Resource Description Framework

Web-Based Knowledge Representation
February 25

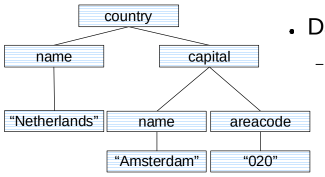
Jeen Broekstra
jbroeks@cs.vu.nl

Contents

- What was XML again?
- What is RDF?
- RDF in more detail
  - data model
  - syntax
- RDF Schema
  - Defining vocabulary
  - Schema syntax
- What are we still missing?

What was XML again?

```
<country name="Netherlands">
   <capital name="Amsterdam">
      <areacode>020</areacode>
   </capital>
</country>
```



- Syntax:
  - angle brackets, elements and attributes, etc.
- Data model:
  - ordered, labeled tree

So why not just use XML?

- No **agreement** on:
  - structure
    - is country a:
      - object?
      - class?
      - attribute?
      - relation?
      - something else?
    - what does nesting mean?
  - vocabulary
    - is country the same as nation?

```
<country name="Netherlands">
   <capital name="Amsterdam">
      <areacode>020</areacode>
   </capital>
</country>
```

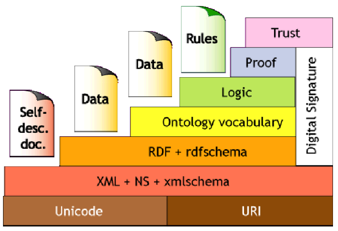
```
<nation>
   <name>Netherlands</name>
   <capital>Amsterdam</capital>
   <capital_areacode>
   020
   </capital_areacode>
</nation>
```

- Are the above XML documents the same?
- Do they convey the same information?
- Is that information machine-accessible?

What is RDF?

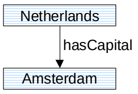
- RDF
  - stands for Resource Description Framework
  - is a W3C Recommendation (http://www.w3.org/RDF)
- RDF is a data model
  - for representing **metadata** (data about data)
  - for describing the semantics of information in a machine-accessible way
- What can you use it for?
  - intelligent information brokering
  - meaning-based computing
  - agent communication

So where does RDF fit in?



RDF in detail: the data model

- *statements* are (subject, predicate, object) *triples*:
  - (Netherlands, hasCapital, Amsterdam)



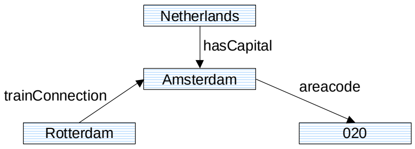
- statements describe *properties* of *resources*
- a resource is any object that can be pointed at by a URI:
  - a document, a picture, a paragraph on the Web
    - http://www.cs.vu.nl/index.html
  - a book in the library, 'real-world' objects
    - isbn://5031-4444-3333

What is a URI?

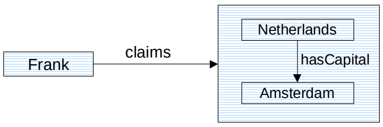
- URI = Uniform Resource Identifier
  - Standardized in RFC 2396
- “The generic set of all names/addresses that are short strings that refer to resources”
- URLs (Uniform Resource Locators) are a partiular type of URI, used on the WWW.
- In RDF, URIs often look like 'normal' URLs, often with fragment identifiers to point at specific parts of a document:
  - http://somedomain.com/some/path/to/file#fragmentID

Back to RDF: linking statements

- The subject of one statement can be the object of another
- such collections of statements form a directed, labeled graph



Reification: statements about statements



“Frank claims that the Netherlands has a capital called Amsterdam”

RDF syntax: XML

- RDF has an XML syntax that has a specific meaning:
  - every Description element describes a resource
  - every attribute or nested element inside a Description is a property of that resource

```
<Description about=”http://www.countries.org/countries#Netherlands”>
   <hasCapital resource=”http://www.cities.org/cities#Amsterdam”/>
</Description>
<Description about=”http://www.cities.org/cities#Amsterdam”>
   <areacode>020</areacode>
</Description>
```

- Does this solve the structure problem?
- Does this solve the vocabulary problem?

RDF/XML syntax: just a syntax

- different ways to write down the same model

```
<Description about=”http://www.countries.org/countries#Netherlands”>
   <hasCapital resource=”http://www.cities.org/cities#Amsterdam”/>
</Description>
<Description about=”http://www.cities.org/cities#Amsterdam”>
   <areacode>020</areacode>
</Description>
```

```
<Description about=”http://www.countries.org/countries#Netherlands”>
   <hasCapital resource=”http://www.cities.org/cities#Amsterdam”/>
</Description>
<Description about=”http://www.cities.org/cities#Amsterdam”
                  areacode=”020”/>
```

```
<Description about=”http://www.countries.org/countries#Netherlands”>
   <hasCapital resource=”http://www.cities.org/cities#Amsterdam”>
      <areacode>020</areacode>
   </hasCapital>
</Description>
```

RDF/XML syntax: namespaces

- like in ’normal’ XML, you can define namespaces to disambiguate elements and attributes:

```
<rdf:RDF xmlns:rdf=”http://www.w3.org/TR/1999/Rec-rdf-syntax-19990222”
           xmlns:geo=”http://www.geography.org/schema.rdf#”
           xmlns:words=”http://www.dictionary.org/schema.rdf#”>
 <rdf:Description rdf:about=”http://www.countries.org/countries#Netherlands”>
   <geo:hasCapital rdf:resource=”http://www.cities.org/cities#Amsterdam”/>
   <words:hasCapital> N </words:hasCapital>
 </rdf:Description>
 <rdf:Description rdf:about=”http://www.cities.org/cities#Amsterdam”>
   <geo:areacode>020</geo:areacode>
 </rdf:Description>
</rdf:RDF>
```

So what can we use this for?

- we can:
  - make explicit statements about web resources
  - have the machine
    - know that these are statements
    - know how the statements relate
    - compare values
- BUT
  - we still miss a way to define a vocabulary:
    - should we use ’country’ or ’nation’?
    - Is the Netherlands a country? Are there more countries? What properties can countries have?

RDF Schema

- RDF gives a data model for meta data annotation, and a way to write it down in XML, but it can not define the vocabulary for a domain.
- **RDF Schema** allows you to define vocabulary terms and the relations between these terms
  - It gives ’extra meaning’ to particular RDF predicates and resources
  - this ’extra meaning’, or semantics, define how a term should be interpreted

RDF Schema (2)

- RDF Schema terms (a few examples)
  - Class, Property
  - type, subClassOf, domain, range
- These terms are the RDF Schema building blocks, or **core primitives**.
- Vocabulary definition with these terms:
  - <Country, **type**, **Class**>
  - <Capital, **subClassOf**, City>
  - <hasCapital, **domain**, Country>
- Notice: these are just RDF statements, but RDF Schema terms are used to give extra meaning

The semantics of RDF Schema

- The ’extra meaning’ or **semantics** of RDF Schema are expressed in natural language:
  - 2.3.2 rdfs:subClassOf

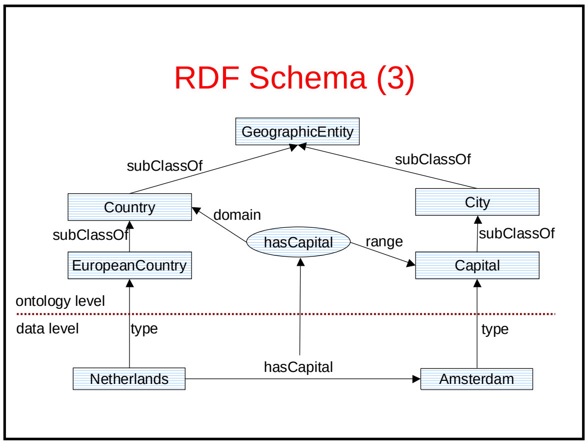
    “This property specifies a subset/superset relation between classes. The rdfs:subClassOf property is transitive. If class A is a subclass of some broader class B, and B is a subclass of C, then A is also implicitly a subclass of C. Consequently, resources that are instances of class A will also be instances of class C, since A is a subset of both B and C. Only instances of rdfs:Class can have the rdfs:subClassOf property and the property value is always of rdf:type rdfs:Class. A class may be a subclass of more than one class.”
- Question: is A a subclass of A?

RDF Model Theory

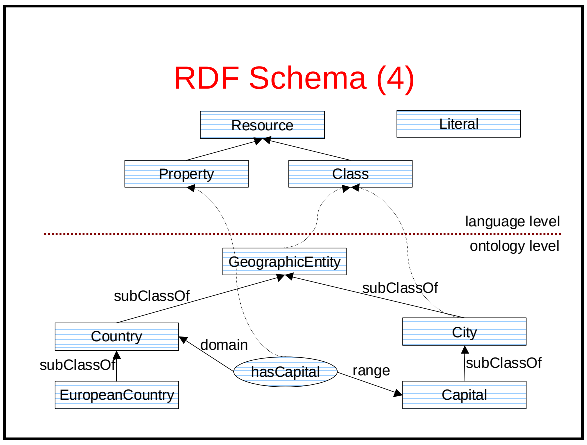
- W3C Working Draft
  - http://www.w3.org/TR/rdf-mt
- set-theoretical semantics for RDF and RDFS
- specifies entailment, for example:
  - [rdfs7b] (reflexivity)

    ```
    (xxx, rdf:type, rdfs:Class)
     =>
    (xxx, rdfs:subClassOf, xxx)
    ```
  - [rdfs8] (transitivity)

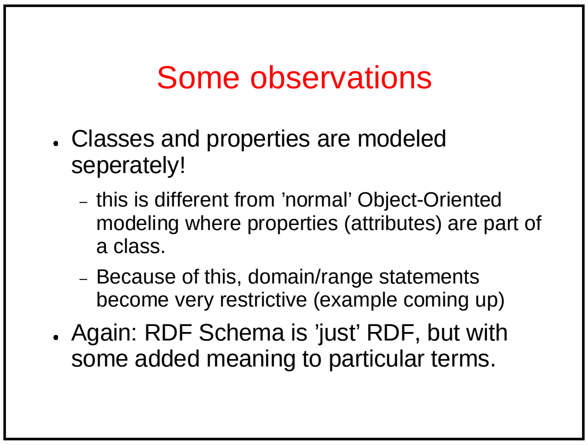
    ```
    (xxx, rdfs:subClassOf, yyy) &
    (yyy, rdfs:subClassOf, zzz)
     =>
    (xxx, rdfs:subClassOf, zzz)
    ```



RDF Schema (3)

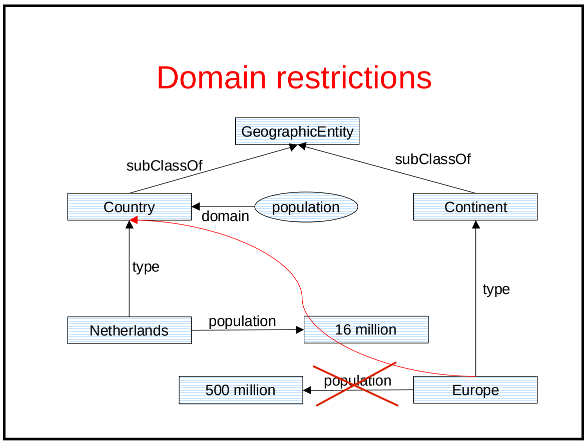


RDF Schema (4)

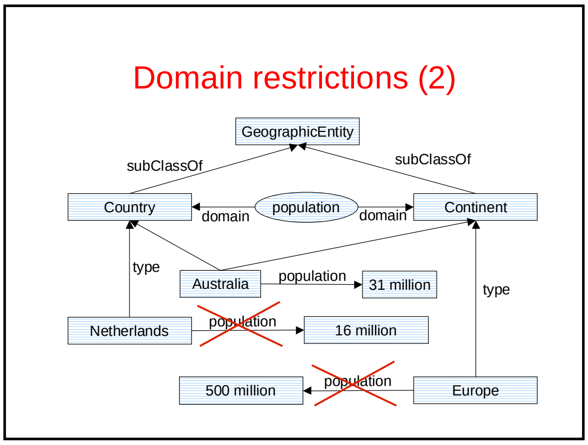


Some observations

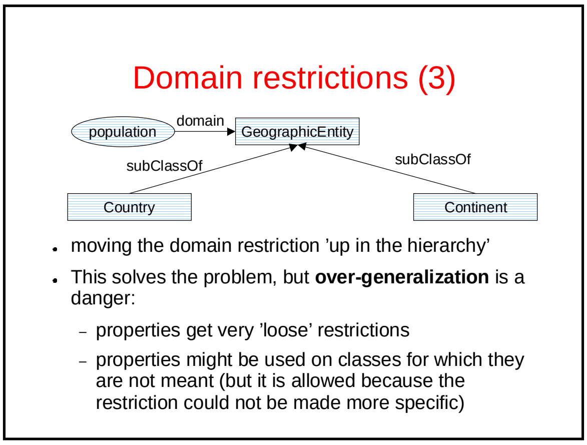
- Classes and properties are modeled seperately!
  - this is different from 'normal' Object-Oriented modeling where properties (attributes) are part of a class.
  - Because of this, domain/range statements become very restrictive (example coming up)
- Again: RDF Schema is 'just' RDF, but with some added meaning to particular terms.



Domain restrictions



Domain restrictions (2)



Domain restrictions (3)

- moving the domain restriction 'up in the hierarchy'
- This solves the problem, but **over-generalization** is a danger:
  - properties get very 'loose' restrictions
  - properties might be used on classes for which they are not meant (but it is allowed because the restriction could not be made more specific)

RDF Schema syntax

- Class definition

```
<rdf:Description rdf:about=”http://www.geography.org/schema.rdf#Country”>
   <rdf:type rdf:resource=”http://www.w3.org/2000/01/rdf-schema#Class”/>
   <rdfs:subClassOf
   rdfs:resource=”http://www.geography.org/schema.rdf#GeographicEntity”/>
</rdf:Description>
```

```
<rdfs:Class rdf:about=”http://www.geography.org/schema.rdf#Country”>
   <rdfs:subClassOf
   rdfs:resource=”http://www.geography.org/schema.rdf#GeographicEntity”/>
</rdfs:Class>
```

- Property definition

```
<rdf:Property rdf:about=”http://www.geography.org/schema.rdf#hasCapital”>
   <rdfs:domain rdfs:resource=”http://www.geography.org/schema.rdf#Country”/>
   <rdfs:range rdfs:resource=”http://www.geography.org/schema.rdf#Capital”/>
</rdfs:Class>
```

Putting it all together

- The schema file: http://www.geography.org/schema.rdf

```
<rdf:RDF xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
         xmlns:rdfs=”http://www.w3.org/2000/01/rdf-schema#”>
   <rdfs:Class rdf:about=”#Country”>
      <rdfs:subClassOf rdfs:resource=”#GeographicEntity”/>
   </rdfs:Class>
  <rdf:Property rdf:about=”#hasCapital”>
     <rdfs:domain rdfs:resource=”#Country”/>
     <rdfs:range rdfs:resource=”#Capital”/>
  </rdf:Property>
</rdf:RDF>
```

- The (meta)data file: wherever you like

```
<rdf:RDF xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
         xmlns:geo=”http://www.geography.org/schema.rdf#”>
   <geo:Country rdf:about=”#Netherlands”>
      <geo:hasCapital rdfs:resource=”#Amsterdam”/>
   </geo:Country>
   <geo:Capital rdf:about=”#Amsterdam”/>
</rdf:RDF>
```

So why use RDF / RDFS?

- Because it’s there!
  - RDF and RDF Schema provide a common agreement, an **open standard** for annotating web resources and making their semantics explicit.
  - Technically speaking it’s not the best possible solution, but a compromise
    - we trade in some convenience for **interoperability**: the ability to communicate with arbitrary partners based on the fact that we both use RDF

Ontology language?

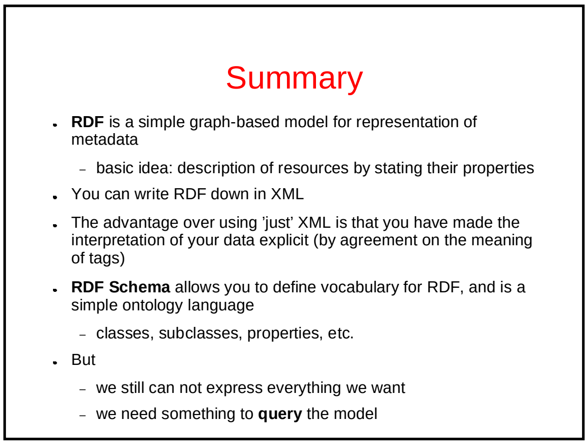
- Ontology: a formal specification of a shared conceptualization
- RDF Schema allows:
  - specification
(we have just seen that)
  - sharing
(because it is an open, web-based standard)
  - formality?
- Is RDF Schema expressive enough?

What is still missing?

- Cardinality constraints
  - “a country can have exactly *one* capital”
- Conjunction, disjunction, negation, equivalence
  - “countries and cities are disjoint: something can not be both a city and a country”
- Localized constraints
  - “when the property 'population' is used on a city, its value must be between 20.000 and 10 million”
- A way to access this information!
  - having it written down is nice and all, but if you want to use it for question answering you need a **query language** (like SQL for databases)

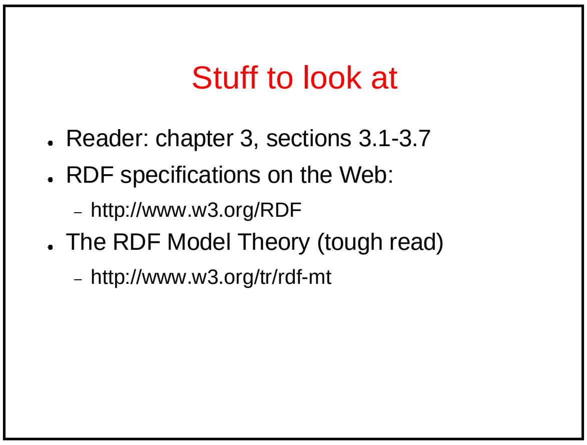
Research activities

- W3C Semantic Web Activity
  http://www.w3.org/2001/sw/
  - working on revision of the specs for RDF and RDF Schema
  - working on a formal spec: the RDF Model Theory
  - working on more expressive ontology language: **OWL**
    - OWL is derived from OIL, a language that was developed here at the Vrije Universiteit
- Several international projects in which the VU is involved
  - SWAP, OntoWeb, Wonderweb, Obelix, ...



Summary

- **RDF** is a simple graph-based model for representation of metadata
  - basic idea: description of resources by stating their properties
- You can write RDF down in XML
- The advantage over using 'just' XML is that you have made the interpretation of your data explicit (by agreement on the meaning of tags)
- **RDF Schema** allows you to define vocabulary for RDF, and is a simple ontology language
  - classes, subclasses, properties, etc.
- But
  - we still can not express everything we want
  - we need something to **query** the model



Stuff to look at

- Reader: chapter 3, sections 3.1-3.7
- RDF specifications on the Web:
  - http://www.w3.org/RDF
- The RDF Model Theory (tough read)
  - http://www.w3.org/tr/rdf-mt