RDF

F. Toumani

Institut d'Informatique, LIMOS, UCA

February 6, 2018

Agenda

- Introduction
- RDF data model
- RDF Vocabularies
- Semantics of RDF graphs
- RDF DATA

- A framework for expressing information about resources (documents, people, physical objects, and abstract concepts)
- Enable exchanging information about resources **between applications without loss of meaning**
- The information may be made available to applications other than those for which it was originally created

- A framework for expressing information about resources (documents, people, physical objects, and abstract concepts)
- Enable exchanging information about resources **between applications without loss of meaning**
- The information may be made available to applications other than those for which it was originally created

Common (standard) framework $\Rightarrow$ availability of common RDF parsers and processing tools

- A framework for expressing information about resources (documents, people, physical objects, and abstract concepts)
- Enable exchanging information about resources **between applications without loss of meaning**
- The information may be made available to applications other than those for which it was originally created

Common (standard) framework $\Rightarrow$ availability of common RDF parsers and processing tools
Intended to be used to **publish** and **interlink** data on the Web

Bob is born on the 4th of July 1990. He is friend of Alice and he is interested in Mona Lisa.

Bob is born on the 4th of July 1990. He is friend of Alice and he is interested in Mona Lisa.

```
<person>
    <name> Bob</name>
    <birth-day> 4th of July 1990</birth-day>
    <friend> Alice</friend>
    <intersted-in> Mona Lisa</friend>
</person>
```

Bob is born on the 4th of July 1990. He is friend of Alice and he is interested in Mona Lisa.

```
<person>
    <name> Bob</name>
    <birth-day> 4th of July 1990</birth-day>
    <friend> Alice</friend>
    <intersted-in> Mona Lisa</friend>
</person>
<person name="Bob", birth-day= "4th of July 1990">
    <friend name="Alice"/>
    <intersted-in> Mona Lisa</friend>
</person>
```

Bob is born on the 4th of July 1990. He is friend of Alice and he is interested in Mona Lisa.

```
<person>
    <name> Bob</name>
    <birth-day> 4th of July 1990</birth-day>
    <friend> Alice</friend>
    <intersted-in> Mona Lisa</friend>
</person>
<person name="Bob", birth-day= "4th of July 1990">
    <friend name="Alice" />
    <intersted-in> Mona Lisa</friend>
</person>
<document>
  <person>
    <name birth − day = "4th of July 1990" > Bob</name>
    <friends>
        <name> Alice</friend>
    <friends>
    <intersted-in> Mona Lisa</friend>
  </person>
</document>
```

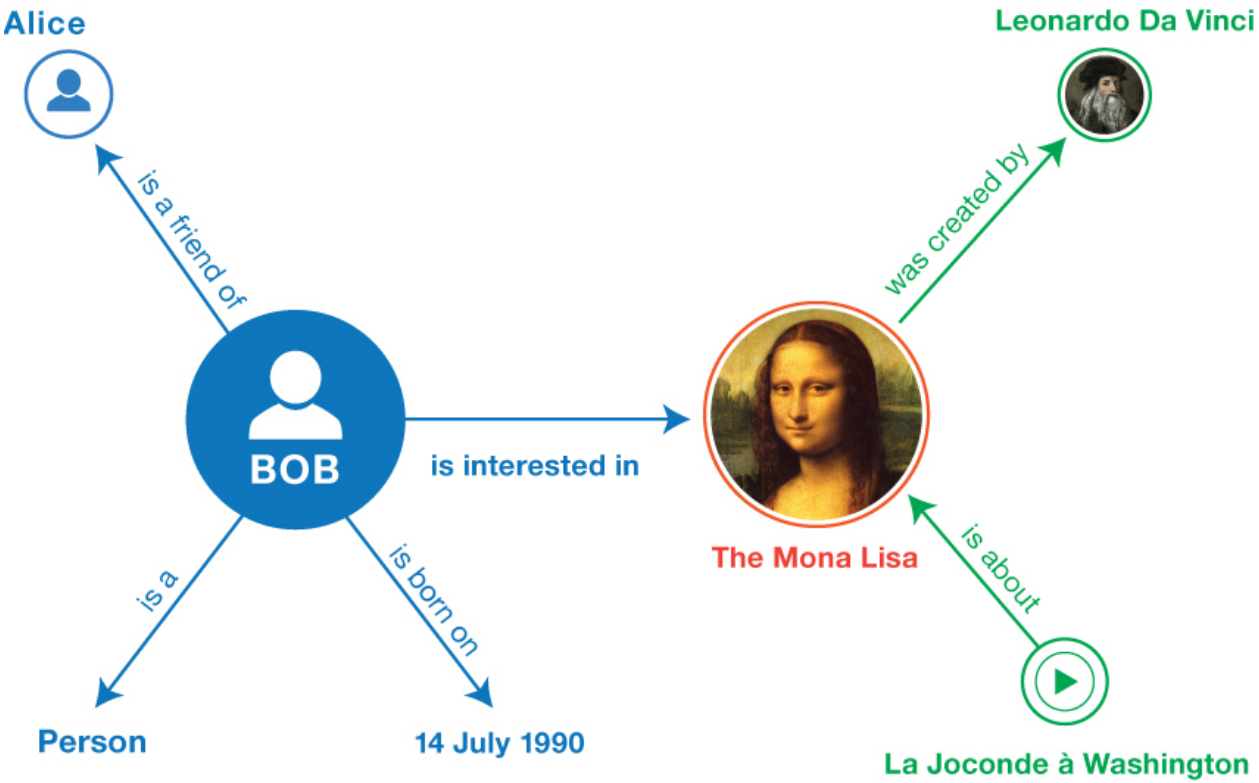
<Bob> <is a> <person>
<Bob> <is a friend of> <Alice>
<Bob> <is born on> <the 4th of July 1990>
<Bob> <is interested in> <the Mona Lisa>

<Bob> <is a> <person>
<Bob> <is a friend of> <Alice>
<Bob> <is born on> <the 4th of July 1990>
<Bob> <is interested in> <the Mona Lisa>

<the Mona Lisa> <was created by> <Leonardo da Vinci>
<the video La Joconde à Washington> <is about> <the Mona Lisa>

<Bob> <is a> <person>
<Bob> <is a friend of> <Alice>
<Bob> <is born on> <the 4th of July 1990>
<Bob> <is interested in> <the Mona Lisa>

<the Mona Lisa> <was created by> <Leonardo da Vinci>
<the video La Joconde à Washington> <is about> <the Mona Lisa>



RDF data model
Triples

RDF makes statements about resources as **triples**

<subject> <predicate> <object>

RDF data model
Triples

RDF makes statements about resources as **triples**

<subject> <predicate> <object>

<Bob> <is a> <person>

RDF data model
Triples

RDF makes statements about resources as **triples**

<subject> <predicate> <object>

<Bob> <is a> <person>

Ressource Relationship Ressource

RDF makes statements about resources as **triples**

<subject>	<predicate>	<object>
<Bob>	<is a>	<person>
Ressource	Relationship	Ressource

- An RDF statement expresses a **directed** relationship between two resources
- Ability to connect triples: the same resource could be in the subject position of one triple and the object position of another

RDF makes statements about resources as **triples**

<subject> <predicate> <object>

<Bob> <is a> <person>

Ressource Relationship Ressource

- An RDF statement expresses a **directed** relationship between two resources
- Ability to connect triples: the same resource could be in the subject position of one triple and the object position of another

<Bob> <is a friend of> <Alice>
<Alice> <is a> <person>
<Bob> <is interested in> <the Mona Lisa>
<the Mona Lisa> <was created by> <Leonardo da Vinci>

<the video La Joconde à Washington> <is about> <the Mona Lisa>

A global identifier that denotes a resource

A global identifier that denotes a resource
Example: **http://dbpedia.org/page/Tim_Berners-Lee**
(IRI for Tim Berners-Lee in DBpedia)

A global identifier that denotes a resource
Example: **http://dbpedia.org/page/Tim_Berners-Lee**
(IRI for Tim Berners-Lee in DBpedia)

- Identifies a resources, e.g., an URL
- Generalization of URI (Uniform Resource Identifier)
- An IRI can appear in all three positions of a triple
- RDF is agnostic about what the IRI represents
- IRIs may be given meaning by particular vocabularies or conventions
  Example: DBpedia IRI have the form
  http://dbpedia.org/resource/Name

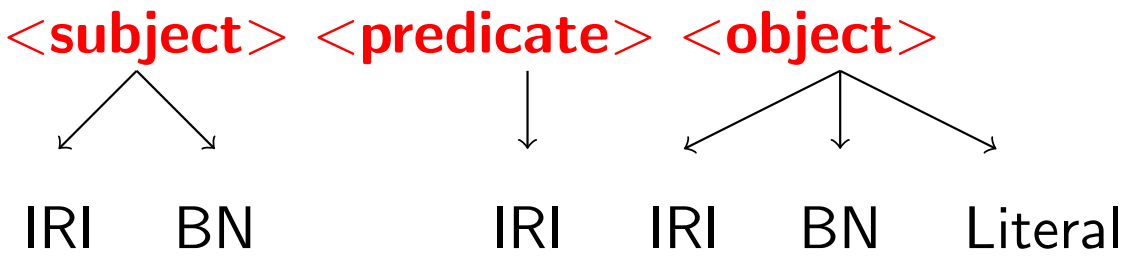
- Literals
  - Basic values that are not IRIs (e.g., strings, dates, . . .)
  - Literals are associated with a datatype enabling such values to be parsed and interpreted correctly
  - Literals may only appear in the object position of a triple
- Blank nodes
  - A resource without a global identifier
  - A blank node acts as a local identifier
  - Can always be replaced by a new, globally unique IRI (a Skolem IRI)
  - Blank nodes can appear in the subject and object position of a triple

An RDF term is either an IRI, a blank node or a literal

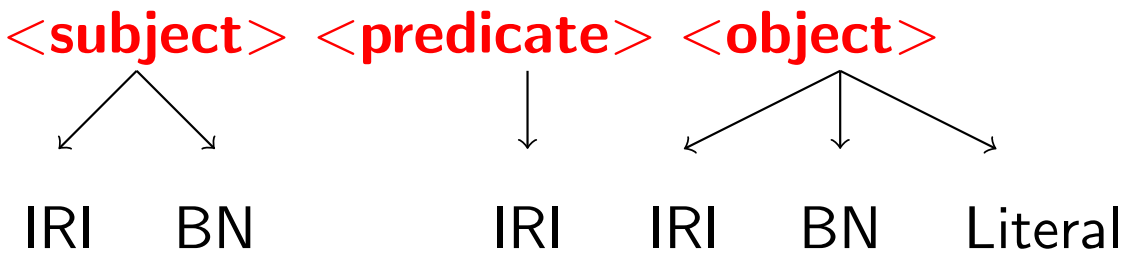
RDF triples and datasets

<subject> <predicate> <object>

RDF triples and datasets

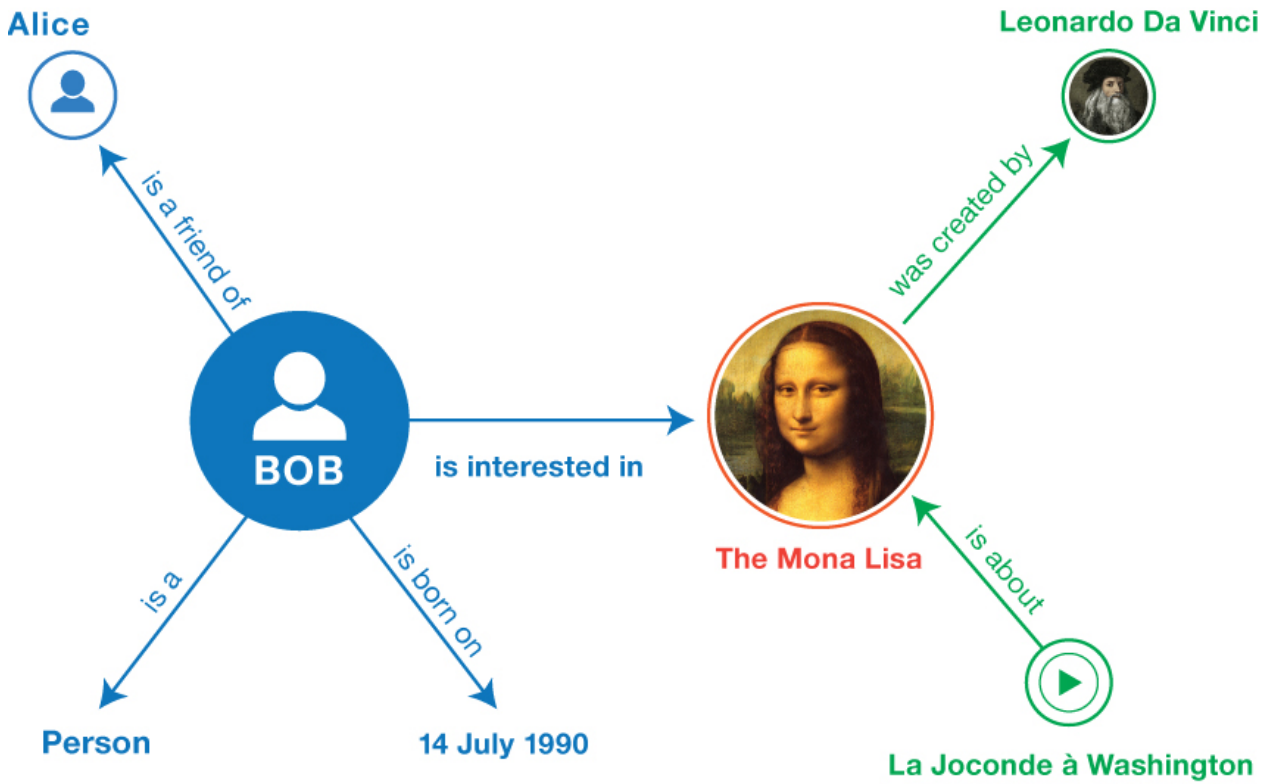


RDF triples and datasets

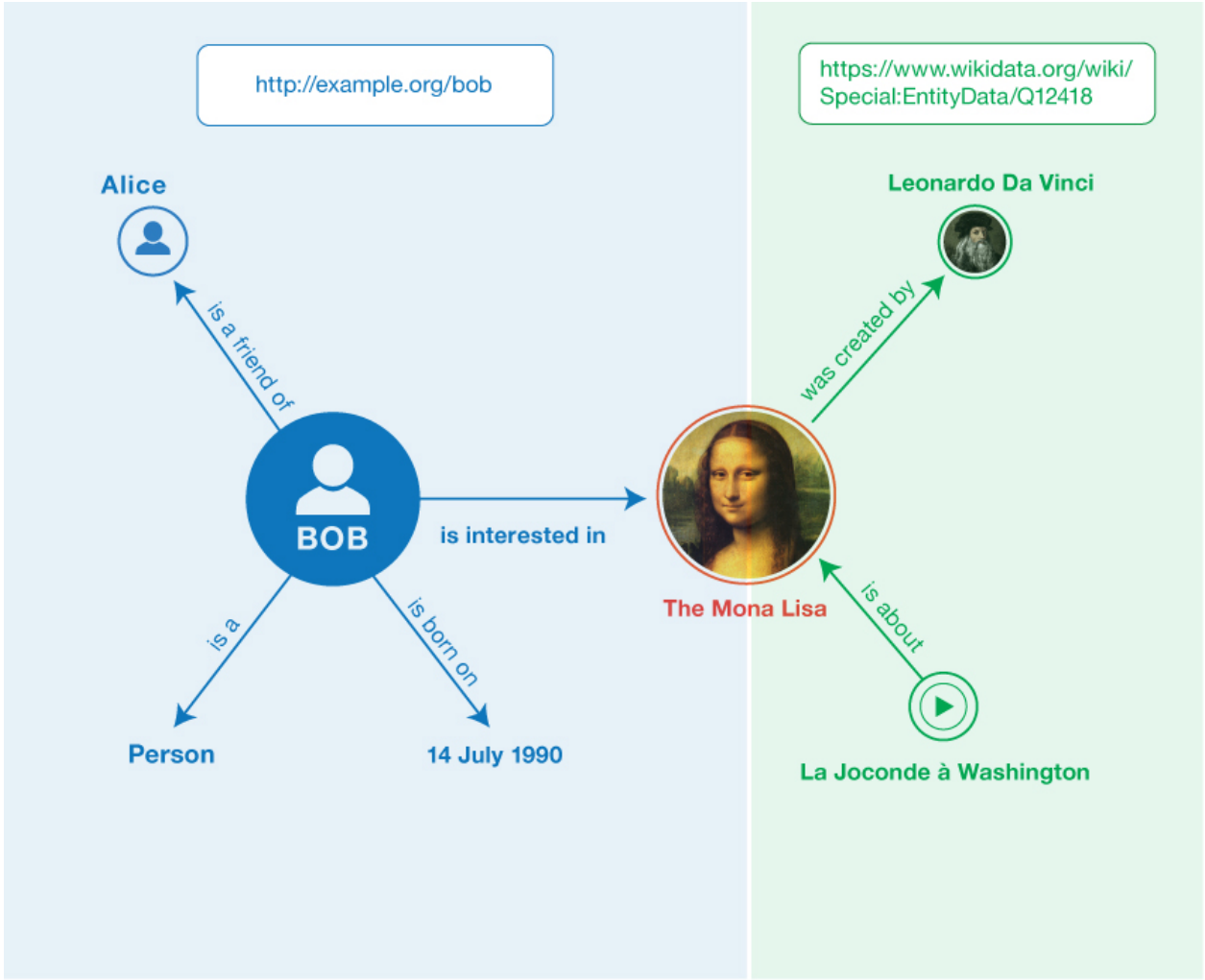


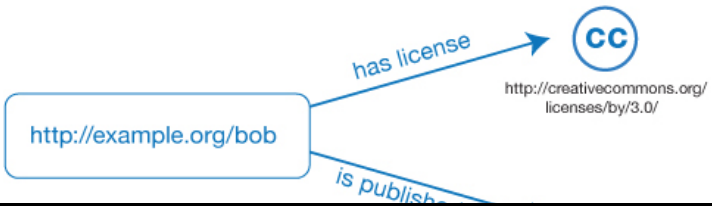
- An RDF Dataset: a set of RDF triples
- RDF Dataset $T \equiv$ an RDF graph G_T
  - the set of nodes of G_T is the set of RDF terms that occur as subject or object of the triples in T
  - there is an edge (s, o) in G_T labeled with p iff the triple (s, p, o) occurs in T

Example of an RDF graph



Multigraphs
Example





Literals and datatypes

- typed literals can be expressed using XML schema
  Example :
- language tags

Blank nodes
– figure of a graph with a blank node
a coma

Serialization of RDF graphs

Different **equivalent** serialization formats

- Turtle family of RDF languages
  - N-Triples
  - Turtle
  - TriG
  - N-Quads
- JSON-LD (JSON-based RDF syntax);
- RDFa (for HTML and XML embedding);
- RDF/XML (XML syntax for RDF).

Turtle familly of RDF languages
N-Triplets

- A line-based, plain-text serialization of RDF graphs
- Each line represents a triple
- Full IRIs are enclosed in angle brackets (<>)
- The period at the end of the line signals the end of the triple
- The datatype is appended to the literal through a ^^ delimiter
- The date representation follows the conventions of the XML Schema datatype date

Turtle familly of RDF languages

Example of N-Triplets

<http://ex.org/bob#me> <http://22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .

Turtle familly of RDF languages

Example of N-Triplets

<http://ex.org/bob#me> <http://22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://..#me> <http://../birthDate> "1990-07-04"^^ <http://www.w3.org/2001/XMLSchema#date> .

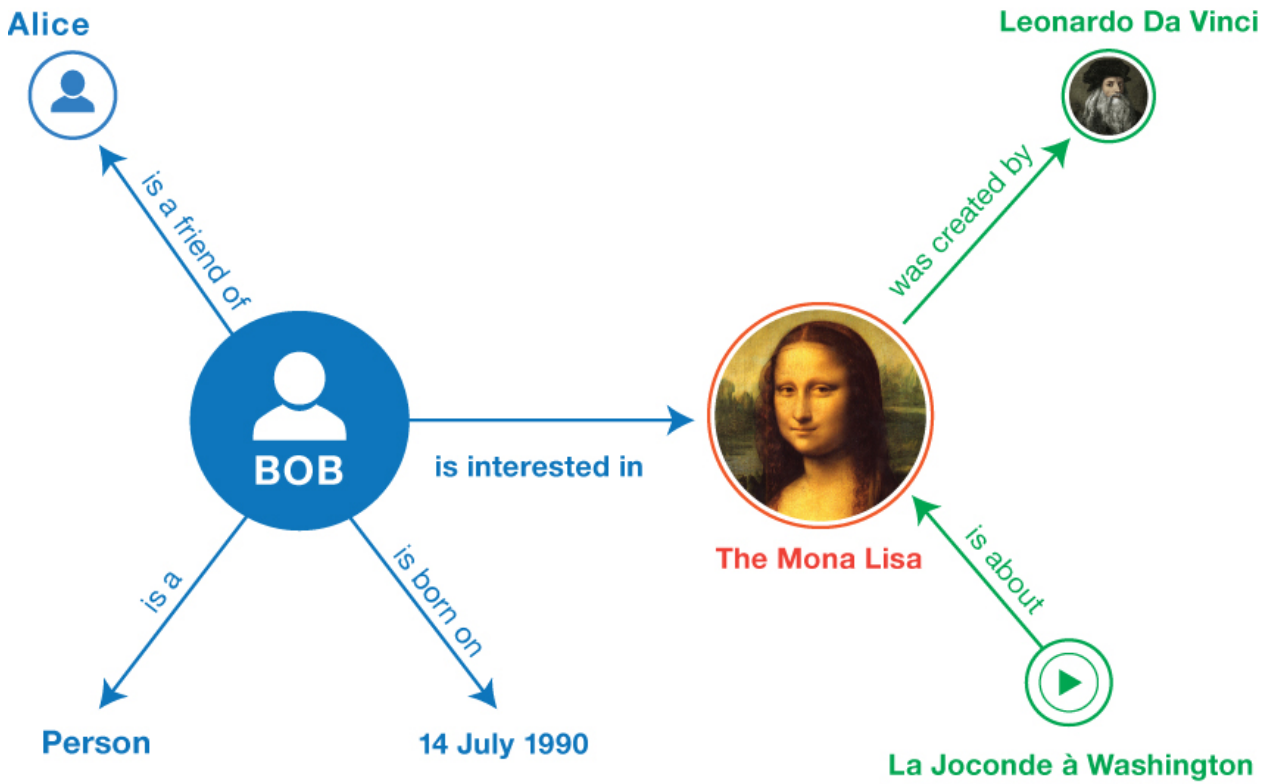
Turtle familly of RDF languages

Example of N-Triplets

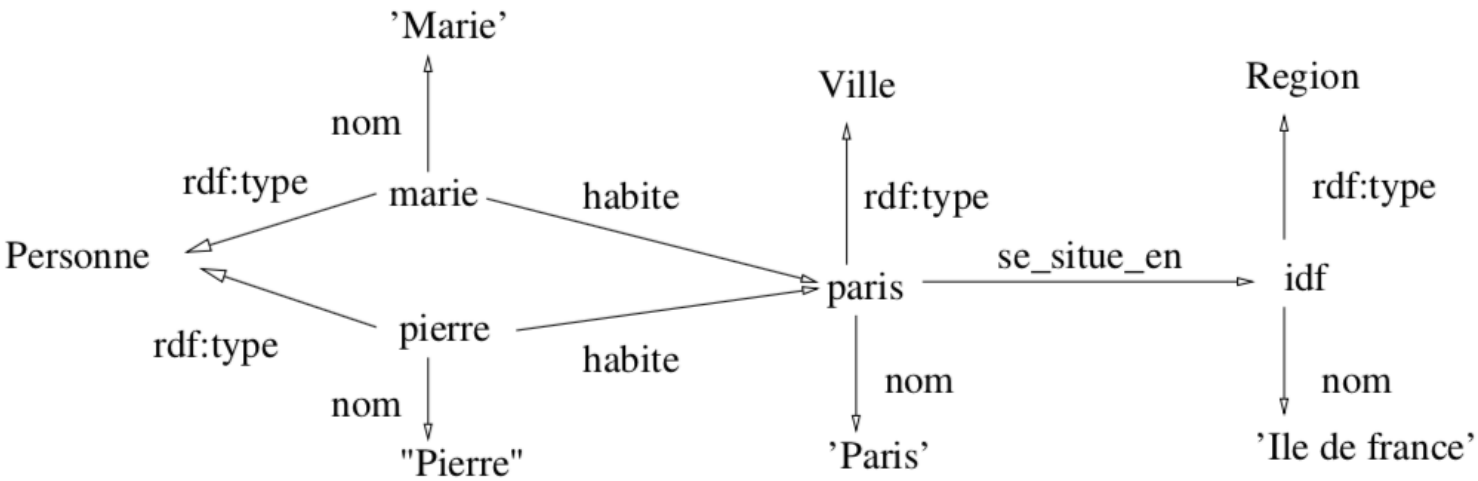
<http://ex.org/bob#me> <http://22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://..#me> <http://../birthDate> "1990-07-04"^^ <http://www.w3.org/2001/XMLSchema#date> .
<http://example.org/bob# me> <http://xmlns.com/foaf/0.1/topic_interest>
<http://www.wikidata.org/entity/Q12418> .
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/title> "Mona Lisa" .
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/creator>
<http://dbpedia.org/resource/Leonardo_da_Vinci> .
<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619>
<http://purl.org/dc/terms/subject> <http://www.wikidata.org/entity/Q12418> .

Turtle familly of RDF languages

Example of an RDF graph



Exercice


'Marie'
nom
rdf:type
marie
habite
Ville
rdf:type
Region
rdf:type
Personne
paris
se_situe_en
idf
rdf:type
pierre
habite
nom
nom
nom
"Pierre"
'Paris'
'Ile de france'

Turtle is an extension of N-Triples

- basic N-Triples syntax augmented with a number of syntactic shortcuts
- support for namespace prefixes
- lists and shorthands for datatyped literals
- blank nodes in Turtle are expressed as `_:` followed by a blank node label

Turtle provides a trade-off between ease of writing, ease of parsing and readability

<http://www.wikidata.org/entity/Q12418>
<http://purl.org/dc/terms/creator>
<http://dbpedia.org/resource/Leonardo_da_Vinci> .

<http://www.wikidata.org/entity/Q12418>
<http://purl.org/dc/terms/creator>
<http://dbpedia.org/resource/Leonardo_da_Vinci> .

@prefix wikid: <http://www.wikidata.org/entity/Q12418> .
@prefix purl <http://purl.org/dc/terms/i> .
@base <http://dbpedia.org/resource> .

<http://www.wikidata.org/entity/Q12418>
<http://purl.org/dc/terms/creator>
<http://dbpedia.org/resource/Leonardo_da_Vinci> .

@prefix wikid: <http://www.wikidata.org/entity/Q12418> .
@prefix purl <http://purl.org/dc/terms/i> .
@base <http://dbpedia.org/resource> .

wikid:Q12418 purl:creator Leonardo_da_Vinci .

Turtle familly of RDF languages

Turtle predicate list

```
@base <http://example.org/>
@prefix foaf: <http://xmlns.com/foaf/0.1/>
@prefix schema: <http://schema.org/>
@prefix wd: <http://www.wikidata.org/entity/>
<bob#me> foaf:knows <alice#me>
<bob#me> schema:birthDate "1990-07-04"^^xsd:date .
<bob#me> foaf:topic_interest wd:Q12418 .
```

Turtle familly of RDF languages

Turtle predicate list

```
@base <http://example.org/>
@prefix foaf: <http://xmlns.com/foaf/0.1/>
@prefix schema: <http://schema.org/>
@prefix wd: <http://www.wikidata.org/entity/>
<bob#me> foaf:knows <alice#me>
<bob#me> schema:birthDate "1990-07-04"^^xsd:date .
<bob#me> foaf:topic_interest wd:Q12418 .

<bob#me>
    foaf:knows <alice#me> ;
    schema:birthDate "1990-07-04"^^xsd:date ;

    foaf:topic_interest wd:Q12418 .
```

Turtle familly of RDF languages

Turtle object list and blank nodes

```
@base <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix schema: <http://schema.org/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix wd: <http://www.wikidata.org/entity/> .
```

Turtle familly of RDF languages

Turtle object list and blank nodes

```
@base <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix schema: <http://schema.org/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix wd: <http://www.wikidata.org/entity/> .
<bob#me> foaf:knows <alice#me> .
<bob#me> foaf:knows <marie#me> .
<bob#me> foaf:knows <toto#me> .
<bob#me> foaf:knows _:someone .
```

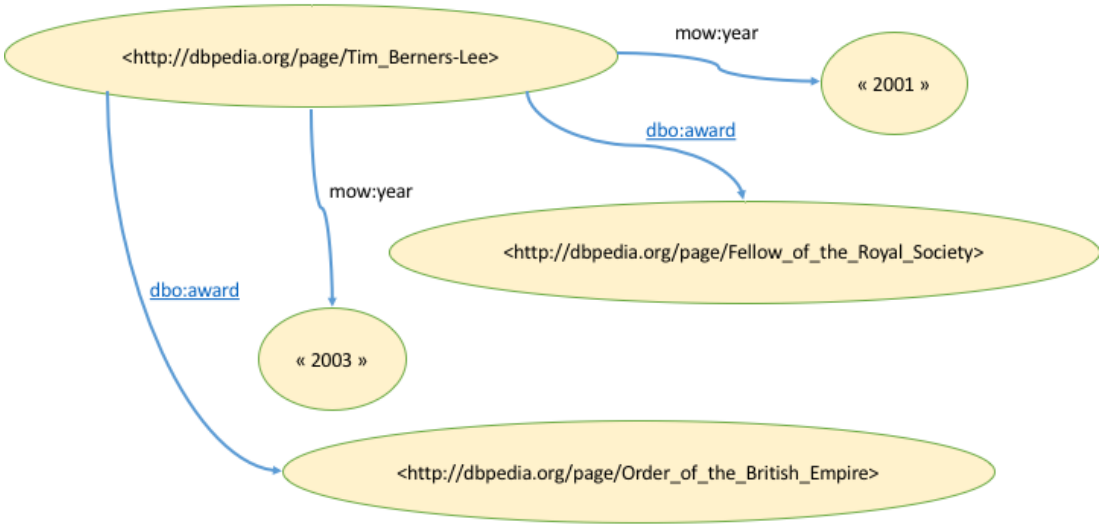
Turtle familly of RDF languages

Turtle object list and blank nodes

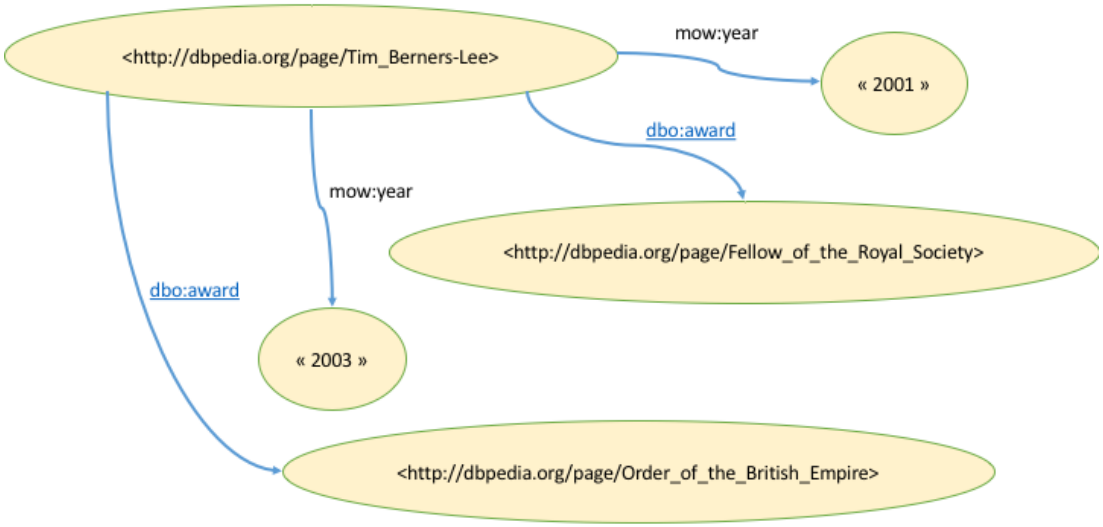
```
@base <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix schema: <http://schema.org/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix wd: <http://www.wikidata.org/entity/> .
<bob#me> foaf:knows <alice#me> .
<bob#me> foaf:knows <marie#me> .
<bob#me> foaf:knows <toto#me> .
<bob#me> foaf:knows _:someone .
```

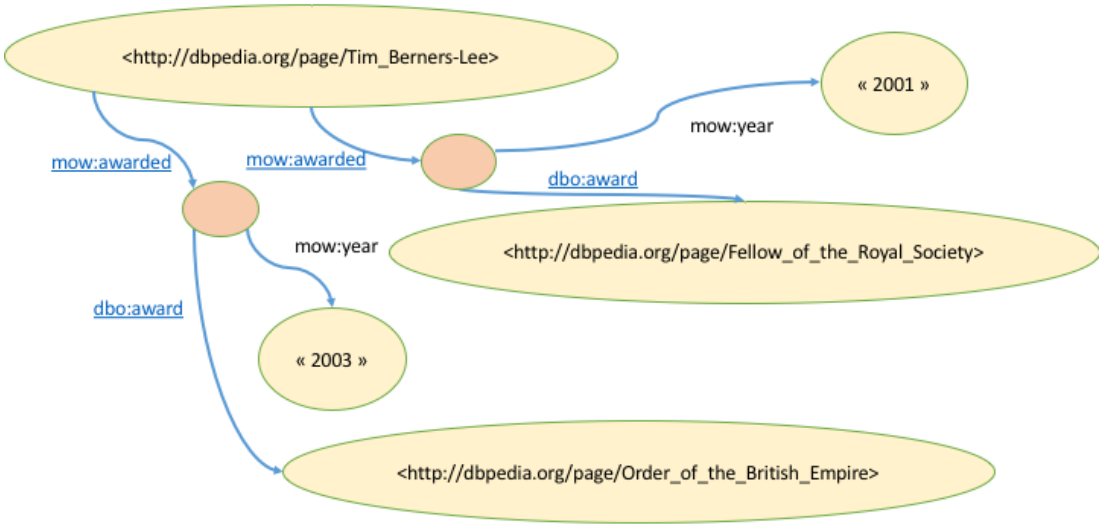
```
<bob#me>    foaf:knows        <alice#me> ,
                              <marie#me> ,
                              <toto#me> ,
                              [] .
```

Turtle familly of RDF languages
Multivalued relations



Turtle familly of RDF languages
Multivalued relations





Turtle familly of RDF languages

Multivalued relations

```
@base <http://example.org/> .
@prefix dbp: <¡http://dbpedia.org/page/> .
@prefix mow: <http://www.mytest.fr#> .
[]      dbo:award dbo:Fellow_of_the_Royal_Society ;
           mow:year "2001" .
[]      dbo:award dbo:Order_of_the_British_Empiry ;
           mow:year "2003" .
```

```
@base <http://example.org/> .
@prefix dbp: <¡http://dbpedia.org/page/> .
@prefix mow: <http://www.mytest.fr#> .
[]      dbo:award dbo:Fellow_of_the_Royal_Society ;
           mow:year "2001" .
[]      dbo:award dbo:Order_of_the_British_Empiry ;
           mow:year "2003" .
```

Nested anonymous blank

```
dbp:Tim_Berners-Lee    mow:awarded
                       [ dbo:award dbo:Fellow_of_the_Royal_Society ;
                            mow:year "2001" . ] ,
                       [ dbo:award dbo:Order_of_the_British_Empiry ;
                            mow:year "2003" . ] .
```

- RDF Lists
  - Container (open lists)
  - Collections (closed lists)

Turtle familly of RDF languages
Aggregation of RDF data

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.org/test#> .
ex:SolarSystem ex:hasPlanets [
      a rdf:Seq ;
      rdf:_1 ex:Mercury ;
      rdf:_2 ex:Venus ;
      rdf:_3 ex:Earth ;
      rdf:_4 ex:Mars ;
      rdf:_5 ex:Jupiter ;
      rdf:_6 ex:Saturn
].
```

Turtle familly of RDF languages
Aggregation of RDF data

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.org/test#> .
ex:SolarSystem ex:hasPlanets [
      a rdf:Seq ;
      rdf:_1 ex:Mercury ;
      rdf:_2 ex:Venus ;
      rdf:_3 ex:Earth ;
      rdf:_4 ex:Mars ;
      rdf:_5 ex:Jupiter ;
      rdf:_6 ex:Saturn
].   @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.org/test#> .
ex:SolarSystem ex:hasPlanets [
rdf:first ex:Mercury ; rdf:rest [
rdf:first ex:Venus ; rdf:rest [
rdf:first ex:Earth ; rdf:rest [
rdf:first ex:Mars ; rdf:rest [
rdf:first ex:Jupiter ; rdf:rest [
rdf:first ex:Saturn ;
rdf:rest rdf:nil

] ] ] ] ] ].
```

Other RDF languages
RDF/XML

```
<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about="subject">
        <predicate rdf:resource="object" />
        <predicate>literal value</predicate>
    <rdf:Description>

</rdf:RDF>
```

RDF/XML

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
         xmlns:dcterms="http://purl.org/dc/terms/"
         xmlns:foaf="http://xmlns.com/foaf/0.1/"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:schema="http://schema.org/">
   <rdf:Description rdf:about="http://example.org/bob#me">
      <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
      <schema:birthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1990-07-04</schema:birthDate>
      <foaf:knows rdf:resource="http://example.org/alice#me"/>
      <foaf:topic_interest rdf:resource="http://www.wikidata.org/entity/Q12418"/>
   </rdf:Description>
   <rdf:Description rdf:about="http://www.wikidata.org/entity/Q12418">
      <dcterms:title>Mona Lisa</dcterms:title>
      <dcterms:creator rdf:resource="http://dbpedia.org/resource/Leonardo_da_Vinci"/>
   </rdf:Description>
   <rdf:Description rdf:about="http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619">
      <dcterms:subject rdf:resource="http://www.wikidata.org/entity/Q12418"/>
   </rdf:Description>
</rdf:RDF>
```

RDF Schema

- The RDF data model provides a way to make statements about resources
- The RDF data model does not make any assumptions about what resource IRIs stand for
- In practice, RDF is typically used in combination with vocabularies or other conventions that provide semantic information about these resources
- RDF Schema allows one to define semantic characteristics of RDF data
  For example, one can state that the IRI http://www.example.org/friendOf can be used as a property and that the subjects and objects of http://www.example.org/friendOf triples must be resources of class http://www.example.org/Person.

- Class to specify categories that can be used to classify resources
- Property as a relation between an instance and its class
- Class and property hierarchies
- Domain and range constraints to specify type restrictions on the subjects and objects of particular triples

RDF Schema

- Class (a class)
- Property (a class)
- subClassOf (a property)
- subPropertyOf (a property)
- domain (a property)
- range (a property)

RDF Schema

- Class (a class)

  C rdf:type rdfs:Class
- Property (a class)

  P rdf:type rdf:Property
- subClassOf (a property)

  C1 rdfs:subClassOf C2
- subPropertyOf (a property)

  P1 rdfs:subPropertyOf P2
- domain (a property)

  P rdfs:domain C
- range (a property)

  P rdfs:range C

- "Friend of a Friend" (FOAF) vocabulary for describing social networks
- Dublin Core
  The Dublin Core Metadata Initiative maintains a metadata element set for describing a wide range of resources. The vocabulary provides properties such as "creator", "publisher" and "title".
- Schema.org
  Schema.org is a vocabulary developed by a group of major search providers. The idea is that webmasters can use these terms to mark-up Web pages, so that search engines understand what the pages are about.
- SKOS
  SKOS is a vocabulary for publishing classification schemes such as terminologies and thesauri on the Web

RDF Schema
Vocabularies: Example

Main FOAF terms, grouped in broad categories.

- **Core** - These classes and properties form the core of FOAF. They describe characteristics of people and social groups that are independent of time and technology; as such they can be used to describe basic information about people in present day, historical, cultural heritage and digital library contexts. In addition to various characteristics of people, FOAF defines classes for Project, Organization and Group as other kinds of agent. Related work: Dublin Core, SKOS, DOAP, SIOC, Org vocabulary, Bio vocabulary.
- **Social Web** - in addition to the FOAF core terms, there are a number of terms for use when describing Internet accounts, addressbooks and other Web-based activities. Related work: Portable Contacts, W3C Social Web group.
- **Linked Data utilities** - FOAF began as the 'RDFWeb' project, and established a widely adopted model for publishing simple factual data via a networked of linked RDF documents. FOAF remains important to the growing "Linked Data" community, while also maintaining a concern for information that is not readily summarised as simple factual data. FOAF is an attempt to use the Web to integrate factual information with information in human-oriented documents (eg. videos, books, spreadsheets, 3d models), as well as information that is still in people's heads. This history explains why FOAF includes a few 'demonstration' terms that served largely educational purposes (eg. geekcode), alongside a few technical utility terms (eg. focus, LabelProperty) that support wider information-linking efforts. The dictionary-based design of FOAF allows a certain pragmatism: we simply record here a set of terms that are useful to the Web community, while keeping an emphasis on the central idea of FOAF, which is about linking networks of information with networks of people.

FOAF Core

- Agent
- Person
- name
- title
- img
- depiction (depicts)
- familyName
- givenName
- knows
- based_near
- age
- made (maker)
- primaryTopic (primaryTopicOf)
- Project
- Organization
- Group
- member
- Document
- Image

Social Web

- nick
- mbox
- homepage
- weblog
- openid
- jabberID
- mbox_sha1sum
- interest
- topic_interest
- topic (page)
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- publications
- currentProject
- pastProject
- account
- OnlineAccount
- accountName
- accountServiceHomepage
- PersonalProfileDocument
- tipjar
- sha1
- thumbnail
- logo

Main FOAF terms, grouped in broad categories.

- **Core** - These classes and properties form the core of FOAF. They describe characteristics of people and social groups that are independent of time and technology; as such they can be used to describe basic information about people in present day, historical, cultural heritage and digital library contexts. In addition to various characteristics of people, FOAF defines classes for Project, Organization and Group as other kinds of agent. Related work: Dublin Core, SKOS, DOAP, SIOC, Org vocabulary, Bio vocabulary.
- **Social Web** - in addition to the FOAF core terms, there are a number of terms for use when describing Internet accounts, addressbooks and other Web-based activities. Related work: Portable Contacts, W3C Social Web group.
- **Linked Data utilities** - FOAF began as the 'RDFWeb' project, and established a widely adopted model for publishing simple factual data via a networked of linked RDF documents. FOAF remains important to the growing "Linked Data" community, while also maintaining a concern for information that is not readily summarised as simple factual data. FOAF is an attempt to use the Web to integrate factual information with information in human-oriented documents (eg. videos, books, spreadsheets, 3d models), as well as information that is still in people's heads. This history explains why FOAF includes a few 'demonstration' terms that served largely educational purposes (eg. geekcode), alongside a few technical utility terms (eg. focus, LabelProperty) that support wider information-linking efforts. The dictionary-based design of FOAF allows a certain pragmatism: we simply record here a set of terms that are useful to the Web community, while keeping an emphasis on the central idea of FOAF, which is about linking networks of information with networks of people.

FOAF Core

- Agent
- Person
- name
- title
- img
- depiction (depicts)
- familyName
- givenName
- knows
- based_near
- age
- made (maker)
- primaryTopic (primaryTopicOf)
- Project
- Organization
- Group
- member
- Document
- Image

Social Web

- nick
- mbox
- homepage
- weblog
- openid
- jabberID
- mbox_sha1sum
- interest
- topic_interest
- topic (page)
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- publications
- currentProject
- pastProject
- account
- OnlineAccount
- accountName
- accountServiceHomepage
- PersonalProfileDocument
- tipjar
- sha1
- thumbnail
- logo

Example

Here is a very basic document describing a person:

```
<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>
```

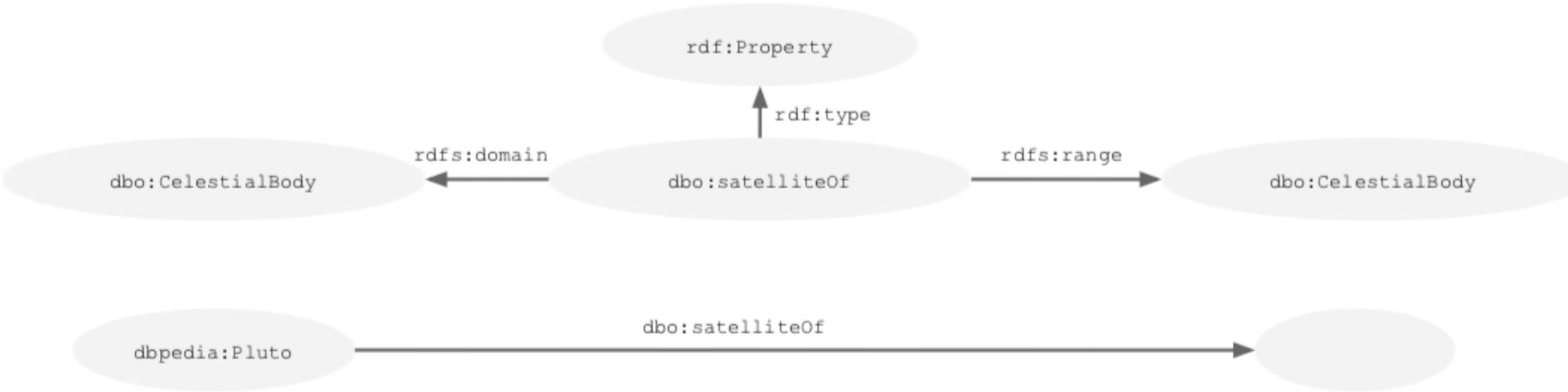
Semantics of RDF graphs

standard syntax + agreement about the meaning (semantics)
Intuitive semantics of RDF:

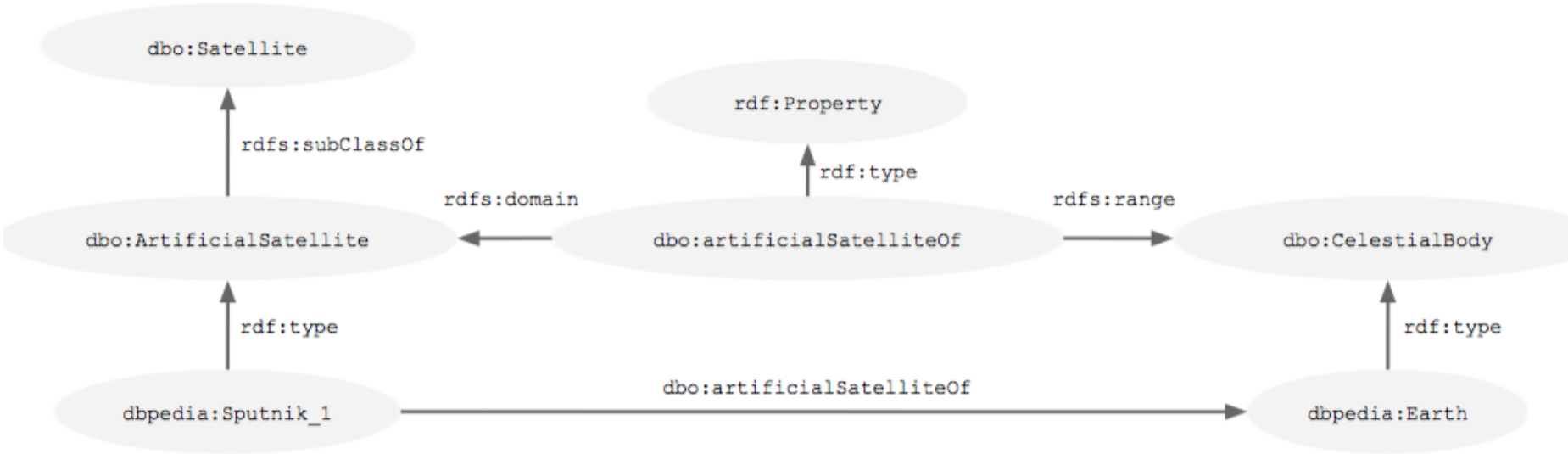
- The IRIs used to name the subject, predicate, and object are "global" in scope, naming the same thing each time they are used.
- Each triple is "true" exactly when the predicate relation actually exists between the subject and the object.
- An RDF graph is "true" exactly when all the triples in it are "true".

These notions, and others, are specified with mathematical precision in the RDF Semantics document [RDF11-MT].

Reasoning with RDF graphs



Reasoning with RDF graphs



Reasoning with RDF graphs