Chapter II

Semantic Web

The Semantic Web

- is a concept by Tim Berners-Lee to extend the world wibe web with machine-readable data describing the semantics of the content
- the idea is to extend human-readable data with additional information (meta data) that can be interpreted by a machine, so inquiries can be processed according to their meaning, rather than their spelling
- to create these additional data and to implement a semantic web, several methods can be used
  - taxonomies
  - thesauri
  - resource description frameworks
  - ontologies
  - topic maps

Semantic Web – Examples (I)

- FOAF – Friend of a Friend
  - is a project to model a machine-readable social network
  - a FOAF document contains information about a person (such as name, age, gender, addresses (website, weblog, home…), messenger-id…) and other people this person knows
  - the documents then refer to each other and can be analysed by a software, which is then able to visualise the details and the social structure
  - FOAF uses RDF
  - the project: http://www.foaf-project.org/
  - the specification: http://xmlns.com/foaf/0.1/

Semantic Web – Examples (II)

- DOAC – Description of a Career
  - is a project to describe the curriculum of a person in machine-readable form
  - a DOAC document contains information about the skills of a person (such as education, spoken languages, experiences, previous jobs, drivers license
  - it is compatible with the European Unino Europass Curriculum (see http://europass.cedefop.europa.eu/)
  - DOAC uses RDF
  - project: http://ramonantonio.net/doac/
  - specification: http://ramonantonio.net/doac/0.1/

Semantic Web – Examples (III)

- SemanticGov
  - is a project of an international consortium, sponsored by the European Union, to improve the administration of the EU (e.g. the co-operation between authorities)
  - the project duration is 36 months (01.01.2006 – 31.12. 2008), the budget is around 4.37 billion euros
  - project: http://www.semantic-gov.org/

Meta Data

- data containing information about other data (the latter mostly a larger amount of data)
- are used to describe information resources
  - to improve their discovering
  - to document their mutual relations
- are saved
  - within the document (as a meta tag)
  - in assigned reference books (such as catalogs)
  - as an attribute which is held together with the document

Classification

- is a method to divide objects into categories or classes
- the division bases on the moulding of the objects common properties
- can be done
  - manually (categorising, sorting, indexing)
  - automatically (supervised learning)
- classification systems (divided by structure)

	mono-hierarchical	**poly-hierarchical**
heredity	single (strong hierarchy)	multiple (weak hierarchy)
superclasses	one	more than one
structure	tree	non-cyclic directed graph

Taxonomies

- are classification systems with mono-hierarchical structures
- to create a simple semantic
- the root contains general information
- when navigating through the tree structure from the root element on the information gets more and more specific

Thesauri

- are classification systemes with poly-hierarchical structures
- are systematically ordered, networked collections of terms (a so-called "controlled vocabulary", that means a list of terms with an unambiguous, non-redundant definition that have been enumerated explicitly), connected via associative and parent-child relationships
- used to describe/represent topics, for subject indexing and/or document retrieval
- examples:
  - OpenThesaurus: http://www.openthesaurus.de/
  - UNESCO thesaurus: http://databases.unesco.org/thesaurus/

RDF (I)

- the Resource Description Framework (RDF) was developed as a foundation stone for the semantic web by the W3C in 1999
- it is a formal language for the provision of meta data via the WWW and defines a fundamental vocabulary to formulate arbitrary statements about resources
- a statement is a triple consisting of (in this order)
  - a subject
  - a predicate (property)
  - an object
  - -> a person (*subject*) has (*predicate*) a name (*object*)
  - all elements of a triple (the *resources*) are identified by a URI
  - an RDF document is a collection of linked statements

RDF (II)

- the predicate tells something about the subject
- the object is the value of the predicate; it can be a resource (identified by a URI) or just a literal (a constant value)
- predicates and objects can be subjects in different statements
- resources can be used for grouping, they don't have a name then (i.e. a name can be grouped in name and first name)
- statements can be subjects in another statement -> this is called a reification

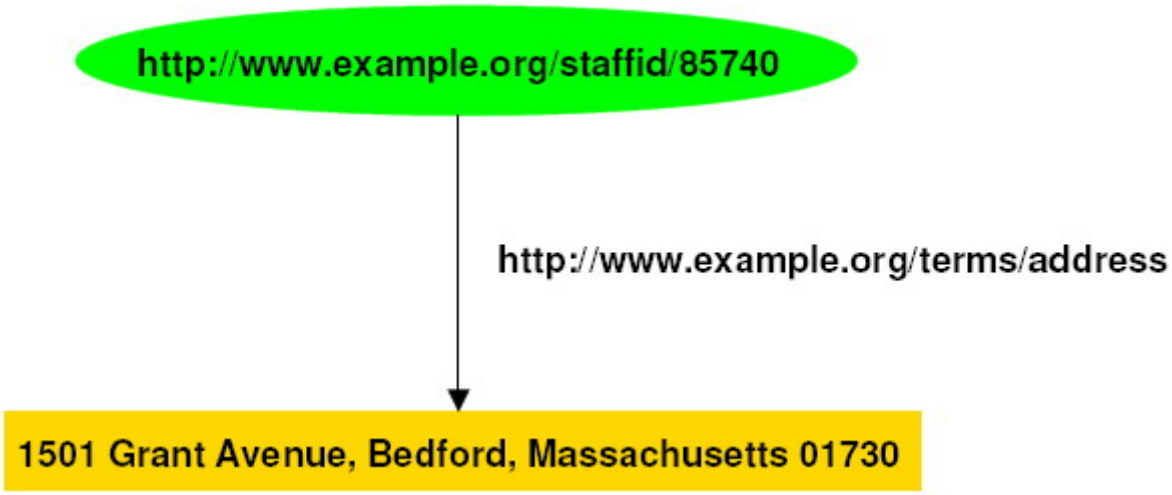
RDF Graph

- is the standard development method for RDF
- is a labeled, directed, easy readable graph
- modeling:
  - subjects are modeled as a node in form of an ellipse
  - predicates are modeled as an arc
  - objects represented by a URI are modeled as a node in form of an ellipse
  - objects represented by a literal are modeled as a node in form of a box
  - resources used for grouping are modeled as blank nodes (these cannot be referenced)
- SPARQL (SPARQL Protocol and RDF Query Language) is a W3C query language recommendation for RDF graphs
  - http://www.w3.org/TR/rdf-sparql-query/

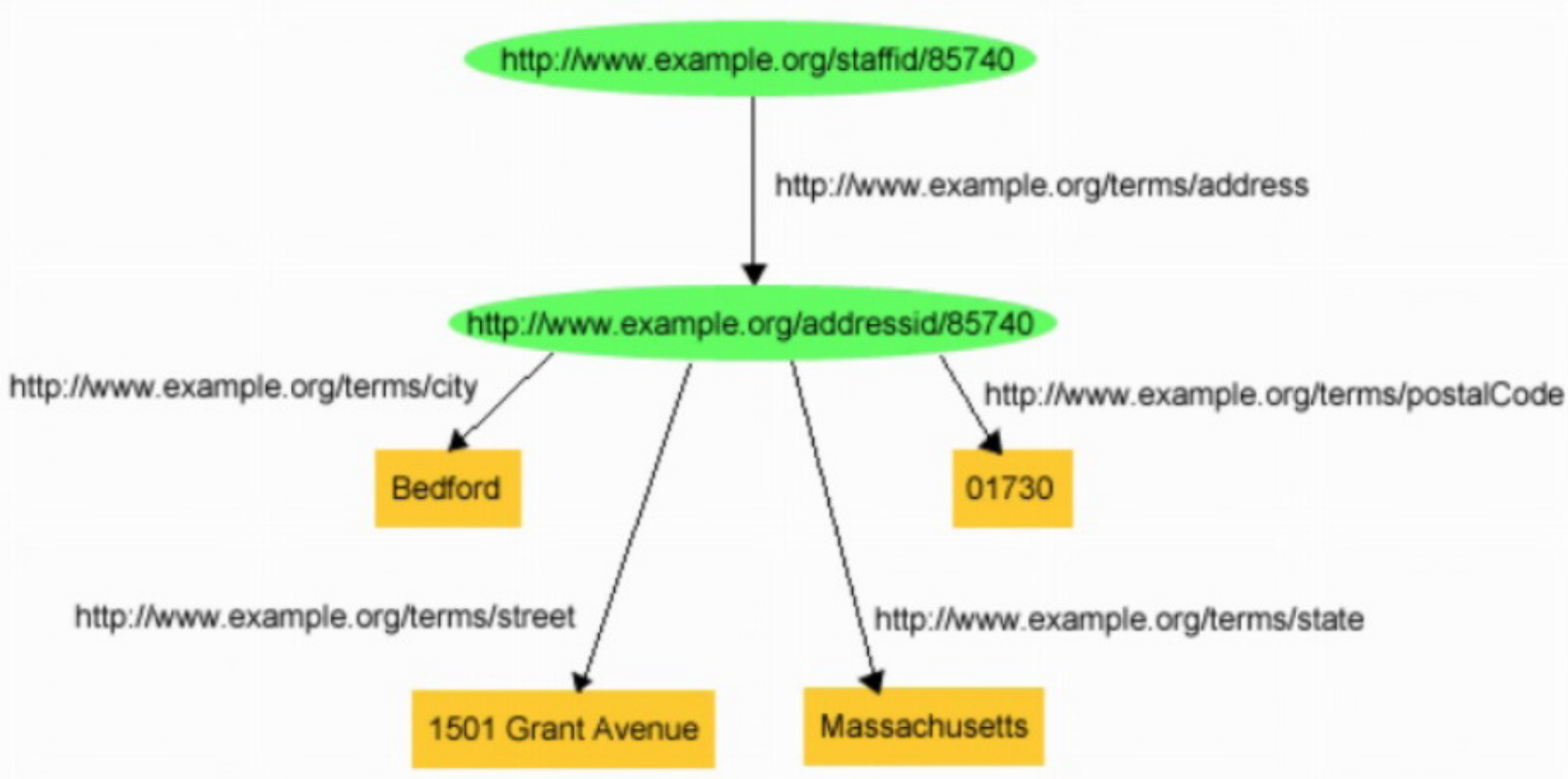
RDF Graph - Example

- there is a person identified by http://www.w3.org/people/em/contact#me, whose name is Eric Miller, whose email address is em@w3.org, and whose title is Dr.

Blank Nodes – Example (I)



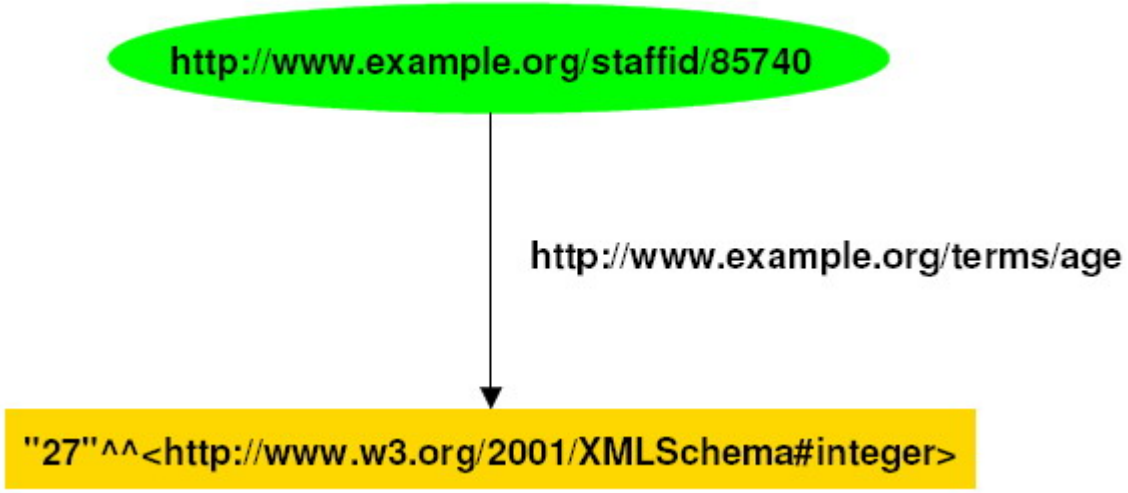
Blank Nodes – Example (II)



Blank Nodes – Example (III)

```
exstaff:85740        exterms:address       exaddressid:85740 .
exaddressid:85740    exterms:street        "1501 Grant Avenue" .
exaddressid:85740    exterms:city          "Bedford" .
exaddressid:85740    exterms:state         "Massachusetts" .
exaddressid:85740    exterms:postalCode    "01730" .
```

Typed Literals




```
exstaff:85740        exterms:age        "27"^^xsd:integer
```

RDF Syntax

- to implement RDF models two different syntaxes exist
  - N3 (Notation 3) by Tim Berners-Lee
    - http://www.w3.org/DesignIssues/Notation3.html
  - an XML based syntax which is the most used
    - http://www.w3.org/TR/rdf-syntax-grammar/

RDF Serialisation

- an RDF graph is encoded as XML elements, attributes, element content and attribute values

- URI references of predicates are written as a combination of a prefix denoting a namespace URI and a local element name (so-called XML QNames)
- URI references of subjects and objects are written as XML attribute values
- literal nodes (which are always object nodes) become XML element text content or attribute values

RDF Serialisation - Example

- there is a person identified by http://www.w3.org/people/em/contact#me, whose name is Eric Miller, whose email address is em@w3.org, and whose title is Dr.
  -
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

Multiple Statements

- an RDF graph consisting of multiple statements can be represented using multiple description elements:
  -
```
<?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/"
           xmlns:exterms="http://www.example.org/terms/">
    <rdf:Description rdf:about="http://www.example.org/index.html">
      <exterms:creation-date>August 16, 1999</exterms:creation-date>
    </rdf:Description>
    <rdf:Description rdf:about="http://www.example.org/index.html">
      <dc:language>en</dc:language>
    </rdf:Description>
  </rdf:RDF>
```

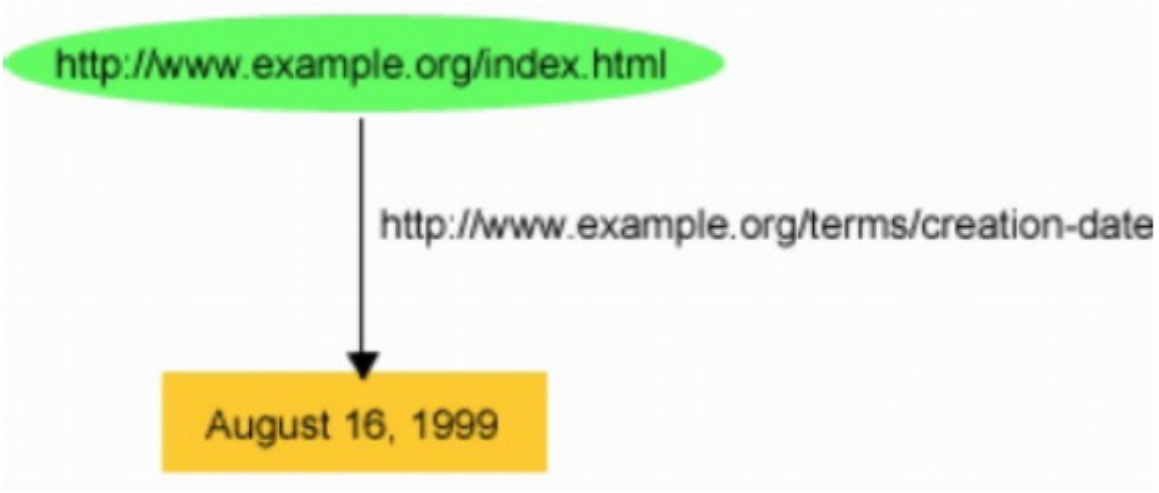
Multiple Predicates

- a description element may also contain multiple predicates (predicates may even import more than one namespace):
  -
```
<?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/"
          xmlns:exterms="http://www.example.org/terms/">
   <rdf:Description rdf:about="http://www.example.org/index.html">
     <exterms:creation-date>August 16, 1999</exterms:creation-date>
     <dc:language>en</dc:language>
     <dc:creator
rdf:resource="http://www.example.org/staffid/85740"/>
   </rdf:Description>
 </rdf:RDF>
```

Serialisation - Blank Nodes

- blank nodes are serialised using node identifiers
  -
```
<?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/"
           xmlns:exterms="http://example.org/stuff/1.0/">
    <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
      <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
      <exterms:editor rdf:nodeID="abc"/>
    </rdf:Description>
    <rdf:Description rdf:nodeID="abc">
      <exterms:fullName>Dave Beckett</exterms:fullName>
      <exterms:homePage rdf:resource="http://purl.org/net/dajobe/"/>
    </rdf:Description>
  </rdf:RDF>
```

Serialisation: Typed Literals (I)



Serialisation: Typed Literals (II)

- <?xml version="1.0"?>
  - <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:exterms="http://www.example.org/terms/">
    <rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date
    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
    1999-08-16</exterms:creation-date>
    </rdf:Description>
    </rdf:RDF>

Using XML Entities

-
```
<?xml version="1.0"?>
 <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:exterms="http://www.example.org/terms/">
     <rdf:Description rdf:about="http://www.example.org/index.html">
       <exterms:creation-date rdf:datatype="&xsd;date">1999-08-16
       </exterms:creation-date>
     </rdf:Description>
   </rdf:RDF>
```

RDF-Containers

- `rdf:Bag`
  - group of resources or literals, possibly including duplicate members, without order
- `rdf:Seq`
  - group of resources or literals, possibly including duplicate members, where the order of the members is significant
- `rdf:Alt`
  - group of resources or literals that are alternatives

RDF-Containers - Example

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:s="http://example.org/students/vocab#">
    <rdf:Description rdf:about="http://example.org/courses/6.001">
      <s:students>
        <rdf:Bag>
          <rdf:li rdf:resource="http://example.org/students/Amy"/>
          <rdf:li rdf:resource="http://example.org/students/Tom"/>
          <rdf:li rdf:resource="http://example.org/students/Jim"/>
        </rdf:Bag>
      </s:students>
    </rdf:Description>
</rdf:RDF>
```

Resources

- literature
  - Powers, S. (2003): Practical RDF. O'Reilly.
- specification
  - http://www.w3.org/RDF/
- validator
  - http://www.w3.org/RDF/Validator/
- RDF/XML syntax specification
  - http://www.w3.org/TR/rdf-syntax-grammar/