

Semantic Web: standards and languages for knowledge representation

N. Aussenac-Gilles

IRIT- CNRS

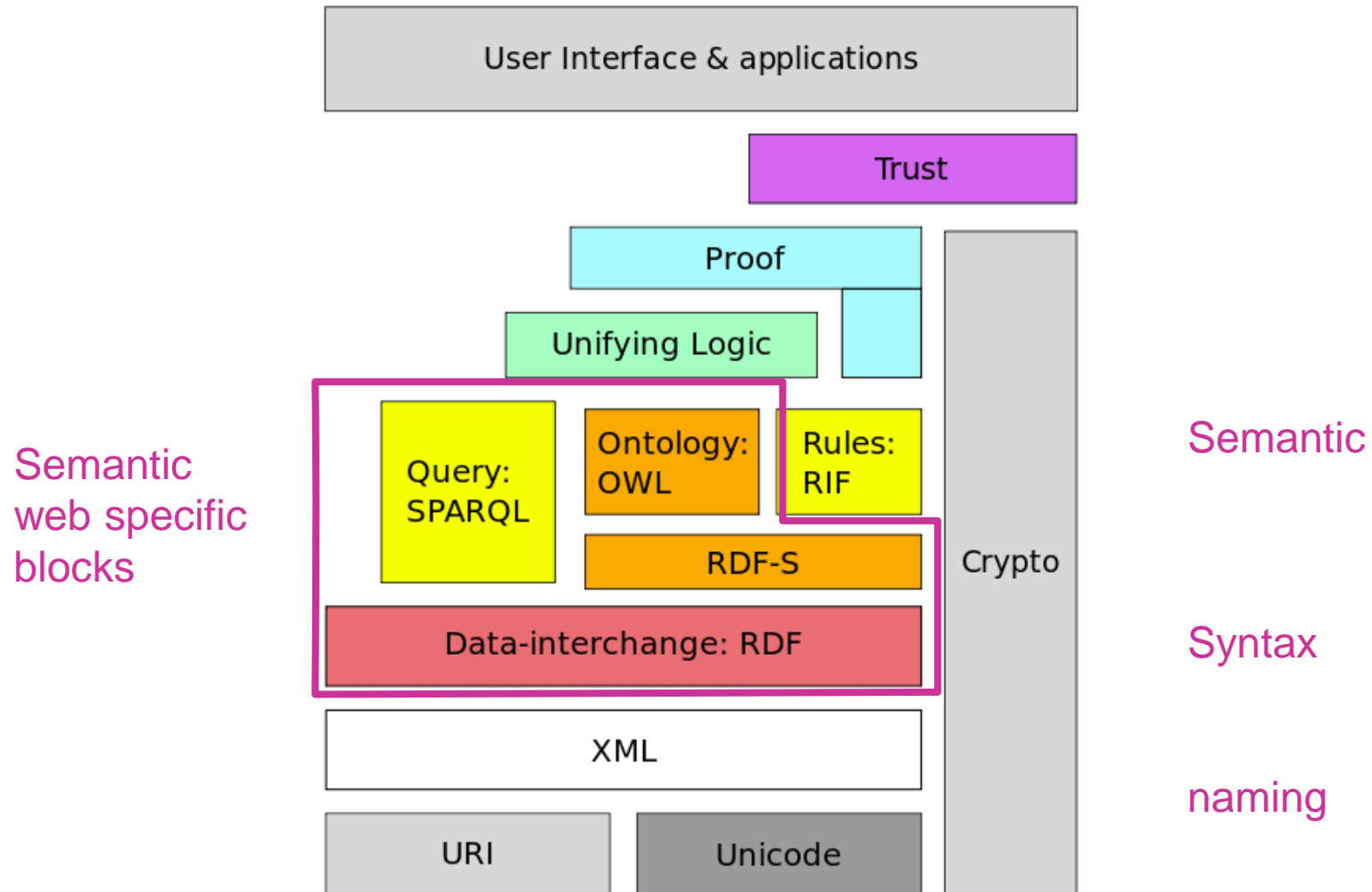
aussenac@irit.fr

MELODI group

<http://www.irit.fr/-Equipe-MELODI->



The Semantic Web layer cake (2006)



RDF vocabulary

■ Classes

rdf:Description – triple description

rdf:Resource - the class resource, everything

rdf:XMLLiteral - the class of XML literal values

rdf:Property - the class of properties

rdf:Statement - the class of RDF statements

rdf:Alt, rdf:Bag, rdf:Seq - containers of alternatives, unordered containers, and ordered containers

rdf:List - the class of RDF Lists

rdf:nil - an instance of rdf:List representing the empty list

■ Properties

rdf:type - an instance of rdf:Property used to state that a resource is an instance of a class

rdf:first - the first item in the subject RDF list

rdf:rest - the rest of the subject RDF list after the first item

rdf:value - idiomatic property used for structured values

rdf:subject - the subject of the subject RDF statement

rdf:predicate - the predicate of the subject RDF statement

rdf:object - the object of the subject RDF statement

RDF: exercise 1

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:region="http://www.country-regions.fake/">
```

```
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Oxford">
```

```
    <dc:title>Oxford</dc:title>
```

```
    <dc:coverage>Oxfordshire</dc:coverage>
```

```
    <dc:publisher>Wikipedia</dc:publisher>
```

```
    <region:population>10000</region:population>
```

```
    <region:principaltown rdf:resource="http://www.country-regions.fake/oxford"/>
```

```
  </rdf:Description>
```

```
</rdf:RDF>
```

- Quel est la notation de RDF utilisée ?
- Quels sont les espaces de noms référencés et à quoi correspondent-ils ?
- Quel est l'entité principalement décrite ?
- Combien y a-t-il de triplets ?
- Réécrire la même information en TURTLE
- Dessiner le graphe correspondant

exercise 1

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:region="http://www.country-regions.fake/">
```

```
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Oxford">
```

```
    <dc:title>Oxford</dc:title>
```

```
    <dc:coverage>Oxfordshire</dc:coverage>
```

```
    <dc:publisher>Wikipedia</dc:publisher>
```

```
    <region:population>10000</region:population>
```

```
    <region:principaltown rdf:resource="http://www.country-regions.fake/oxford"/>
```

```
  </rdf:Description>
```

```
</rdf:RDF>
```

Exercice 1 en TURTLE

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
@prefix region: <http://www.country-regions.fake/> .
```

```
<http://en.wikipedia.org/wiki/Oxford>
```

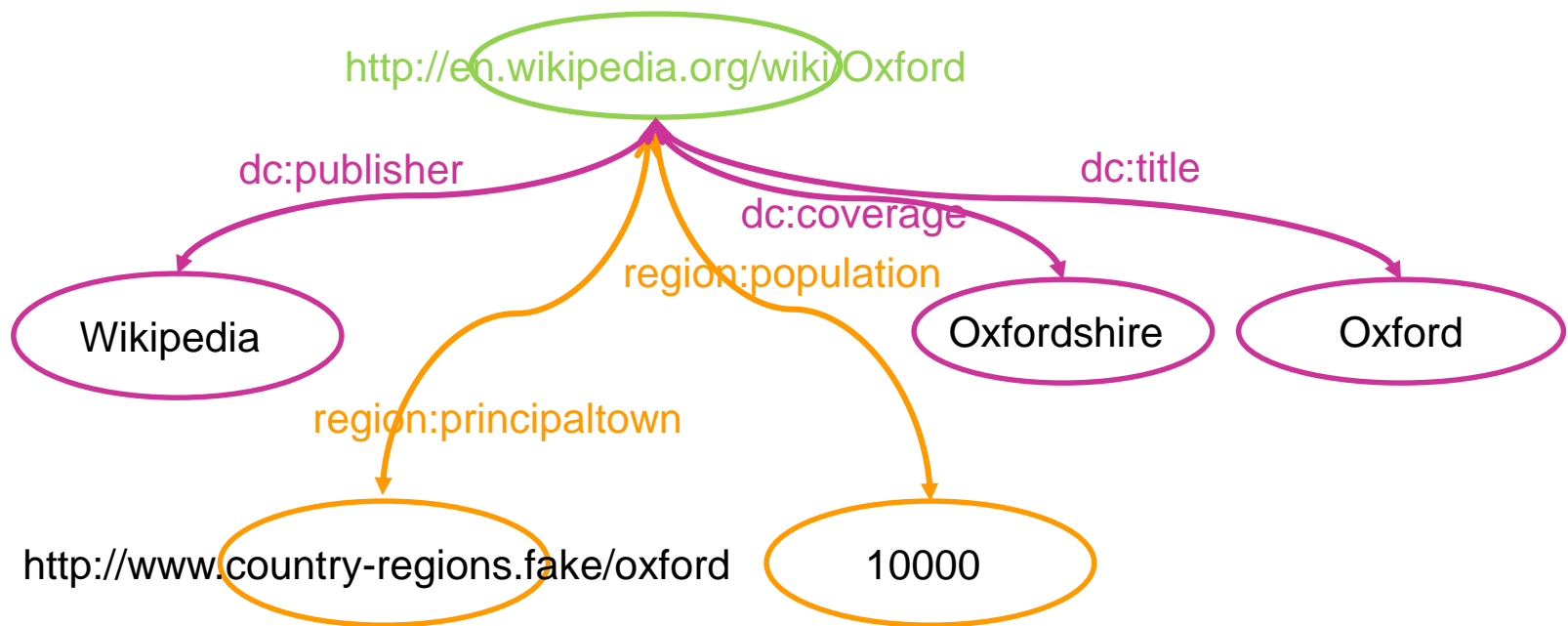
```
  dc:title "Oxford " ;
```

```
  dc:coverage "Oxfordshire " ;
```

```
  dc:publisher "Wikipedia " ;
```

```
  region:population "1000 " ;
```

```
  region:principaltown <http://www.country-regions.fake/oxford> .
```



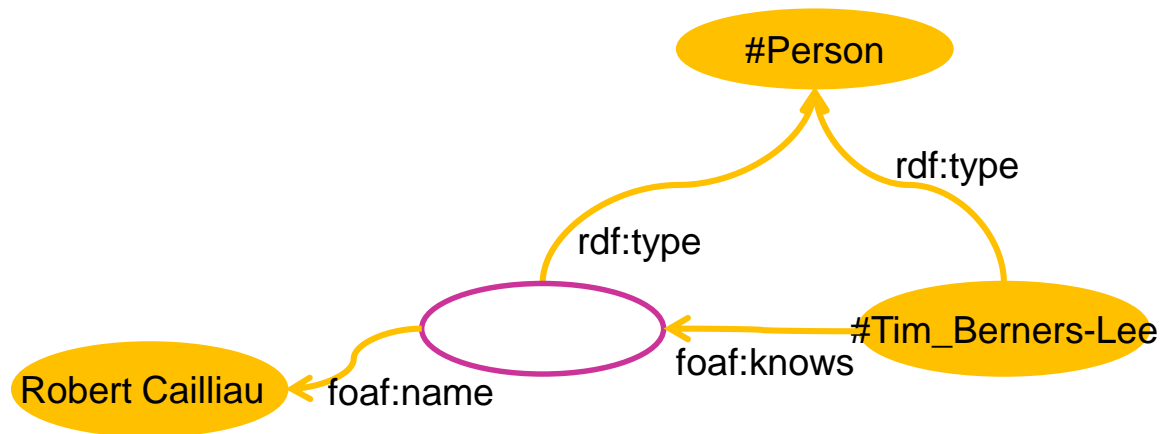
RDF semantics

- Semantics in model theory

- Triple: $\langle s, p, o \rangle$ is associated an atomic formula $p(o, s)$
- Document: existential closure of the conjunction of atomic formulas

- Interpretation in predicate logic

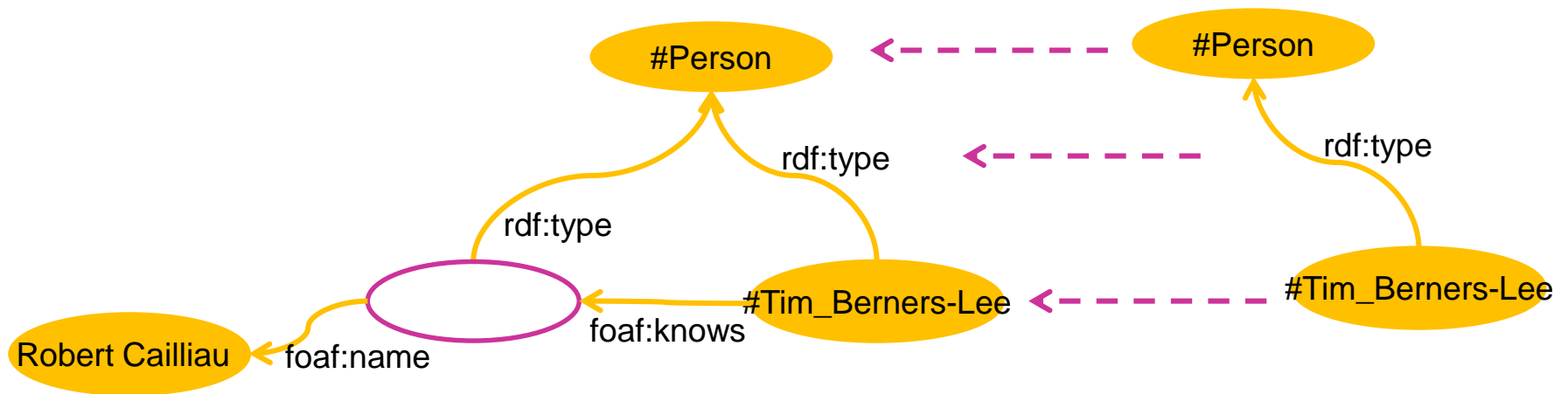
$\exists x, (\text{rdf:type}(\#Tim_Berners-Lee, \#Person) \wedge \text{rdf:type}(x, \#Person) \wedge \text{foaf:knows}(\#Tim_Berners-Lee, x) \wedge \text{foaf:name}(x, \text{"Robert Cailliau"}))$



RDF semantics

■ Reasoning mechanism

- An RDF document is the consequence of another RDF document iff there is a graph homomorphism between them
- Goal = identifies all the consequences of a set of predicates



RDF: strengths and limitations

- compared with DB
 - better manages semi-structured data
 - “universal”
 - Adapted to the web
 - But very verbose, little efficiency for storage and querying
- Semantics
 - Logical model / model theory

RDF and the semantic web

- Difficulties with meta-data
 - May be missing
 - May be conflicting or erroneous
 - Reflect a point of view
 - Refer to local / specific types
- Difficulties to query different structural options

```
<foaf:Person rdf:about="http://dbpedia.org/page/Tim_Berners-Lee">  
<ex:hasActivity rdf:resource=#researcher"/>  
</foaf:Person>
```

```
<foaf:Person rdf:about="http://dbpedia.org/page/Marie_Curie">  
rdf:type rdf:resource=#researcher"  
</foaf:Person>
```

```
<ex:researcher rdf:about="http://dbpedia.org/page/Albert_Einstein" />  
which persons are researchers ?
```

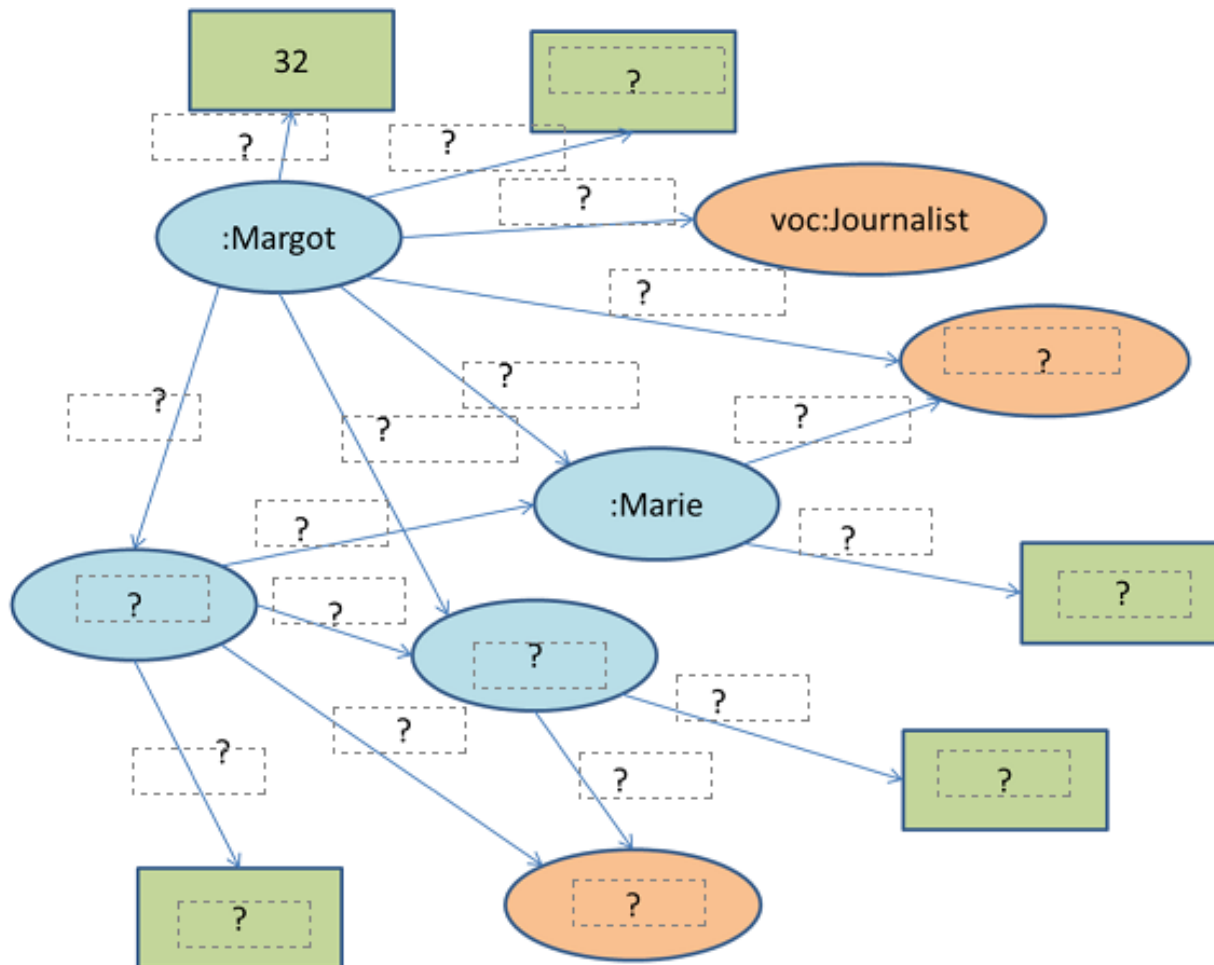
From RDF to RDFS

- **Need** to define properties for generic groups of individuals, such as the class of publishers, of organizations, or of persons
- **Solution:**
 - (XML) schema knowledge
 - RDF Schema (RDFS): part of the RDF W3C recommendation
 - Adequate for simple models, hierarchies of classes and instances

RDF:exercise 2

<https://www.fun-mooc.fr/courses/inria/41002S02/session02/>

- More about RDF
 - <https://www.w3.org/TR/rdf11-primer/>
- Margot est une femme journaliste, âgée de 32 ans, mariée à Arthur qui est un homme avec qui elle a deux enfants, Marie qui est une femme et Simon qui est un homme. Pour chaque personne, on spécifie aussi explicitement son prénom.
 - List the entities that will be resources
 - List the literals
 - Define voc, a vocabulary with the needed `rdf:Property`,
 - Fill the graph



Vocabulary

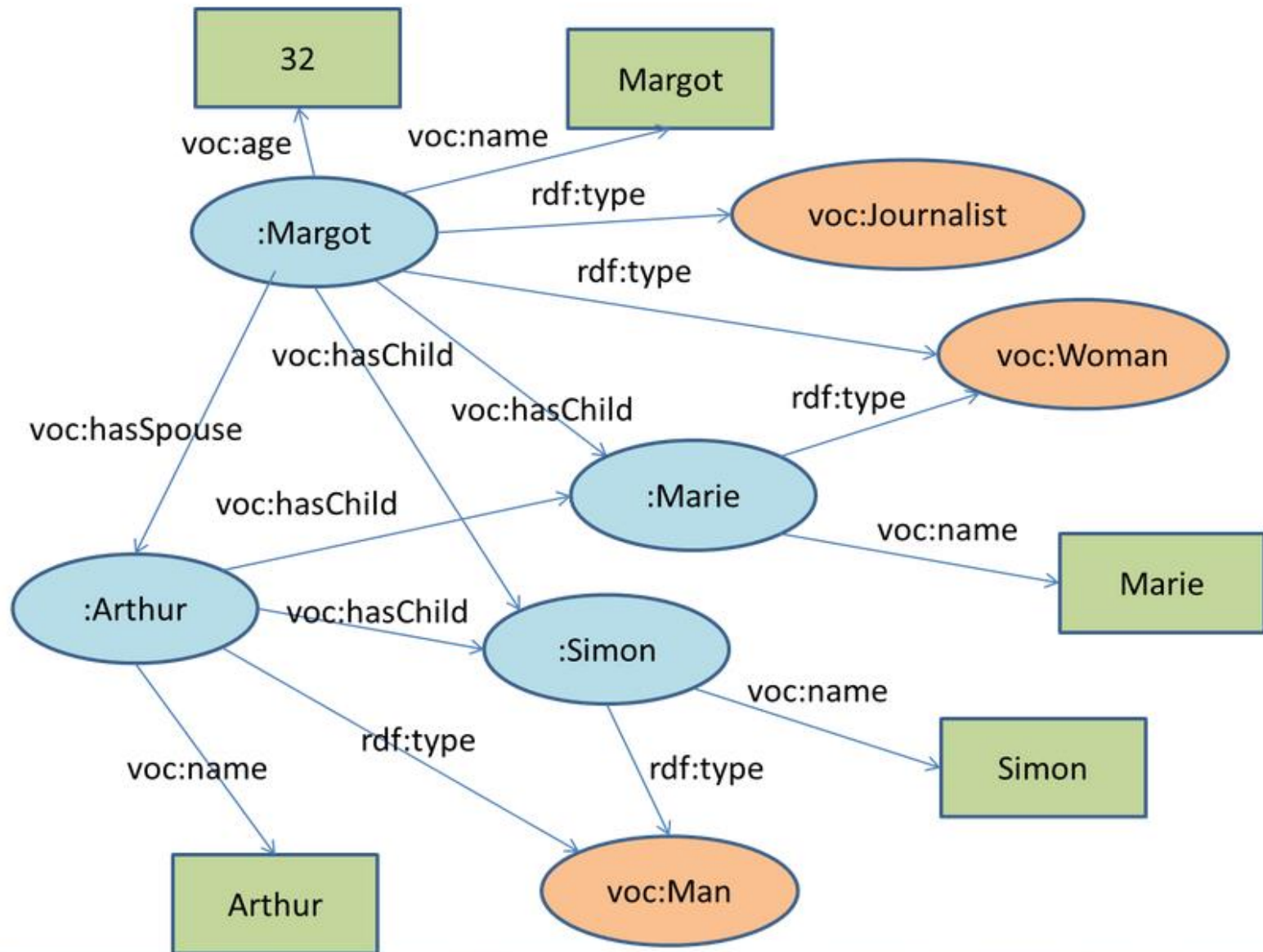
voc:hasSpouse
 voc:hasChild
 voc:Man
 voc:Woman
 voc:Journalist

Entities

:Margot
 :Marie
 :Simon
 :Arthur

Literals

« Marie »
 « Simon »
 « Arthur »
 « Margot »
 32



RDF-XML writing

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [ <!ENTITY vocabulaire "http://www.unice.fr/voc"> <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:voc="&vocabulaire;"
  xml:base="http://www.unice.fr/data">

  <AAA rdf:about="#Margot">
    <voc:name>Margot</voc:name>
    <voc:age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">32</voc:age>
    <BBB rdf:resource="#Arthur"></BBB>
    <voc:hasChild rdf:resource="#Simon"> </voc:hasChild>
    <voc:hasChild>
      <rdf:Description XXX="#Marie">
        <voc:name>Marie</voc:name>
        <rdf:type CCC="« voc:Woman"> </rdf:type>
      </rdf:Description>
    </voc:hasChild>
    <DDD rdf:resource="« voc:Journalist"></DDD>
  </AAA>
  <EEE rdf:about="#Arthur">
    <voc:name>Arthur</voc:name>
    <rdf:type rdf:resource="voc:Man">
    <voc:hasChild rdf:resource="#Simon"></voc:hasChild>
    <voc:hasChild rdf:resource="#Marie"></voc:hasChild>
  </EEE>
  <voc:Man rdf:about="#Simon">
    <voc:name>Simon</voc:name>
  </voc:Man>
</rdf:RDF>
```

AAA = voc:Woman

BBB = voc:hasSpouse

CCC = rdf:resource

DDD = rdf:type

EEE = rdf:Description

XXX = rdf:about

RDF TURTLE notation

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix voc: <http://www.unice.fr/voc#> .

@prefix xml: <http://www.w3.org/XML/1998/namespace> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.unice.fr/data#Margot> a voc:Journalist AAA BBB ;

 voc:age "32"^^xsd:string ;

 voc:hasChild <http://www.unice.fr/data#Marie>, <CCC>;

 voc:hasSpouse <http://www.unice.fr/data#Arthur> ;

 voc:name "Margot" .

<http://www.unice.fr/data#Arthur> DDD voc:Man ;

 voc:hasChild <http://www.unice.fr/data#Marie>, <http://www.unice.fr/data#Simon> ;

 voc:name "Arthur" .

<http://www.unice.fr/data#Marie> a voc:Woman ;

 voc:name « Marie" .

<EEE> a FFF ;

 GGG HHH .

Advanced checking

- To check the actual equivalence of the statements given as solutions of the two previous exercises, you can use RDF Translator, an on-line service

<http://rdf-translator.appspot.com/>

- You can use the W3C on-line validation service, <http://www.w3.org/RDF/Validator/>, to get the list of triples and visualize them as a graph.

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://en.wikipedia.org/wiki/Oxford	http://purl.org/dc/elements/1.1/title	"Oxford"
2	http://en.wikipedia.org/wiki/Oxford	http://purl.org/dc/elements/1.1/coverage	"Oxfordshire"
3	http://en.wikipedia.org/wiki/Oxford	http://purl.org/dc/elements/1.1/publisher	"Wikipedia"
4	http://en.wikipedia.org/wiki/Oxford	http://www.country-regions.fake/population	"10000"
5	http://en.wikipedia.org/wiki/Oxford	http://www.country-regions.fake/principaltown	http://www.country-regions.fake/oxford

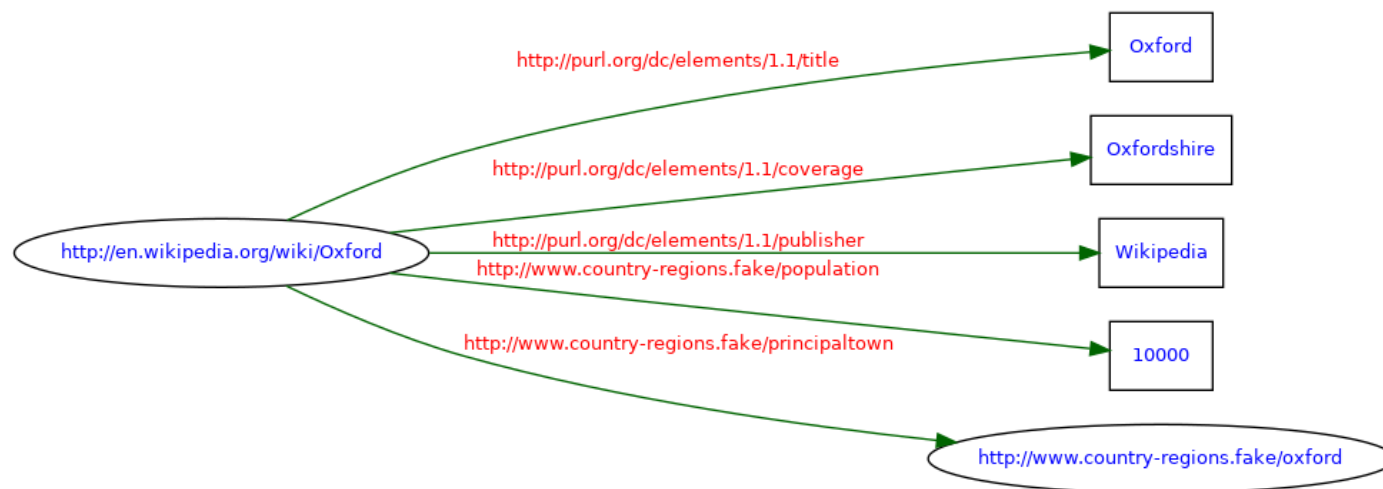
The original RDF/XML document

```

1: <?xml version="1.0"?>
2: <rdf:RDF
3:   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:   xmlns:dc="http://purl.org/dc/elements/1.1/"
5:   xmlns:region="http://www.country-regions.fake/">
6:   <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Oxford">
7:     <dc:title>Oxford</dc:title>
8:     <dc:coverage>Oxfordshire</dc:coverage>
9:     <dc:publisher>Wikipedia</dc:publisher>
10:    <region:population>10000</region:population>
11:    <region:principaltown rdf:resource="http://www.country-regions.fake/oxford"/>
12:   </rdf:Description>
13: </rdf:RDF>
14:

```

Graph of the data model



Checking RDF models

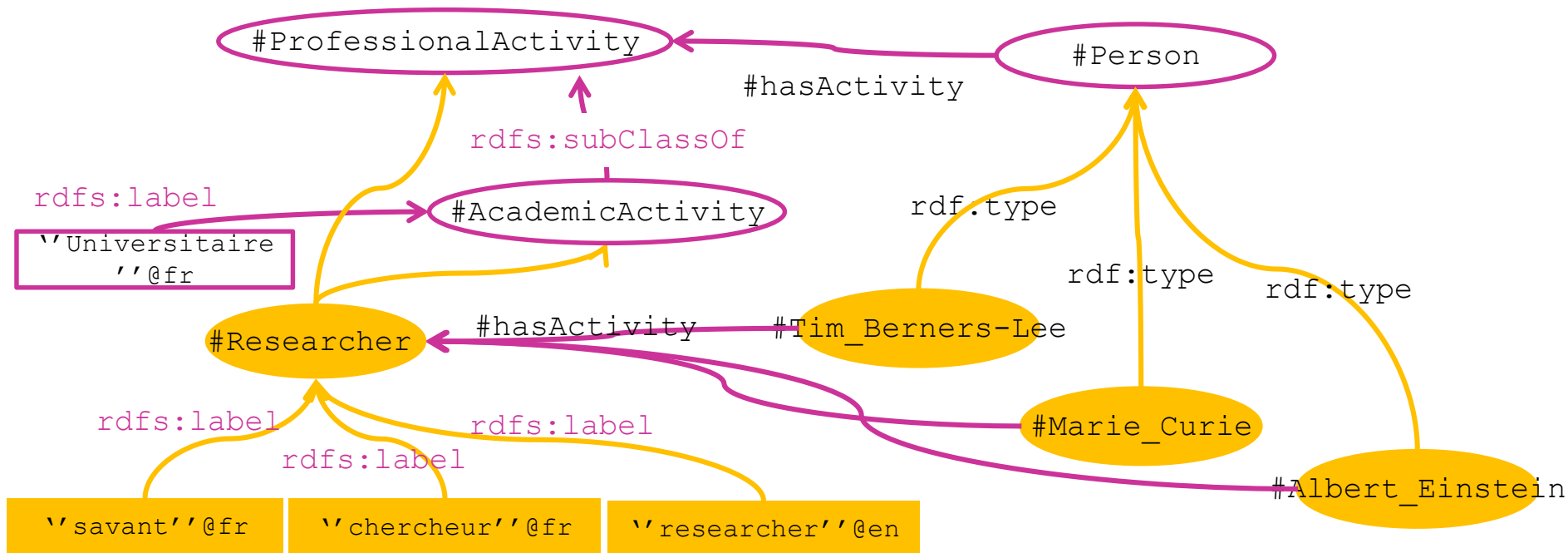
- Extracting RDF from RDFa annotated pages
 - RDFa Distiller and Parser
<http://www.w3.org/2012/pyRdfa/>
 - Y. Herman, web service
- Checking RDFa annotations
 - RDFa validator (web service)
<http://www.w3.org/2012/pyRdfa/Validator.html>

From RDF to RDFS

- **Need** to define properties for generic groups of individuals, such as the class of publishers, of organizations, or of persons
- **Solution:**
 - (XML) schema knowledge
 - RDF Schema (RDFS): part of the RDF W3C recommendation
 - Adequate for simple models, hierarchies of classes and instances

RDF SChema

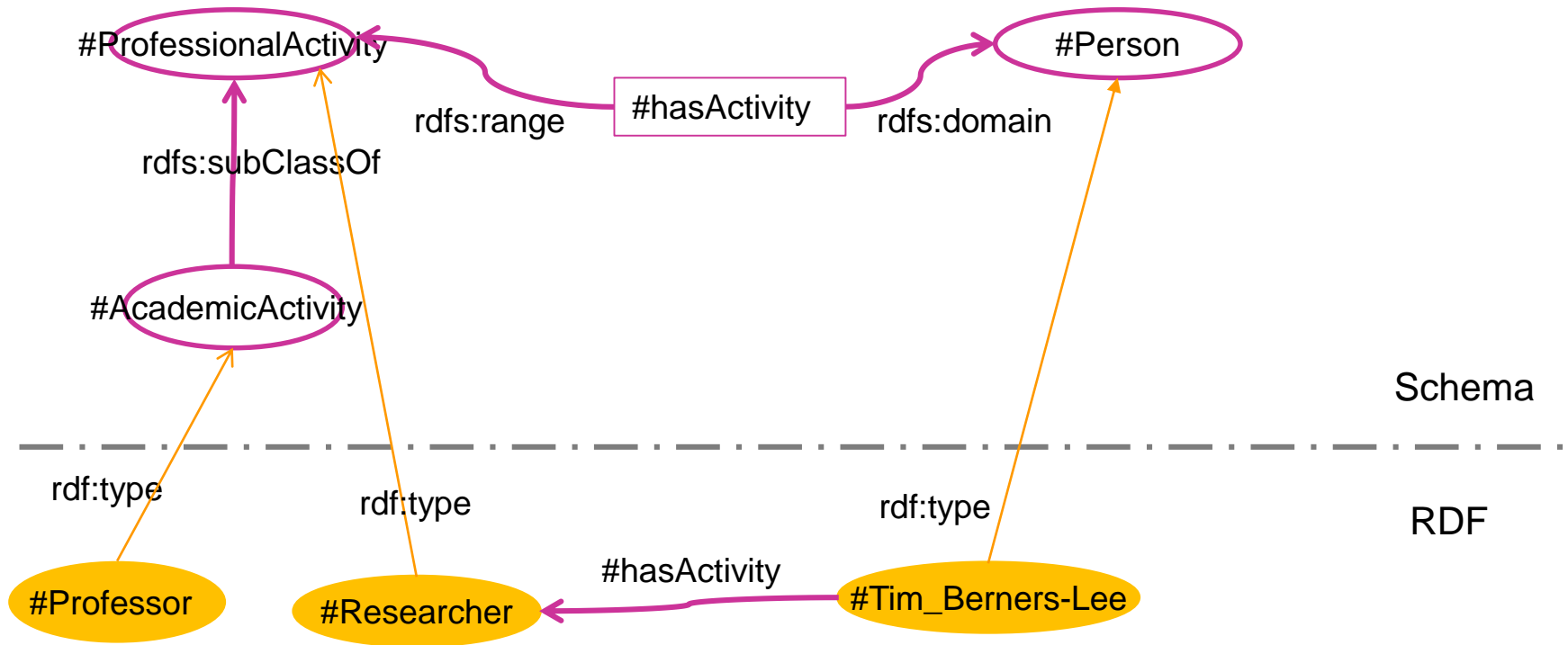
RDF and shared (formal) vocabularies



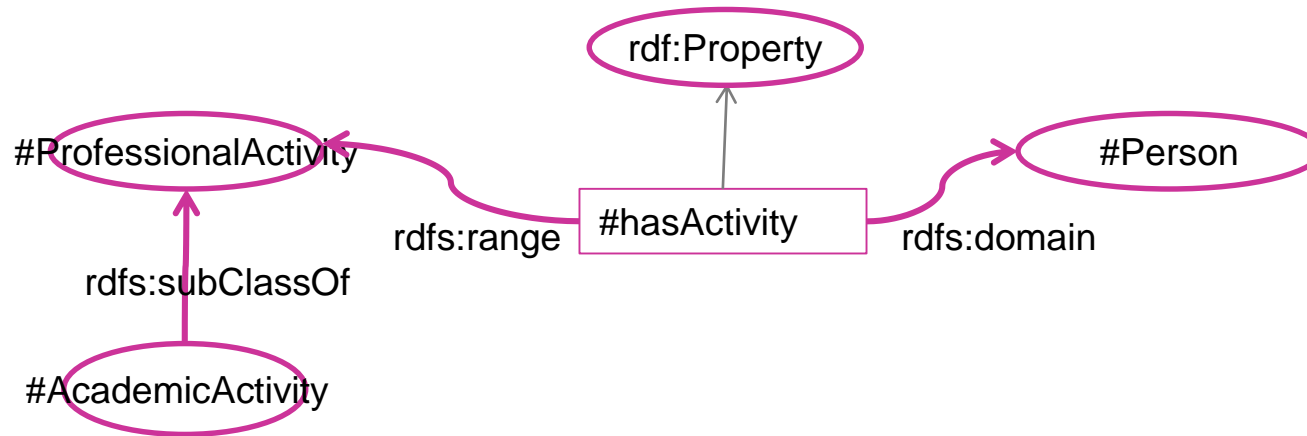
- Give the status of “class” to types `rdfs:Class`
- Organize types in a hierarchy `rdfs:subClassOf`
- Distinguish identifiers (ID) from natural language labelling of entities/classes: `rdfs:label`
- Reach a semantic agreement: **ontologies**

RDF and RDFS (RDFSchema)

- RDF schema: labelled and oriented graph made of RDF triples
- Property oriented



RDFS and RDF



RDFs

RDF

- **rdfs:Class**: defined as rdfs:Resources
- 3 triples define a property
 - Property rdf:type rdf:Property `#hasActivity rdf:type rdf:Property`
 - Property rdfs:domain Class1 `#hasActivity rdfs:domain #Person`
 - Property rdfs:range Class2 `#hasActivity rdfs:range #ProfessionalActivity`
- **rdfs:subClassOf** defined as a rdf:Property

RDFS `xml:base` : give a URI to your own vocabulary

XML Syntax

```
<rdf:RDF
  xml:base="http://www.irit.fr/MELODI/ontologies/humans.rdfs"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  (...)
</rdf:RDF>
```

Turtle syntax

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@base <http://www.irit.fr/MELODI/ontologies/humans.rdfs>
(...)
```

RDFS main classes

- **rdfs:Resource**
 - Super class of any rdfs resource
- **rdfs:Class**
 - Ex: N7:students, N7:N7_students, are types of rdfs:Class
 - A URI can have one or several types
- **rdfs:subClassOf**
 - Ex: N7:N7_students rdfs:subClassOf N7:students
- **rdfs:Literal**
 - Ex: xsd^year, integer, **rdfs:Datatype**, **rdfs:XMLLiteral** are sub-classes of (rdfs:subClassOf) rdfs:Literal
 - 2008, “10”, “Tim Berners Lee” are instances of (rdf:type) rdfs:Literal
- **rdf:Statement**
 - Instances: (#Tim_Berners-Lee, #hasBirthYear, “1955”)

RDFS semantics

- `rdf:type`
- `rdfs:subClassOf`
 - Instance-Class `#Michael rdf:type #ErasmusStudent`
 - Class-Class `#ErasmusStudent rdfs:subClassOf #Student`
 - Infered triple `#Michael rdf:type #Student`

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ex="http://www.irit.fr/ontology/ex#">

  <rdfs:Class rdf:about="#ErasmusStudent">
    <rdfs:subClassOf rdf:resource="#Student"/>
  </rdfs:Class>
  <ex:ErasmusStudent rdf:about="#Michael" />
</rdf:RDF>
```

RDFS property definition

■ rdfs:domain and rdfs:range

```
<rdf:Property rdf:ID="#isPartOf">  
  <rdfs:domain rdf:resource="#materialObject"/>  
  <rdfs:range rdf:resource="#materialObject"/>  
</rdf:Property>
```

■ rdfs:subPropertyOf

```
<rdf:Property rdf:about="#DoorFunctionalPartOf">  
  <rdfs:subPropertyOf rdf:resource="#isPartOf" />  
  <rdfs:domain rdf:resource="#handle" / → = "#handle rdfs:subClassOf #materialObject  
  <rdfs:range rdf:resource="#door" / → = "#door rdfs:subClassOf #materialObject  
</rdf:Property>
```

■ Semantics

$(\text{truc}, \#doorFunctionalPartOf, bidule) \rightarrow$
 $((\text{truc}, \text{rdf:type}, \text{rdf:resource}=\#handle) \wedge (\text{bidule}, \text{rdf:type}, \text{rdf:resource}=\#door) \wedge (\text{truc}, \#isPartOf, bidule))$

RDFS semantics

■ Semantics of `rdfs:subClassOf`

`a rdfs:subClassOf b` means

$$\forall x ((x \text{ rdf:type } a) \rightarrow (x \text{ rdf:type } b))$$

- Defines a hierarchy of classes
- Transitive and reflexive

■ Semantics of `rdfs:domain` and `rdfs:range`

`P rdfs:domain C1` and `P rdfs:range C2` means

$$\forall x,y (x \text{ P } y) \rightarrow ((x \text{ rdf:type } C1) \text{ and } (y \text{ rdf:type } C2))$$

RDFS property definition

- property restrictions are interpreted globally and conjunctively, e.g.
 - `ex:authorOf rdfs:range ex:Cookbook .`
 - `ex:authorOf rdfs:range ex:Storybook .`
- means: everything which is authored by somebody is both a cookbook and a storybook
- thus: always use most generic classes for domain/range statements

RDFS additional properties

- used to add human-readable information (comments or names)
 - **rdfs:label** : assigns an alternative name (encoded as literal) to an arbitrary resource
 - **rdfs:comment** : assigns a more comprehensive comment (also literal)
 - **rdfs:seeAlso**, **rdfs:definedBy** : refer to resources (URIs) containing further information about the subject resource

RDFS for lightweight ontologies

- Certain semantic aspects of a domain of interest
- Hierarchy of classes
- Domain and range of properties
 - `#Human rdfs:subClassOf #Primate`
 - `#SpeaksWith rdf:type rdfs:Property`
 - `#SpeaksWith rdfs:domain #Human`
 - `#SpeaksWith rdfs:range #Human`
 - `(a, #SpeaksWith, b) -> a rdf:type #Human` (and consequently also `a rdf:type #primate`)
- No cardinality constraint, no negation, no disjunction

RDFa: inserting RDF into HTML pages

```
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML+RDFa1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:foaf=http://xmlns.com/foaf/0.1/
  xmlns:dc=http://purl.org/dc/terms/                xml:lang="fr">
  <head>
    <title >Description de Tim Berners Lee en XHTML + RDFa</title>
  </head>
  <body>
    <div typeof="foaf:person" about="http://www.w3.org/People/Berners-Lee/card#i">
      Le <span rel="dc:creator" href="http://www.w3.org"> créateur du W3C</span> a
      pour nom Le <span property="foaf:name"> Timothy Berners-Lee</span> et pour surnom
      <span property="foaf:nick"> Timbl</span>.
    </div>
  </body>
</html>
```

RDF and RDFS vocabularies : examples

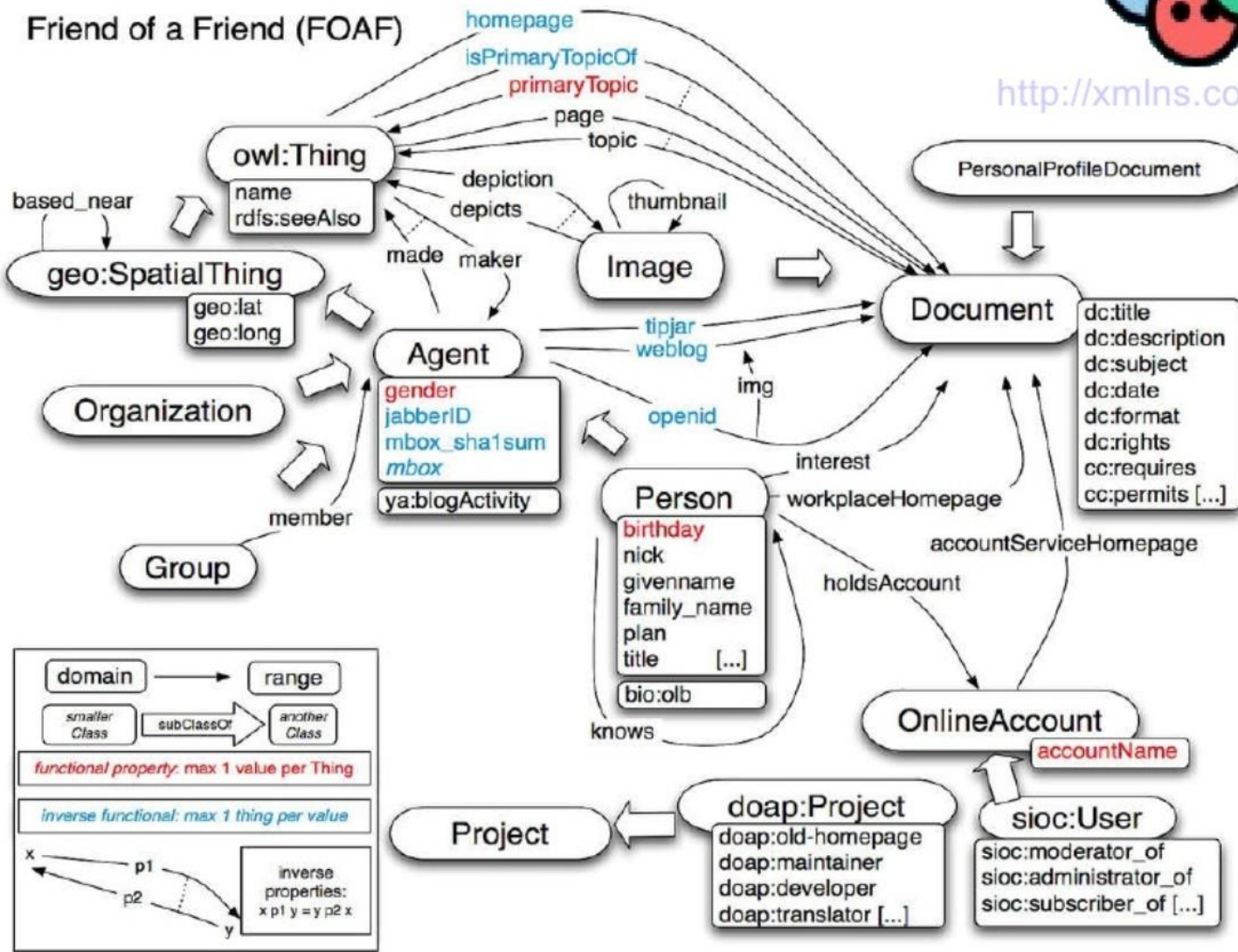
- FOAF persons
- DC Dublin Core :
<http://dublincore.org/schemas/rdfs/>
- SKOS : thesaurus and vocabularies

FOAF

Ontologie créée et maintenue par Dan Brickley et Libby Miller



<http://xmlns.com/foaf/>



Towards ontology representation

■ Class definition

- Identify the type, nature of each entity on the web
 - Tim Berners-Lee est une personne
 - Une personne est un être vivant
- In RDFs, any resource has a type which is `rdfs:Class`

■ Properties

- Entities have features or properties
- In RDFs, resources have `rdfs:properties`

■ Classes and properties are defined in RDFs (formal) vocabularies

Towards ontology representation

- Reasoning capabilities
 - Make explicit some of the hidden assumptions behind classes and properties
 - If a person A worksWith a person B, then B worksWith V
 - In RDF, such inferences have to be clearly written as rules or axioms or properties
 - worksWith is a simetric property
- Logic rewriting of RDFs models make it possible to infer new facts and new knowledge.