Knowledge management
RDF and RDFS
Outline

• RDF
  – Semantic Web and Metadata
  – What is RDF and what is not?
  – Why use RDF?
  – RDF Elements
• RDF Schema (RDFS)
• RDF Storing
Semantic Web

• The Web today: Documents for humans.

• Problem: hard to machine-process on a semantic level.

• Vision: Make the information in the Web machine-processable, for intelligent services, better user interaction and autonomous agents

• Realization idea: Semantic annotation of objects + query and reasoning mechanisms

• Requirement: machine-processable languages for annotation and representation reasoning tools and a naming mechanism
Metadata

• Metadata describes other data ("Data about Data")
  – One application’s metadata is another application’s data
  – Metadata can itself be described by metadata (but that doesn’t make it meta-metadata)

• Metadata is useful
  – e.g. A lot could be gained from structured description of pages, servers, search services, and other resources

• Example(s):
  – a library catalogue contains information (metadata) about publications (data)
  – a file system maintains permissions (metadata) about files (data)
**Resource Description Framework (RDF)**

- **RDF** is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata model but which has come to be used as a general method of modeling information, through a variety of syntax formats.

  *Source: Wikipedia*

RDF is a standard syntax to represent (edge labeled) directed graphs in XML
Why use RDF?

- Improve on HTML and XML
- Machine *understandable* metadata
- Support structured values
- Base for a variety of descriptions: cataloging, privacy, accessibility, IPR, ...
- Store and query metadata
What is RDF?

• An abstract formalism
• A directed graph data model
• A set of binary statements ("triples")
  – Subject Predicate Object

→ RDF can be used to encode ontologies
What is RDF not?

- A relational database
- A (database) management system
- A query language
- A file
- A new version of HTML or XML
RDF Elements

- Resources
- Properties
- Literal values
- Statements (Assertions)
  - Resource Property Resource
  - Resource Property Literal
- Namespaces (disambiguation of homonym identifiers)
RDF Resources

• Almost everything is a resource
• RDF stores statements about resources:
  – Tangible things of the real world
  – Electronic objects
  – Abstract ideas such as classes/topics/…
• Resources are identified by URIs

URIs are rigid designators in a global domain
RDF Description

• Attributes
  – *about*
  • refers to a URI of an existing resource
  – *ID*
  • signals the creation of a new resource

```xml
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
      .
      .
      .
    </rdf:Description>
    <rdf:Description rdf:ID="myID">
      .
      .
      .
    </rdf:Description>
  </rdf:RDF>
```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage>
        <rdf:Description rdf:about="http://purl.org/net/dajobe/">
          </rdf:Description>
      </ex:homePage>
    </rdf:Description>
  </ex:editor>
</rdf:Description>
RDF properties

• Properties
  – Properties: special kind of resources
  – A specific aspect, characteristic, attribute, or relation used to describe a resource or binary relations between two resources
  – E.g., “WorkFor”, “hasAuthor”, “father-of”
RDF properties

- Property names must be associated with a schema
- Qualify property names with a namespace prefix

Example:

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
>
    <rdf:Description about="http://www.www.fr/">
        <dc:creator>myID</dc:creator>
    </rdf:Description>
</rdf:RDF>
```
RDF literals

- Literals
  - Concrete data values
  - E.g. “John Smith”, “1”, “2005-03-07”
  - Literal values are data
  - Untyped literals are just strings
  - Typed literals borrow from XML Schema Datatypes: String, date, float ...
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <dc:title>"RDF Syntax Specification"</dc:title>
  <ex:editor>
    <rdf:Description>
      <ex:fullName>“Joe Blog”</ex:fullName>
      <ex:homePage rdf:resource="http://purl.org/net/dajobe/"/>
    </rdf:Description>
  </ex:editor>
</rdf:Description>
RDF statements (assertions)

- Statements
  - Different name: Assertion = Triple = Statement
  - Express facts about resources
  - A statement contains three parts: subject, predicate, and object
  - A set of assertions creates a graph
  - A graph cannot contain only resources
  - Example of statements:
    - http://www.w3.org/ has the format text/html
    - http://www.debruijn.net/ has first name Jos
    - http://www.polleres.net/page.html is the Web page of http://www.polleres.net/axel
RDF Syntax Example

**Exercise**: Use RDF to describe these resources:
- A bank named KMB
- KMB has client John
RDF Containers

• Bag
  – An unordered list of resources or literals
• Sequence
  – An ordered list of resources or literals
• Alternative
  – A list of resources or literals that represent alternatives for the value of a property
Using the Bag Container

• **Statement:**
  - The authors of the book 0201000237 are Alfred, John and Jeffrey

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <rdf:Description about="urn:ISBN:0-201-00023-7">
        <dc:creator>
            <rdf:Bag>
                <rdf:li>Alfred</rdf:li>
                <rdf:li>John</rdf:li>
                <rdf:li>Jeffrey</rdf:li>
            </rdf:Bag>
        </dc:creator>
    </rdf:Description>
</rdf:RDF>
```
Using the Seq Container

• **Statement:**
  The students of the course km in alphabetical order are Elizabeth, George and John

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:s="http://www.schemas.org/Course/">
  <rdf:Description about="http://www.www.fr/courses/km">
    <s:students>
      <rdf:Seq>
      </rdf:Seq>
    </s:students>
  </rdf:Description>
</rdf:RDF>
```
Using the Alt Container

**Statement:**
- The formats of the book 0201000237 are plain text, html and postscript

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <rdf:Description about="urn:ISBN:0-201-00023-7">
        <dc:format>
            <rdf:Alt>
                <rdf:li>text/html</rdf:li>
                <rdf:li>text/plain</rdf:li>
                <rdf:li>application/postscript</rdf:li>
            </rdf:Alt>
        </dc:format>
    </rdf:Description>
</rdf:RDF>
```
RDF Namespaces

- Namespaces are used to separate vocabularies
- A namespace is defined by a URI
- Example:
  - A book has a name and an author also has a name
Ontological Reasoning in RDF

Type constraint violation: The range of owns is Fish.

OR  There is no inconsistency: Wanda is a fish!
RDFS

- RDF provides a data model to define relations between resources (on the Web).
- A framework for defining meta data for Web resources
- However the triple data model is insufficient without sharing the same (knowledge) vocabularies (+semantics)
- RDF-S allows to define RDF vocabularies
- RDF-S allows to define class hierarchies, property hierarchies, and allowed relations (triples)
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#">

<owl:Ontology rdf:about="http://www.w3.org/2000/01/rdf-schema#"/>

<rdfs:Class rdf:about="http://www.w3.org/2000/01/rdf-schema#Resource">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-schema#"/>
  <rdfs:label>Resource</rdfs:label>
  <rdfs:comment>The class resource example</rdfs:comment>
</rdfs:Class>
</rdf:RDF>
RDFS

• RDFS is a vocabulary to create vocabularies (!)
  – properties defined by a particular community
  – characteristics of properties and/or constraints on corresponding values
• Comparable to XML Schema or XML DTD
• Used to standardize which tags the creator of a graph is allowed to use for annotating resources
• Introduces notions such as: Property, Class, SubClassOf, Domain, Range
RDF Schema (cont’d)

• Defines the types of relations a resource of a certain type may have

• **Important:** Compatibility check of a graph to a schema is **not** automatically performed upon parsing
  
  ➔ RDF triples that are inconsistent can be added to a graph and are not detected unless a consistency check is performed
RDFS Constructs

- **X rdf:type rdfs:class**
  - Declares the resource X as a class for other resources

- **R rdf:type rdf:Property**
  - Declares resource R as a property

- **R rdfs:domain X**
  - Declares the subject of R as an X

- **R rdfs:range Y**
  - Declares the object of R as an Y
RDFS extends the RDF vocabulary:
RDFS vocabulary is defined in the namespace:
http://www.w3.org/2000/01/rdf-schema# (associated with namespace prefix rdfs:)

- **RDFS Classes:**
  - rdfs:Resource
  - rdfs:Class
  - rdfs:Literal
  - rdfs:Datatype
  - rdfs:Container
  - rdfs:ContainerMembership Property

- **RDFS properties**
  - rdfs:domain
  - rdfs:range
  - rdfs:subPropertyOf
  - rdfs:subClassOf
  - rdfs:member
  - rdfs:seeAlso
  - rdfs:isDefinedBy
  - rdfs:comment
  - rdfs:label
Figure taken from [de Bruijn et al., 2004]
RDFS meta-data

- Any meta-data can be attached to a resource, using:
  - rdfs:comment
    - Human-readable description of the resource, e.g.
      `<ex:Person> rdfs:comment “A person is any human being”`
  - rdfs:label
    - Human-readable version of the resource name, e.g.
      `<ex:Person> rdfs:label “Human being”`
  - rdfs:seeAlso
    - Indicate additional information about the resource, e.g.
      `<ex:Person> rdfs:seeAlso <http://xmlns.com/wordnet/1.6/Human>`
RDF-S: example – RDFS ontology about Persons and Universities

ex:Institution rdf:type rdfs:Class
ex:Person rdf:type rdfs:Class
ex:Research-institution rdfs:subClassOf ex:Institution
ex:Educational-institution rdfs:subClassOf ex:Institution
ex:University rdfs:subClassOf ex:Research-institution
ex:University rdfs:subClassOf ex:Educational-institution
ex:Faculty rdfs:subClassOf ex:Person
ex:Student rdfs:subClassOf ex:Person
ex:Phd-Student rdfs:subClassOf ex:Faculty
ex:Phd-Student rdfs:subClassOf ex:Student
ex:Msc-Student rdfs:subClassOf ex:Student
ex:Msc-Student rdfs:comment "An Msc student is not a faculty"
ex:Phd-Student rdfs:comment "A student who is also a faculty is a PhD Student"
ex:employed-by rdf:type rdfs:Property
ex:manager rdf:type rdfs:Property
ex:employed-by rdfs:domain ex:Person
ex:employed-by rdfs:range ex:Institution
ex:manager-of rdfs:domain ex:Person
ex:manager-of rdfs:range ex:Person
Storing RDF

• RDF graphs can be serialized as and stored in the file system

• For more DBMS-like applications, there are RDF repositories that provide
  – Query functionality
  – Access control
  – Distribution

• Examples:
  – Sesame
  – 3-Store
  – JENA
  – YARS
Outline

• Ontologies
• RDF
• RDF Schema (RDFS)
• Outlook at RDF Storing
References

- Semantic Web: www.w3.org/2001/sw
- Ontologies: www.ontology.org
- RDF and RDFS: www.w3.org/rdf
- RDF syntax: http://www.w3.org/TR/rdf-syntax-grammar/
- RDFS: http://www.w3.org/TR/rdf-schema/