# Chapter 13
# Metadata and Semantic Web

> “The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”
> – Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001

- application-level description of Web sources and facts
- metadata (e.g. keywords in header section of Web pages)
  problem: which keywords to use?
- “Ontologies”: describe and relate concepts
  (e.g., geographics, bioinformatics, ...)
- Portals that provide information and services on a given topic.

Literature: e.g. Eckstein/Eckstein: “XML und Datenmodellierung” (dpunkt), Ch. 6,7



## Ontologies

Description of concepts

- what concepts
- properties
- relationships (hierarchy, uses-relationship, any others)

Example: Biology

biological *taxonomy*: animals, plants, mammals, fishes, insects, carnivores, felidae, canidae, bovinae

- taxonomy: hierarchy
- sometimes also: typical properties (base for taxonomy)
- Ontology: additionally describe all kinds of relationships (e.g. that carnivores eat animals)
- more than schema: schema with annotations *what* notions mean

## WEB PORTALS

Provide services/access for a given ontology, e.g.

- tourism, traveling
- bioinformatics

Communication with clients, servers, and users via notions of an ontology:

- information about the information/services provided by resources ("Yellow-pages", represented e.g. in RDF (Resource Description Framework))
- queries and results/answers according to a given format (data in XML, according to a DTD/XML Schema)
- mapping from individual DTD/XML Schemas to global ontology
- Portal: maps different schemas to the common one (based on ontology).



## USE ONTOLOGIES FOR DESCRIBING NON-STRUCTURED CONTENTS

- describe content of pictures and films
  - pictures: description of the picture (where some item is found on the picture), and of its "message" ("what is depicted there" (application level))
  - films: additionally, temporal aspects

important: common ontologies for each of the aspects.

Portals then use these annotations:

**Example**

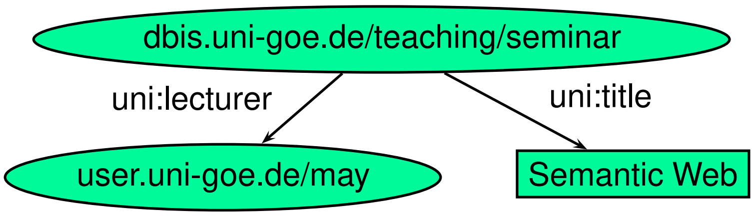
An annotated information video on the "BSc/MSc studies in Applied Computer Science at Göttingen University" could contribute to answers:

- how are german BSc/MSc studies structured, what are ECTS points?
- what is the contents of "Informatik I" in GOE?
- where can I study bioinformatics in Germany?
- (for local students:) how many CP do I need in the "Application Area"?

## 13.1 RDF: Resource Description Framework

Extension of the ideas of conceptual modeling (ER-Model, UML)

- first Working Draft: 1997
- triple-based "RDF statements": (subject, predicate, object)
- graphical representation (⇒ RDF graph), XML representation
- subject and object are *resources*
  (cf. terminology for XLink)
- a resource is everything ... that can be described by a URI
  (graphical notation: an ellipse)
  - a Web page, a book, a lecture ...






### Outline

- use `http://www.w3.org/RDF/Validator/` for visualization
  - RDF graph
  - RDF triples
- first example: resources are identified with Web pages – i.e. actual resources
- second example: we don't need actual resources - virtual resources are sufficient if people agree about them.

## Example (with actual resources)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:mon="http://www.mondial.de/mondial.rdfs#">
  <rdf:Description rdf:about="http://www.deutschland.de">
    <mon:name>Germany</mon:name>
    <mon:code>D</mon:code>
    <mon:has_province>
      <rdf:Description rdf:about="http://www.niedersachsen.de">
        <mon:name>Niedersachsen</mon:name>
        <mon:has_city>
          <rdf:Description rdf:about="http://www.hannover.de">
            <mon:name>Hannover</mon:name>
            <mon:in_country rdf:resource="http://www.deutschland.de"/>
            <mon:population>
              <rdf:Description> <mon:year>1995</mon:year>
                                <mon:value>525763</mon:value>
              </rdf:Description>
            </mon:population>
          </rdf:Description>
        </mon:has_city>
        <mon:has_capital rdf:resource="http://www.hannover.de"/>
      </rdf:Description>
    </mon:has_province>
  </rdf:Description>
</rdf:RDF>
```

- there can be different RDF files that describe Germany, Lower Saxonia (e.g. more cities) and Hannover in more detail.



## Example B (with actual resources)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.education.de/uni.rdfs#">
  <rdf:Description rdf:about="http://dbis.informatik.uni-goettingen.de">
    <uni:name>DBIS</uni:name>
    <uni:uni rdf:resource="http://www.uni-goettingen.de"/>
    <uni:institute rdf:resource="http://www.informatik.uni-goettingen.de"/>
    <uni:person>
      <rdf:Description rdf:about="http://user.informatik.uni-goettingen.de/~may">
        <uni:name>Wolfgang May</uni:name>
      </rdf:Description>
    </uni:person>
    <uni:leader rdf:resource="http://user.informatik.uni-goettingen.de/~may"/>
    <uni:lecture>
      <rdf:Description rdf:about="http://dbis.informatik.uni-goettingen.de/Teaching/SSD">
        <uni:name>Semistructured Data and XML</uni:name>
        <uni:lecturer rdf:resource="http://user.informatik.uni-goettingen.de/~may"/>
        <uni:assistance rdf:resource="http://user.informatik.uni-goettingen.de/~behrends"/>
        <uni:assistance rdf:resource="http://user.informatik.uni-goettingen.de/~fritzen"/>
      </rdf:Description>
    </uni:lecture>
  </rdf:Description>
</rdf:RDF>
```

- there can be different RDF files that describe the university at Göttingen, the CS institute, E.Behrends and O.Fritzen in more detail.

## RDF AS AN XML APPLICATION

Root element type: <rdf:RDF >
contains <rdf:Description> elements

### <rdf:Description> Element

(subject, predicate, object)

- which resource is described: @rdf:about="*uri*"
- subelements for describing properties:
  - element name: name of the property
  - value of the property:
    - element contents:
      text content or one or more nested <rdf:Description> elements
    - attribute @rdf:resource="*uri*": property points to another resource that has an RDF description of its own elsewhere



## URIS AND URLS

URI according to RFC 2396 (http://www.ietf.org/rfc/rfc2396.txt)

- a URI can be a locator (URL), a "name" (URN), or a "general" URI.
  General form: *scheme*:*hierarchical_part*
- URIs: a sequence of characters; often with a hierarchical structure
- URLs are those URIs that identify resources via their physical access mechanism (i.e., http, ftp, gopher, file, ...), in general named after a protocol.
  http://www.informatik.uni-goettingen.de/people.html
  mailto:may@informatik.uni-goettingen.de
  news:de.comp.text.tex
- URI schemes can be "registered" at www.IANA.org (Internet Assigned Numbers Authority)
  example (non-registered): isbn:3-89722-153-5
- URIs serve as agreed *identifiers* in a certain community.
- some of them are valid URLs, some of them not.

RDF uses general URIs, not only URLs.

## Example (with virtual resources)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:mon="http://www.mondial.de/mondial.rdfs#">
  <rdf:Description rdf:about="mon://country/de">
    <mon:name>Germany</mon:name>
    <mon:code>D</mon:code>
    <mon:has_province>
      <rdf:Description rdf:about="mon://country/de/prov/niedersachsen">
        <mon:name>Niedersachsen</mon:name>
        <mon:has_city>
          <rdf:Description rdf:about="mon://country/de/prov/niedersachsen/city/hannover">
            <mon:name>Hannover</mon:name>
            <mon:in_country rdf:resource="mon://country/de"/>
            <mon:population>
              <rdf:Description> <mon:year>1995</mon:year>
                                <mon:value>525763</mon:value>
              </rdf:Description>
            </mon:population>
          </rdf:Description>
        </mon:has_city>
        <mon:has_capital rdf:resource="mon://country/de/prov/niedersachsen/city/hannover"/>
      </rdf:Description>
    </mon:has_province>
  </rdf:Description>
</rdf:RDF>
```

- the graph is the same, only the URIs differ



## Example B (with virtual resources)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.education.de/uni.rdfs#">
  <rdf:Description rdf:about="bla://unigoe.de/inf/dbis">
    <uni:name>DBIS</uni:name>
    <uni:uni rdf:resource="bla://unigoe.de"/>
    <uni:institute rdf:resource="bla://unigoe.de/inf"/>
    <uni:person>
      <rdf:Description rdf:about="bla://unigoe.de/inf/people/may">
        <uni:name>Wolfgang May</uni:name>
      </rdf:Description>
    </uni:person>
    <uni:leader rdf:resource="bla://unigoe.de/inf/people/may"/>
    <uni:lecture>
      <rdf:Description rdf:about="bla://unigoe.de/inf/Teaching/SSD">
        <uni:name>Semistructured Data and XML</uni:name>
        <uni:lecturer rdf:resource="bla://unigoe.de/inf/people/may"/>
        <uni:assistance rdf:resource="bla://unigoe.de/inf/people/behrends"/>
        <uni:assistance rdf:resource="bla://unigoe.de/inf/people/fritzen"/>
      </rdf:Description>
    </uni:lecture>
  </rdf:Description>
</rdf:RDF>
```

- the graph is the same, only the URIs differ

## URIs

- basic idea of RDF: resources are globally identified by their URI
  - can be a real URL
  - can be a virtual URI
  - RDF instances can mix them (e.g. when describing Web pages, their authors etc.)
- virtual URIs in a global application scenario
  - virtual structuring of the resources
  - globally known and agreed URI schema
  - all members of a "community" can describe the resources by RDF



## URI Representation/Construction Mechanisms

- describe a remote resource via its full URI (as above)
  - attribute @rdf:about="*uri*" identifies a remote resource
- define "new" resources:
  base URI for the whole RDF instance, extended with local name parts:
  - set xml:base="*uri*" (e.g. in the root element)
  - @rdf:ID="*localpart*": defines and describes a resource; the resource's global URI is then composed as *xmlbase#localpart*
  - then, use @rdf:resource="*#localpart*" in the object position for referencing it.
- only locally known IDs:
  - attribute @rdf:nodeID= "*name*": defines and describes a *local* resource that can be referenced only inside the same RDF instance by its ID
  - then, use @rdf:nodeID="*id*" in the object position of a property instead of @rdf:resource="*uri*"

## Example (with base URI and local name parts)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:mon="http://www.mondial.de/mondial.rdfs#"
         xml:base="mon:mondial">
  <rdf:Description rdf:ID="de">
    <mon:name>Germany</mon:name>
    <mon:code>D</mon:code>
    <mon:has_province>
      <rdf:Description rdf:ID="de-prov-ndrs">
        <mon:name>Niedersachsen</mon:name>
        <mon:has_city>
          <rdf:Description rdf:ID="de-city-hann">
            <mon:name>Hannover</mon:name>
            <mon:in_country rdf:resource="#de"/>
            <mon:population>
              <rdf:Description> <mon:year>1995</mon:year>
                                <mon:value>525763</mon:value>
              </rdf:Description>
            </mon:population>
          </rdf:Description>
        </mon:has_city>
        <mon:has_capital rdf:resource="#de-city-hann"/>
      </rdf:Description>
    </mon:has_province>
  </rdf:Description>
</rdf:RDF>
```

- global URIs are e.g. mon:mondial#de, mon:mondial#de-city-hann



## Example (with local IDs)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:mon="http://www.mondial.de/mondial.rdfs#">
  <rdf:Description rdf:nodeID="de">
    <mon:name>Germany</mon:name>
    <mon:code>D</mon:code>
    <mon:has_province>
      <rdf:Description rdf:nodeID="de-prov-ndrs">
        <mon:name>Niedersachsen</mon:name>
        <mon:has_city>
          <rdf:Description rdf:nodeID="de-city-hann">
            <mon:name>Hannover</mon:name>
            <mon:in_country rdf:nodeID="de"/>
            <mon:population>
              <rdf:Description> <mon:year>1995</mon:year>
                                <mon:value>525763</mon:value>
              </rdf:Description>
            </mon:population>
          </rdf:Description>
        </mon:has_city>
        <mon:has_capital rdf:nodeID="de-city-hann"/>
      </rdf:Description>
    </mon:has_province>
  </rdf:Description>
</rdf:RDF>
```

- these resources cannot be referenced by other RDF instances.

## Abbreviated Forms and Further Functionality

- short form: property attributes instead of property subelements
- omit "blank" description nodes (those without a relevant ID) by
  <*property-name* rdf:parseType="Resource"> ... </*property-name*>
- additional ways to describe collections (aboutEach, parseType="Collection")



### Example: Abbreviated Forms

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:mon="http://www.mondial.de/mondial.rdfs#">
  <rdf:Description rdf:about="mon://country/de"
    mon:name="Germany" mon:code="D">
    <mon:has_province>
      <rdf:Description rdf:about="mon://country/de/prov/niedersachsen">
        <mon:name>Niedersachsen</mon:name>
        <mon:has_city>
          <rdf:Description rdf:about="mon://country/de/prov/niedersachsen/city/hannover"
               mon:name="Hannover">
            <mon:in_country rdf:resource="mon://country/de"/>
            <mon:population rdf:parseType="Resource">
              <mon:year>1995</mon:year>
              <mon:value>525763</mon:value>
            </mon:population>
          </rdf:Description>
        </mon:has_city>
        <mon:has_capital rdf:resource="mon://country/de/prov/niedersachsen/city/hannover"/>
      </rdf:Description>
    </mon:has_province>
  </rdf:Description>
</rdf:RDF>
```

## Example: a Collection-Valued Property

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.education.de/uni.rdfs#">
  <rdf:Description rdf:about="http://dbis.informatik.uni-goettingen.de">
    <uni:name>DBIS</uni:name>
    <uni:uni rdf:resource="http://www.uni-goettingen.de"/>
    <uni:institute rdf:resource="http://www.informatik.uni-goettingen.de"/>
    <uni:person>
      <rdf:Description rdf:about="http://user.informatik.uni-goettingen.de/~may"
                 uni:name="Wolfgang May"/>
    </uni:person>
    <uni:leader rdf:resource="http://user.informatik.uni-goettingen.de/~may"/>
    <uni:lecture>
      <rdf:Description rdf:about="http://dbis.informatik.uni-goettingen.de/Teaching/SSD">
        <uni:name>Semistructured Data and XML</uni:name>
        <uni:lecturer rdf:resource="http://user.informatik.uni-goettingen.de/~may"/>
        <uni:semesters>
         <rdf:Seq>
          <rdf:li rdf:resource="http://dbis.informatik.uni-goettingen.de/Teaching/SSD/SS03"/>
          <rdf:li rdf:resource="http://dbis.informatik.uni-goettingen.de/Teaching/SSD/SS04"/>
         </rdf:Seq>
        </uni:semesters>
      </rdf:Description>
    </uni:lecture>
  </rdf:Description>
</rdf:RDF>
```

- note the names _1 and _2 for the sequence elements in the graph.



# TYPING

The <rdf:Description> element can contain additional type information about the described resource:

- subelement <rdf:type rdf:resource="*uri*">:
  (reference to the) type of the resource (not only the type name, but a uri that "represents" the type ... see later).
- the value of the rdf:type property is a resource.
- if there is a description of the type via its URI/URL (e.g., by an OWL ontology), additional "typical" properties of the resource become known (inheritance).

Note: Typing means "semantic" typing in the RDF graph (corresponding to the notion of "classes"), not in a given XML representation (that would use XML Schema *complex types*).

- Aside: software engineering/programming languages distinguish between the notion of "class" (more semantic) and "type" (more detailed, syntactic notion)

## RDF Datatypes

- classes: complex, application-specific notion
- literal datatypes: basic, common notion

RDFS does not define its own datatypes ... it uses those from XMLSchema:

- Range and domain of <rdf:Property> can refer to XML Schema types, e.g.

  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>

```
<rdf:Description rdf:ID="D">
   <rdf:type rdf:resource="dbis.uni-goe.de/Mondial/mondial.rdfs#country"/>
   <mondial:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
         Germany</mondial:name>
   <mondial:area rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
         356910</mondial:area>
</rdf:Description>
```

(note: `http://www.w3.org/2001/XMLSchema#` can be defined as an entity in the local DTD to the RDF/RDFS instance and then used as rdf:datatype="&xsd;string")



### Example: Typing

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:mon="http://www.mondial.de/mondial.rdfs#">
  <rdf:Description rdf:about="mon://country/de"
    mon:name="Germany" mon:code="D">
    <rdf:type rdf:resource="http://www.mondial.de/mondial.rdfs#country"/>
    <mon:has_province>
      <rdf:Description rdf:about="mon://country/de/prov/niedersachsen">
        <rdf:type rdf:resource="http://www.mondial.de/mondial.rdfs#province"/>
        <mon:name>Niedersachsen</mon:name>
        <mon:has_city>
          <rdf:Description rdf:about="mon://country/de/prov/niedersachsen/city/hannover"
               mon:name="Hannover">
            <rdf:type rdf:resource="http://www.mondial.de/mondial.rdfs#city"/>
            <mon:in_country rdf:resource="mon://country/de"/>
            <mon:population rdf:parseType="Resource">
              <rdf:type rdf:resource="http://www.mondial.de/mondial.rdfs#measure"/>
              <mon:year rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1995</mon:year>
              <mon:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">525763</mon:value>
            </mon:population>
          </rdf:Description>
        </mon:has_city>
        <mon:has_capital rdf:resource="mon://country/de/prov/niedersachsen/city/hannover"/>
      </rdf:Description>
    </mon:has_province>
  </rdf:Description>
</rdf:RDF>
```

## COMPONENTS OF AN RDF INSTANCE

- application-oriented data: structured by (subject, predicate, object) tripels, values
- application-oriented metadata: information about types/classes, names of properties
- RDF "infrastructure"



Example: Components of an RDF instance

```
<rdf:RDF xmlns:rdf="..." xmlns:mon="http://www.mondial.de/mondial.rdfs#">
 <rdf:Description rdf:ID="de" mon:name="Germany" mon:code="D">
  <rdf:type rdf:resource="http://www.mondial.de/mondial.rdfs#country"/>
  <mon:has_province>
   <rdf:Description rdf:ID="de-ndrs" mon:name="Niedersachsen">
    <rdf:type rdf:resource="http://www.mondial.de/mondial.rdfs#province"/>
    <mon:has_city>
     <rdf:Description rdf:ID="de-hann" mon:name="Hannover">
      <rdf:type rdf:resource="http://www.mondial.de/mondial.rdfs#city"/>
      <mon:in_country rdf:resource="#de"/>
      <mon:population rdf:parseType="Resource">
       <rdf:type rdf:resource="http://www.mondial.de/mondial.rdfs#measure"/>
       <mon:year rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1995</mon:year>
       <mon:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">525763</mon:value>
      </mon:population>
     </rdf:Description>
    </mon:has_city>
    <mon:has_capital rdf:resource="#de-hann"/>
   </rdf:Description>
  </mon:has_province> </rdf:Description>  </rdf:RDF>
```

## Relationship to XPointer / URI vs. IDREF

Consider the following fragment from the previous example:

```
<rdf:Description rdf:ID="de" mon:name="Germany" mon:code="D">
  <mon:has_province>
    <rdf:Description rdf:ID="de-nds" mon:name="Niedersachsen">
      <mon:has_city>
        <rdf:Description rdf:ID="de-hann" mon:name="Hannover">
          <mon:in_country rdf:resource="#de"/>
        </rdf:Description>
      </mon:has_city>
      <mon:has_capital rdf:resource="#de-hann"/>
    </rdf:Description>
  </mon:has_province>
</rdf:Description>
```

- rdf:resource="#de-hann" is a reference to a local extension of the "base uri" (which is e.g. the uri of the RDF file): the "local" part "de-hann" is extended to "*base-uri*#de-hann", and thus points to the element with rdf:ID="de-hann".
  (note that for the XML/DTD IDREF mechanism, the "#" in the reference attribute would be missing)



## World Wide RDF Web

- many such information sources that describe resources
- higher level information management (e.g., portals): use some of these sources for accessing *integrated* information

Example (RDF source see next slide) – the example is not based on real data

- mondial: countries, cities
- `http://sw.mondial.de/mondial.rdfs`: meta information about geography ontology
- another resource: cities and their airports
- `http://sw.iata.org/onto.rdfs` (International Air Transport Assoc.): ontology about flight information
- `bla:sw.iata.org/flights/`*flight*: resource associated with a given flight (e.g. LH42).
- `bla:sw.iata.org/airports/`*abbrev*: resource associated with a given airport (e.g., FRA, CDG).
- there will probably be a Lufthansa RDF database that describes the flights in their terminology

## Example (Cont'd) [Filename: RDF/flightbase.rdf]

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <rdf:Description rdf:ID="de"
     xmlns:mon="http://www.mondial.de/onto.rdfs#">
    <rdf:type rdf:resource="http://www.mondial.de#country"/>
    <mon:name>Germany</mon:name>
    <mon:has_population rdf:parseType="Resource">
      <rdf:type rdf:resource="http://www.mondial.de#measure"/>
      <mon:year>1995</mon:year>  <mon:value>83536115</mon:value>
    </mon:has_population>
    <mon:has_city>
      <rdf:Description rdf:about="http://www.berlin.de">
        <rdf:type rdf:resource="http://www.mondial.de#city"/>
        <mon:name>Berlin</mon:name>
      </rdf:Description>
    </mon:has_city>
    <mon:has_capital rdf:resource="http://www.berlin.de"/>
    <mon:has_neighbor rdf:resource="#fr"/>
  </rdf:Description>

  <rdf:Description rdf:ID="fr"
      xmlns:mon="http://www.mondial.de/onto.rdfs#">
    <rdf:type rdf:resource="http://www.mondial.de#country"/>
    <mon:name>France</mon:name>
    <mon:has_population rdf:parseType="Resource">
      <rdf:type rdf:resource="http://www.mondial.de#measure"/>
      <mon:year>1995</mon:year>  <mon:value>58317450</mon:value>
    </mon:has_population>
    <mon:has_city>
      <rdf:Description rdf:about="http://www.paris.fr">
        <rdf:type rdf:resource="http://www.mondial.de#city"/>
        <mon:name>Paris</mon:name>
      </rdf:Description>
    </mon:has_city>
    <mon:has_capital rdf:resource="http://www.paris.fr"/>
    <mon:has_neighbor rdf:resource="#de"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.berlin.de"
       xmlns:touri="http://www.world-tourism.org/onto.rdfs#">
    <touri:has_airport rdf:resource="bla:sw.iata.org/airports/BLN"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.paris.fr"
       xmlns:touri="http://www.world-tourism.org/onto.rdfs#">
    <touri:has_airport rdf:resource="bla:sw.iata.org/airports/CDG"/>
  </rdf:Description>

  <rdf:Description rdf:about="bla:sw.iata.org/flights/LH42"
       xmlns:iata="http://sw.iata.org/onto.rdfs#">
    <rdf:type rdf:resource="http:sw.iata.org/onto.rdfs#flight"/>
    <iata:from rdf:resource="bla:sw.iata.org/airports/BLN"/>
    <iata:to rdf:resource="bla:sw.iata.org/airports/CDG"/>
  </rdf:Description>
</rdf:RDF>
```



## RDF vs. XML

Everything that can be expressed by XML can also be expressed by RDF

(Exercise: create an RDF mondial instance with local resources)

+ RDF can also be used to *describe* resources
  (pictures, films, ..., programs, Web services, ...)
+ RDF can be represented as a graph, independent from the structure of the (distributed) RDF instances
+ RDF has a connection to global schema description mechanisms

o RDF can be queried in the same way by XPath/XQuery ...

- the query expressions are much less intuitive
  – if local resources are used: relatively easy
  – if global resources are used: appropriate RDFs must be searched for.

? High-level metadata query languages (proposal: e.g. RQL) ...

... what is behind the ".rdfs" urls used above for namespace declarations?

## 13.2 RDF Schema

- RDF ist the instance level
- XML: DTDs and XML Schema for describing the structure/schema of the instance
- RDF Schema: stronger than DTD/XML – "semantic-level"
  - describe the structure of the RDF instance (i.e. the "schema" of the RDF graph, not of the XML-RDF file)
    - "rdf:" elements part described by RDF DTD
    - names of property subelements/attributes follow from an (application-specific) RDFS instance
      (that is "somehow" coupled with the namespace declaration)
  - describes the schema in terms of an (lightweight) ontology (OWL provides then much more features)
    - *notions* are described and related *semantically*:
      - class/subclass
      - property/subproperty
- classes and properties are also resources, described by RDFS instances



## RDF Schema Documents

Similar to RDF/XML documents

- additionally include rdfs namespace declaration

  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

- description of classes

  <rdfs:Class rdf:about="*uri*"> <!-- used by <rdf:type resource="*uri*">-->
  <rdfs:subClassOf rdf:resource="*uri or #local-id*">
  </rdfs:Class>

- description of properties

  <rdfs:Property rdf:about="*uri*"> <!-- used by <*namespace:name*>-->
  <rdfs:subPropertyOf rdf:resource="*uri*"/>
  <rdfs:domain rdf:resource="*uri*"/>
  <rdfs:range rdf:resource="*uri*"/>
  </rdfs:Property>

- instead of @rdf:about="*uri*", also @rdf:ID="*id*" or @rdf:nodeID="*local-id*" can be used.
- same for @rdf:resource="*uri*": @rdf:resource="*#name*" or @rdf:nodeID="*local-id*"

## Defining URIs of RDFS Classes and Properties

Classes and properties are “usual” resources, identified/defined by

<rdfs:Class rdf:about=“*uri*”> ... </rdfs:Class>
reference by rdf:resource=“*uri*”

<rdfs:Class rdf:ID=“*classname*”> ... </rdfs:Class> (+ base-uri)
reference by rdf:resource=“*#classname*” (local)
reference by rdf:resource=“*base-uri#classname*” (from remote)

<rdfs:Class rdf:nodeID=“*classname*”> ... </rdfs:Class>
reference by rdf:nodeID=“*classname*” (only for local definitions)

- references to classes: see above
- references to properties:
  - from the RDF instance: by property names (including namespaces)
    <*namespace:property*> – expanding the xmlns:*namespace*=“*uri*” declaration
  - from other RDFS elements (e.g. in rdf:subproperty): similar to references to classes (@rdf:resource or @rdf:nodeID)



## Version A: “Real” Resources

### Classes

Anything that is defined in an RDFS document - e.g., in

http://www.mondial.de/mondial.rdfs#

as an

<rdfs:Class rdf:ID=“country”>
<!-- SubclassOf-defs etc.-->
</rdfs:Class>

can be used in another RDF document as an rdf:type, e.g.

<rdf:Description rdf:ID=“country-D”>
mondial:name=“Germany” mondial:carcode= “D”>
<rdf:type rdf:resource=“http://www.mondial.de/Mondial/mondial.rdfs#country”/>
</rdf:Description>

## Properties

Consider RDF triples:

- subject, object: identified by a URI or a local ID
- predicate: of the form *namespace:property*, e.g., mondial:capital

## Using Namespaces

Namespaces are bound to a URI by the namespace declaration

xmlns:mondial="http://dbis.uni-goe.de/Mondial/mondial.rdfs#"

extending <mondial:capital> yields the URL (with local part)

http://dbis.uni-goe.de/Mondial/mondial.rdfs#capital

that addresses

```
<rdfs:Property rdf:ID="capital">
  <rdfs:domain rdf:resource="#political_entity"/>
    <!-- i.e., country, province, district, county -->
  <rdfs:range rdf:resource="#city"/>
</rdfs:Property>
```



# VERSION B: "VIRTUAL" RESOURCES

## Classes

- straightforward by <rdfs:Class rdf:about="*uri*"> and <rdf:type rdf:resource="*uri*"/>

## Properties

- the same way: the whole vocabulary is "identified" with a base URI, e.g. www.mondial.de (where no special RDFS instance is located)
- the RDFS document (located somewhere) uses rdf:about= "*base-uri/property-name*"
- the namespace is declared as xmlns:*namespace*="*base-uri/*"
- and *namespace:property* refers to *base-uri/property-name*

## COMPARISON

- Version A: class/property resources are fragments of the RDFS instance:
  + @rdf:resource can actually be dereferenced and yields the class/property definition
  - only practical if the RDFS is non-distributed
    (although remote RDFS instances can also describe this resource by using rdfs:about)

  ⇒ centralized ontologies
- Version B: class/property resources are identified by a virtual URI
  + arbitrary RDFS instances can contribute to the resource description
  - users/clients have to know where the resource descriptions can be found

  ⇒ distributed ontologies (only a central/common schema for class/property URIs required)



## RDFS IS ITSELF AN RDF APPLICATION

Note: rdfs:Class is a shorthand for RDF where classes are defined as instances of the RDFS class “Class”:

```
<rdf:Description id="country">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:Subclass ... >
</rdf:Description>
```

- analogously for `http://www.w3.org/2000/01/rdf-schema#Property`
- analogously for rdfs:range and rdfs:domain as properties whose domain is a property and whose range is the set of classes.
- a class can be a subclass of itself (allows for defining equivalence)

⇒ RDF model theory (allows for formal reasoning)
related to *Description Logics*

## USAGE OF RDF DATATYPES IN RDFS

- usually, the datatype information as described on Slide 457 is not provided in the RDF instance, but in the RDFS Schema:
- Range and domain of <rdf:Property> can refer to XML Schema types, e.g.
  <rdf:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
- Literal values can refer to their XML Schema datatypes as e.g.

```
<rdfs:Property rdf:ID="population">
   <rdfs:domain rdf:resource="#political_entity"/>
      <!-- i.e., country, province, district, county -->
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
</rdfs:Property>
```



## EXAMPLE: DUBLIN CORE

"Dublin Core": a set of properties for describing documents (defined at a metadata workshop in Dublin/Ohio 1995).

- Title: A name given to the resource.
- Creator: An entity primarily responsible for making the content of the resource.
- Subject: The topic of the content of the resource.
- Description: An account of the content of the resource.
- Publisher: An entity responsible for making the resource available
- Contributor: An entity responsible for making contributions to the content of the resource.
- Date
- Type: The nature or genre of the content of the resource.
- Language: A language of the intellectual content of the resource.

etc ... see `http://dublincore.org/documents/dces/`

## Using RDF in the World Wide Web

- The (Semantic) Web is not seen as a collection of documents, but as a collection of correlated information (described via documents)
- using RDF, everybody can make statements about any resource (cf. link-bases in XLink)
  - incremental, world wide data and meta-data
- not assumed that complete information about any resource is available

Reasoning based on RDF

- RDF/RDFS define a *model theory*
- incomplete knowledge when reasoning ("open world assumption" – monotonic or nonmonotonic reasoning)
- potentially even inconsistent
- statements can be equipped with probabilities or labeled as opinions fuzzy reasoning, belief revision ...

... lots of artificial intelligence applications



## 13.3 Web Ontology Language – OWL

- *vocabularies* can be defined by RDFS
  - not so much stronger than the ER Model
  - not only conceptual model, but a "real language"
  - *incremental* world-wide approach
  - "global" vocabulary can be defined by autonomous partners
- but: no way to *describe* the vocabulary

Ontologies/ontology languages further extend the expressiveness

- Topic Maps (in SGML) since early 90s, XTM (XML Topic Maps)
- Ontolingua – non-XML approach from the Knowledge Representation area
- OIL (Ontology Inference Layer): initiative funded by the EU programme for Information Society Technologies (project: On-To-Knowledge, 1.2000-10.2002); based on RDF/RDFS
- DAML (Darpa Agent Markup Language; 2000) ... first ideas for a Semantic Web language
- DAML+OIL (Jan. 2001)
- developed into OWL (1st version March 02, finalized Feb. 04)

## THREE VARIANTS OF OWL

Several expressiveness/complexity/decidability levels:

- OWL Full: extension of RDF
  - classes can also be regarded as individuals (classes of classes ... higher-order reasoning)
- OWL DL
  - fragment of OWL that fits into the Description Logics Framework
  - decidable reasoning
- OWL Lite
  - subset of OWL DL
  - easier migration from frame-based tools
    (note: F-Logic was a frame-based framework)
  - easier reasoning



## OWL INSTANCE STRUCTURE

extend RDF (and collect namespaces ...)

```
<rdf:RDF xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 <!-- now the Ontology header follows -->
 <owl:Ontology
      rdf:about="... base-uri for subsequent stuff ..."/>
 <!-- and then the ontology itself -->
 :
</rdf:RDF">
```

- the <owl:Ontology> element can contain additional information about the version, previous versions etc.

Content of the OWL instance: RDF/RDFS with additional OWL information

- RDFS class definitions
- RDFS property definitions

## CLASSES

Class definition: name introduction or reference + defining statements:

- <owl:Class rdf:ID="*new_name*"> ... </owl:Class> defines a new class (that can then be referenced by the URI *currentdocument#new_name*)
- <owl:Class rdf:about="*uri*"> ... </owl:Class> adds information about a class that is defined elsewhere

The elements in the contents of the owl:Class element then describe restrictions that must (all) be satisfied.



### Expressing Information about Classes

- <rdfs:subClassOf> (with @resource or contents)
- <owl:equivalentClass> or <owl:sameAs> (sameAs derived as subPropertyOf from subClassOf)
- <owl:unionOf>, <owl:intersectionOf>
- <owl:complementOf>, <owl:disjointWith>
- <owl:oneOf> (enumeration of elements)

### Anonymous classes (as classes in the above elements)

Elements <owl:Restriction> ... </owl:Restriction> define *anonymous* classes that satisfy the given restrictions
(owl:Restriction is a subclass of owl:Class):

- <owl:onProperty rdf:resource="..."> ... </owl:onProperty>
  - <owl:hasValue>, <owl:allValuesFrom>, <owl:someValuesFrom>,
  - <owl:cardinality>, <owl:minCardinality> or <owl:maxCardinality>
  - including @rdf:datatype restrictions for the range

## PROPERTIES

derived from <rdf:Property>:

- <rdfs:ObjectProperty> – subclass of rdf:Property; object-valued (i.e. rdfs:range must be a Object class)
- <rdfs:DatatypeProperty> – subclass of rdf:Property; datatype-valued (i.e. rdfs:range must be an xsd:datatype)

Properties can then be specified in more detail:

- <rdfs:subPropertyOf rdf:resource="..."/>
- <owl:constraintProperty>
- <owl:equivalentProperty>
- <owl:inverseOf rdf:resource="...">
- <owl:TransitiveProperty> , <owl:SymmetricProperty>, <owl:FunctionalProperty> , <owl:InverseFunctionalProperty>
  (often as <rdf:Type rdf:resource="&owl;..."/>)



## OBJECTS

- used in enumerations (<owl:oneOf>)
- used to describe actual instances
- <owl:Thing>
  - <owl:sameIndividualAs> – states that 2 resources are the same thing
  - <owl:differentFrom> – forbids deriving equivalence

```
<owl:Thing rdf:ID="Germany">
  <rdf:Type rdf:resource="#Country">
  <owl:sameIndividualAs ...>
  <owl:differentFrom resource="#France">
</owl:Thing>
```

equivalent short form:

```
<Country rdf:ID="Germany"> ... </Country>
```

## SEMANTIC WEB: XML; RDF AND OWL

In contrast to XPath/XQuery, XSLT, XML Schema, XLink etc., RDF and OWL are *not* languages *"inside"* the XML world, but are concepts of their own that have - incidentally- also an XML syntax.

The combination of XML data and RDF/RDFS/OWL concepts is the base for the *Semantic Web*.

A Semantic Web application e.g. exists of

- a "central" portal that uses the following things:
- a set of ontological (OWL, RDFS) sources,
- a set of RDF sources,
- reasoning (using OWL and RDFS information),
- a semantical description of itself for allowing others to use it.

### Literature

`http://www.w3.org/2004/OWL/`: OWL Use Cases and Requirements, OWL Language Overview, OWL Language Guide ...