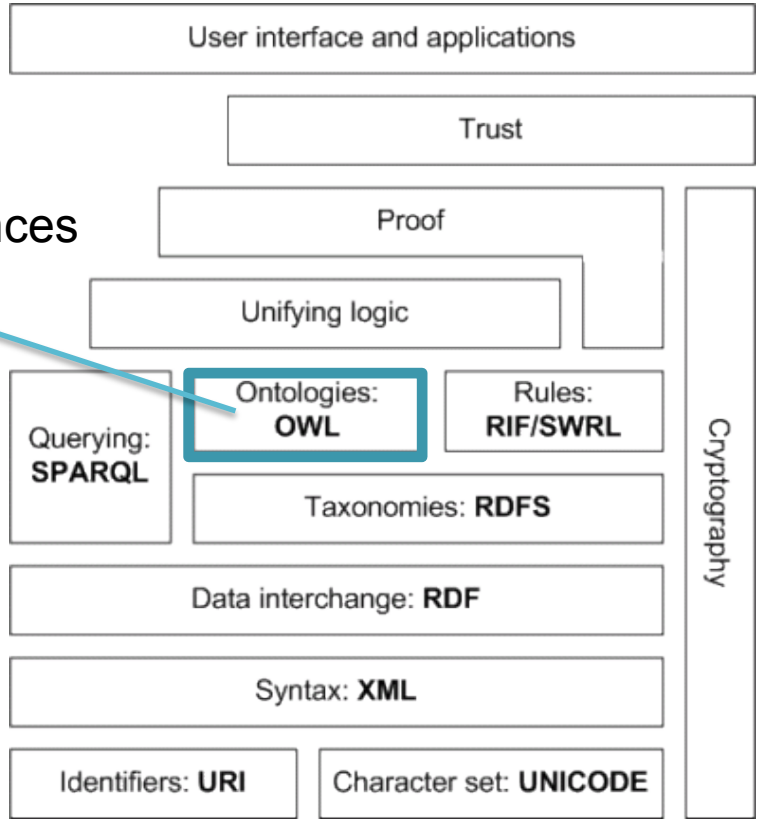
Cannot model with RDFS that 'x knows y' implies 'y knows x'

These inferences require **OWL!**

OWL

Web Ontology Language

Ontologies and inferences



Semantic Web Stack
Berners-Lee (2006)

Introduction to OWL

EUCLID
EdUcational Curriculum for the usage of LInked Data

- Provides more ontological constructs and avoids some of the potential confusion in RDFS
- OWL 2 is divided into sub-languages denominated *profiles*:
  - OWL 2 EL: Limited to basic classification, but with polynomial-time reasoning
  - OWL 2 QL: Designed to be translatable to relational database querying
  - OWL 2 RL: Designed to be efficiently implementable in rule-based systems

More restrictive than OWL DL

- Most triple stores concentrate on the use of RDFS with a subset of OWL features, called OWL-Horst or RDFS++