Semantic Web and RDF
Introduction
Denis Helic
Web today: more than 10 billions of Web pages

How to discover information on the Web (very hard!)?

- Browsing
- Searching

Very often results are disappointing!
How do we find things in real world?

Example 1: The Library

- You want to find a book on Relational Databases
- Use the computer in the library to list books by author, title, subject, etc.
- Identify the book you want and look in the list where the book is
Example 2: The Video Store

- You want to find the latest Star Wars movie
- Use the computer in the store to list books by director, title, actors, etc.
- Identify the video you want and look in the list where the video is

Both examples based on metadata
Metadata is data about data

Example 1: Data about books

- Author: Nick Scerbakov
- Subject: Databases
- Title: From Databases to Hypermedia
- ...

...
Example 2: Data about videos

- Director: George Lucas
- Title: Star Wars - Revenge of the Sith
- Actor: Ewan McGregor

...
Metadata comes in key-value pairs

- Key: Author
- Value: Nick Scerbakov

For different applications we use different sets of metadata

- Library: author, title, subject, ...
- Video store: director, title, actors, ...
We can use metadata for other purposes as well!

- Cataloging
- Printing
- Sorting, etc.
Use metadata to describe Web resources

Web sites, Web pages, etc.

We can apply metadata on the Web in the same way as in the real world

searching, cataloging, printing, etc..

We can apply metadata in a more Web-specific way

provide a summary of a Web site, describe intellectual property rights of a Web page, etc.
Metadata is very useful for multimedia resources

- images, videos, music, etc.

We can try to solve problems with searching for multimedia resources

Example: Describing digital images with metadata

- what objects do I have on an image (e.g. dog, cat, etc.)
Current situation with metadata on the Web

Not that much metadata on the Web

Especially, not that much in HTML pages

<meta> element in HTML

used to raise search engine rankings!

search engines do not query <meta> elements directly

Some systems apply metadata for more than 10 years now! (e.g. Hyperwave)
What are the challenges for metadata on the Web?

Web comprises many different applications

- Many libraries on the Web
- Many video stores on the Web
- ...

Applications use different sets of metadata, but the principles are the same (i.e. metadata comes always in key-value pairs)
What do we need to use metadata effectively on the Web?

Technology:

- That supports the basic metadata principle
- That allows to use different sets of metadata
- That is compatible with current Web standards
What do we need to use metadata effectively on the Web (continued)?

Standard (standardized technology):

- We want to exchange metadata with other people
- We want software to process it automatically

We need to agree on a standard for metadata!
Resource Description Framework (RDF) is such standardized technology
Developed by Web Consortium (Recommendation)

http://www.w3.org/RDF

Part of larger initiative called Semantic Web

http://www.w3.org/2001/sw
New initiative by Web Consortium

Definition (from W3C): *The Semantic Web is the representation of data on the World Wide Web*

Description of Web resources in the form of metadata
Example: Web resource described with some metadata

- Web resource: http://coronet.iicm.edu/lectures/mmis2
- Creator: Denis Helic
- Administrator: Denis Helic
- Title: Multimedia Information Systems
- ...
Design ideas behind Semantic Web

- Based on the current Web technologies, such as URL (URI) and XML

- Extension of the current Web

- Make it very simple to create metadata! (recollect the reason for the success of the Web)

- Make it possible to introduce different metadata sets

- Anyone, anywhere, anytime can provide metadata for a resource by addressing it with its URI (similar to linking in HTML)
Goals of Semantic Web

- Web of descriptions and resources - Semantic Web
- Enable effective information discovery (e.g. search)
- Enable automation (e.g. software agents)
- Enable effective integration (e.g. ftp and http)
Resource Description Framework - RDF

Resource Description Framework is a recent development of Web Consortium

http://www.w3.org/RDF

Purpose of RDF is to provide a standard for exchanging metadata on the Web

The basic RDF specification consists of:

- RDF Data Model: http://www.w3.org/TR/rdf-concepts - status Recommendation

Design goals for RDF data model are:

- To match the basic metadata principle (i.e. to describe Web resources by applying simple key-value pairs)
- A simple data model
- Scalable (it has to scale over the whole Web!)
- Anyone can provide metadata for any resource
- Independent (e.g. anyone can invent metadata sets for their purposes)
- Interchangeable (e.g. easy to format it as XML)
RDF Data Model is based on the following rules:

- A **Resource** is anything that can have a URI (e.g. all Web pages, all Web images, all files accessible through ftp, etc.)
  
  ```
  http://coronet.iicm.edu/lectures/mmis2
  ```

- A **Property** has a name and describes some relationship (e.g. Creator, Title, Subject, etc.)

- A **Statement** consists of a combination of a Resource, a Property and a value
  
  "The creator of [http://coronet.iicm.edu/lectures/mmis2](http://coronet.iicm.edu/lectures/mmis2) is Denis Helic"
Statement may be represented as a triple (N3 notation):

<http://coronet.iicm.edu/lectures/mmis2> <creator> <Denis Helic>.

<Resource> <Property> <Value>.

<Subject> <Predicate> <Object>.
Statement may be also represented graphically

http://coronet.iicm.edu/lectures/mmis2

http://coronet.iicm.edu/mmis2/examples/rdf#creator

Denis Helic
A particular description of a resource includes a number of statements:

"The creator of http://coronet.iicm.edu/lectures/mmis2 is Denis Helic"

"The administrator of http://coronet.iicm.edu/lectures/mmis2 is Denis Helic"

"The title of http://coronet.iicm.edu/lectures/mmis2 is Multimedia Information Systems 2"
Graphically

http://coronet.iicm.edu/lectures/mmis2

http://coronet.iicm.edu/mmis2/examples/rdf#creator
Denis Helic

http://coronet.iicm.edu/mmis2/examples/rdf#administrator
Denis Helic

http://coronet.iicm.edu/mmis2/examples/rdf#title
Multimedia Information Systems
RDF graph consists of nodes and arcs

Oval nodes represent resources

Square nodes represent values

Arcs (arrows) represent properties
A value might be a resource

"The homepage of http://coronet.iicm.edu/lectures/mmis2 is http://coronet.iicm.edu"

Advanced concepts in RDF

- A Property is also a Resource
- A Statement is also a Resource

Both can have their own properties (e.g. when it was created?)
How this model meets the design ideas?

- To match the basic metadata principle - each triple is a combination of a URI and a key-value pair
- A simple data model - just triples
- Scalable - easy to handle even large number of triples
- Anyone can provide metadata for any resource - if you know its URI!
- Independent - no prescribed sets of metadata
- Interchangeable - we need to specify XML syntax for it!
RDF/XML Syntax Specification (1/11)

 Defines how to encode an RDF graph into a valid XML

 We need to represent nodes and arcs as XML elements, attributes, element content and attribute values

 A graph is a collection of paths of the form Node, Arc, Node, ...

 In RDF/XML these turn into sequences of nested elements which alternate between elements for Nodes and Arcs
Example 1
First path: creator

<rdf:Description rdf:about =
    "http://coronet.iicm.edu/lectures/mmis2">
    <creator>Denis Helic</creator>
</rdf:Description>
Second path: administrator

```xml
<rdf:Description rdf:about =
    "http://coronet.iicm.edu/lectures/mmis2">
    <administrator>Denis Helic</administrator>
</rdf:Description>
```

Third path: title

```xml
<rdf:Description rdf:about =
    "http://coronet.iicm.edu/lectures/mmis2">
    <title>Multimedia Information Systems 2</title>
</rdf:Description>
```
Example 1 (complete)

```xml
<rdf:Description rdf:about =
    "http://coronet.iicm.edu/lectures/mmis2">
    <creator>Denis Helic</creator>
    <administrator>Denis Helic</administrator>
    <title>Multimedia Information Systems 2</title>
</rdf:Description>
```

**Example 1:**

`http://coronet.iicm.edu/mmis2/examples/rdf/mmis2.rdf`
Example 2
Example 2 (continued)

```xml
<rdf:Description rdf:about = "http://coronet.iicm.edu/lectures/mmis2">
    <creator>Denis Helic</creator>
    <administrator>Denis Helic</administrator>
    <title>Multimedia Information Systems 2</title>
    <homepage>
        <rdf:Description rdf:about = "http://courses.iicm.edu">
            <creator>Hermann Mauerer</creator>
            <administrator>Karl Trummer</administrator>
            <title>Courses offered by IICM</title>
        </rdf:Description>
    </homepage>
</rdf:Description>
```
To complete the example we need declaration!

```xml
<?xml version="1.0" encoding="UTF-8"?>
<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 = "http://coronet.iicm.edu/mmis/examples/rdf#">
    ...
</rdf:RDF>
```

Example 2:

[http://coronet.iicm.edu/mmis2/examples/rdf/mmis2_complete.rdf](http://coronet.iicm.edu/mmis2/examples/rdf/mmis2_complete.rdf)
Striping pattern!

```xml
<rdf:Description rdf:about = "http://coronete.iicm.edu/lectures/mmis2">
  ...
  <homepage>
    <rdf:Description rdf:about = "http://coronet.iicm.edu">
      <creator>Hermann Mauerer</creator>
    </rdf:Description>
  ...
  </rdf:Description>
</homepage>
</rdf:Description>
```
Gray (Resource) - white (property) stripes

```xml
<rdf:Description rdf:about = "http://courses.iicm.edu/mmis">
    <homepage>
        <rdf:Description rdf:about = "http://courses.iicm.edu">
            <creator>Hermann Mauerer</creator>
        </rdf:Description>
    </homepage>
</rdf:Description>

<Resource-A>
    <property-A>
        <Resource-B>
            <property-B>value</property-B>
        </Resource-B>
    </property-A>
</Resource-A>
```
Check that the RDF/XML document validates

W3C Validator:
http://www.w3.org/RDF/Validator

Namespaces included into graph

Needed because everybody can define properties
To represent collections we use `rdf:Bag`, `rdf:Seq` or `rdf:Alt`.

```
<rdf:Description rdf:about = "http://coronet.iicm.edu/lectures/mmis2">
  <topics>
    <rdf:Bag>
      <topic>Internet</topic>
      <topic>XSLT</topic>
      <topic>RDF</topic>
      <topic>Cocoon</topic>
    </rdf:Bag>
  </topics>
</rdf:Description>
```
RDF Advanced Features (2/8)

- `rdf:Bag` is an unordered collection.
- `rdf:Seq` is a sequence of properties.
- `rdf:Alt` is a set of alternate properties (e.g., represent title in different languages).
Query language for RDF

SELECT ?x
WHERE (http://somewhere/res1, http://somewhere/pred1, ?x)

RDQL, Part of Jena RDF framework:
http://jena.sourceforge.net/

More on the Jena framework in the next lecture
Anyone can provide metadata for any resource - if you know its URI!

The first description:
http://coronet.iicm.edu/mmis2/examples/rdf/mmis2.rdf

An additional description:
http://coronet.iicm.edu/mmis2/examples/rdf/mmis2_add.rdf
<rdf:Description
    rdf:about = "http://coronet.iicm.edu/lectures/mmis2">
    ...
</rdf:Description>

Refer to resource with the URI!
An *aggregator* tool collects all metadata about a resource.

We can query all metadata about the specific resource.

Metadata might be distributed all over the Web, i.e., we have a distributed network of metadata.

Basics of Semantic Web!
Anybody can define metadata sets

RDF Schema: defines all properties

Example Dublin Core Metadata Schema (Library background)

<table>
<thead>
<tr>
<th>Content</th>
<th>Intellectual Property</th>
<th>Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Creator</td>
<td>Date</td>
</tr>
<tr>
<td>Subject</td>
<td>Publisher</td>
<td>Type</td>
</tr>
<tr>
<td>Description</td>
<td>Contributor</td>
<td>Format</td>
</tr>
<tr>
<td>Language</td>
<td>Rights</td>
<td>Identifier</td>
</tr>
<tr>
<td>Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coverage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dublin Core: [http://dublincore.org/](http://dublincore.org/)

In many cases you need just Dublin Core

FOAF Project

Friend of a friend

Creating RDF descriptions of people, contacts, homepages, etc.

To support communication, networking between people on the Web

URL: [http://www.foaf-project.org/](http://www.foaf-project.org/)
Further examples on http://www.rdfdata.org/

Open directory RDF dump
http://rdf.dmoz.org/

US State and Regions
http://www.daml.ri.cmu.edu/ont/USRegionState.daml

AI bibliography
http://www.isi.edu/webscripter/planning.scheduling.daml
Expressing bibliographies in RDF

Allows you to exchange bibliographic information more easily

PublicationDB
http://coronet.iicm.edu/struts-publicationdb

Publications
http://coronet.iicm.edu/denis/pubs/list.rdf
Applications using RDF and other Semantic Web technologies

Easiest way to implement Semantic Web applications

Using a Semantic Web framework

Jena Framework

Open source, Java-based, implemented by Hewlett Packard

Project homepage

http://jena.sourceforge.net/
Jena framework provides

- A programmatic environment for RDF and other Semantic Web technologies
- RDF API, i.e., manipulating RDF graphs using an API
- Reading and writing RDF in RDF/XML, N3 and N-Triples
- In-memory and persistent storage (with RDBMS)
- RDQL - a query language for RDF
- API for other Semantic Web technologies
Build a server-side application for manipulating of RDF graphs (models)

Basic RDF Server (BRDFS)

- Implemented as a master thesis at IICM
- Implemented using Jena Framework

I.e., Jena framework accesible through Web interface
Requirements for BRDFS

- Running as a Web application, i.e., available through HTTP
- Persistent data
- Can create, delete, update models and statements
- Can merge models, query models, etc.
- Easy to use for humans and software agents
BRDFS implementation

As a Web application using Tomcat

Server is an action framework, i.e., similar to Struts

You can issue commands to the server using the action framework

HTML user interface
BRDFS Run-time

At start time the server is connected to a MySQL database

After that every RDF model operation is persisted in that database

Jena provides DB Connection

Jena abstracts the database layer

We just deal with models
Create a new model

```java
ModelMaker maker =
    ModelFactory.createModelRDBMaker(
        jena2_dbconnection_.getJena2DBConnection());
Model m = maker.createModel(modelname);
    m.close();
```

Model name comes from the request

Source code

```java
http://coronet.iicm.edu/mmis2/examples/rdf/brdfs/src/
edu/iicm/brdfs/action/NEWMODEL.java
```
Delete a model

```java
ModelRDB m =
    ModelRDB.open(
        jena2_dbconnection_.getJena2DBConnection(),
        modelname);

m.remove();
m.close();
```

Model name comes from the request

Source code

Merge two models

ModelMaker maker =
    ModelFactory.createModelRDBMaker(
        jena2_dbconnection_.getJena2DBConnection());
Model target = maker.openModel(modelname);
target.begin();
...
    Model merge_model = maker.openModel(tomerge_values[i]);
    target.add(merge_model, true);
    merge_model.close();
target.commit();
target.close();

Source code

http://coronet.iicm.edu/mmis2/examples/rdf/brdfs/src/edu/iicm/brdfs/action/MERGEMODELS.java
Add statements

ModelMaker maker =
    ModelFactory.
    createModelRDBMaker(
        jena2_dbconnection_.getJena2DBConnection());
Model model = maker.openModel(modelname);
model.begin();
Resource resource = model.createResource(subject);
Property property = model.createProperty(namespace, propertyname);
resource.addProperty(property, object);
model.commit();
model.close();

Source code

http://coronet.iicm.edu/mmis2/examples/rdf/brdfs/src/edu/iicm/brdfs/action/ADDstAtementS.java
Query a model

ModelRDB model = ModelRDB.open(
    jena2_dbconnection_.getJena2DBConnection(), modelname);
Query query = new Query(querystring);
query.setSource(model);
QueryExecution qe = new QueryEngine(query);
QueryResults results = qe.exec();
response_writer_.writeQueryResult(results);
results.close();

Source code

http://coronet.iicm.edu/mmis2/examples/rdf/brdfs/src/
edu/iicm/brdfs/action/QUERYMODEL.java
RDQL is a query language for querying RDF models

RDQL Tutorial

http://jena.sourceforge.net/tutorial/RDQL/index.html

Similar to SQL

But operations on triples not on relations

```
SELECT ?x
WHERE ( ?x <http://www.w3.org/2001/vcard-rdf/3.0#FN> "John Smith")
```
BRDFS accesible from
http://coronet.iicm.edu/brdfs

We upload two MMIS2 course RDF files from the last lecture

Intellectual property rights of the MMIS2 homepage
http://coronet.iicm.edu/mmis2/examples/rdf/mmis2.rdf

Information about topics of the course
http://coronet.iicm.edu/mmis2/examples/rdf/mmis2_add.rdf

We merge the models and query the final model

  Shows also how metadata integration is achieved
Get creator of the MMIS2 homepage

SELECT ?creator
WHERE ("http://coronet.iicm.edu/lectures/mmis2"
    "http://coronet.iicm.edu/mmis2/examples/rdf#creator" ?creator)
Get the course topics

SELECT ?topic
WHERE (<http://coronet.iicm.edu/lectures/mmis2>
    <http://coronet.iicm.edu/mmis2/examples/rdf#topic>
    ?topic)
Another master thesis project at IICM

Implement OAI provider for the JUCS journal

JUCS journal

http://www.jucs.org/

You can retrieve BibTeX reference for each of the articles
Open Archives Initiative - OAI

http://www.openarchives.org/

Providing metadata about publications in digital libraries

Metadata format is not specified

Should support at least Dublin Core format

Defines also a number of protocols for accessing and retrieving the data
OAI provider for JUCS built on the top of BRDFS

Metadata managed in RDF

- Different output formats available
- BibTeX format encoded as RDF, i.e., as XML
- Dublin Core format encoded as RDF, i.e., as XML
Automatic adding of references for new issues of the journal

Automatic conversion of BibTeX text format onto BibTeX RDF

Adding of statements to an already existing RDF model

HTML format for people
  - Supports browsing
  - Automatic conversion of XML formats using XSLT's on the client side
Administration interface

http://coronet.iicm.edu/brdfs

Uploading of new BibTeX reference files

Retrieving of the model

Querying of the model
Select IDs and titles of all articles

```
SELECT ?id, ?title
WHERE (?id <http://coronet.iicm.edu/bibtex#journal> "Journal of Universal Computer Science")
(?id <http://coronet.iicm.edu/bibtex#title> ?title)
```
Select titles and authors

SELECT ?title, ?author
WHERE (?x <http://coronet.iicm.edu/bibtex#journal> "Journal of Universal Computer Science")
(?x <http://coronet.iicm.edu/bibtex#title> ?title)
(?x <http://coronet.iicm.edu/bibtex#author> ?author)
Use abbreviations for namespaces

SELECT ?title, ?author
WHERE (?x bibtex:journal "Journal of Universal Computer Science")
(?x bibtex:title ?title)
(?x bibtex:author ?author)
USING bibtex FOR <http://coronet.iicm.edu/bibtex#>
Select titles and authors for specific issues

SELECT ?title, ?author
WHERE (?x bibtex:journal
   "Journal of Universal Computer Science")
(?x bibtex:title ?title)
(?x bibtex:author ?author)
(?x bibtex:volume ?volume)
AND ?volume == 1
USING bibtex FOR <http://coronet.iicm.edu/bibtex#>
Add another issue to the database

- Shows the integration of metadata

HTML user interface

http://coronet.iicm.edu/brdfs/oai/

- With XSLT on the client side
RDF is very simple

It provides a basic set for information modelling

Because of this simplicity it is a perfect assembly language

It is possible to build other information modelling languages on top of it

RDF Schema is one such language

http://www.w3.org/TR/rdf-schema/

RDFS is developed by the Web consortium
RDF Schema is an object-oriented modelling language

It allows you to define classes

It allows you to define relationships between classes

  E.g., subclass relationship

It allows you to define properties of classes
How does RDFS relate to RDF?

- Classes are resources
  - Each class has a URL, so we can define properties for it
- If two classes are related with subClassOf property then we build class hierarchies
- Simple properties of a class are same as properties that have simple values in RDF
RDFS Example

At the top of the class hierarchy we have WaterSource.

Two subclasses of WaterSource are Stream and BodyOfWater.
  - Stream has subclasses such as River, Creek, etc.
  - BodyOfWater has subclasses such as Lake, Sea, etc.

There is a relationship between Stream and BodyOfWater.
  - The relationship is called emptiesInto.
Definition of classes and relationships in RDFS

```xml
<rdfs:Class rdf:ID="WaterSource"/>

<rdfs:Class rdf:ID="Stream">
  <rdfs:subClassOf rdf:resource="#WaterSource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="BodyOfWater">
  <rdfs:subClassOf rdf:resource="#WaterSource"/>
</rdfs:Class>
```
Definition of classes and relationships in RDFS (continued)

```xml
<rdfs:Class rdf:ID="River">
    <rdfs:subClassOf rdf:resource="#Stream"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Lake">
    <rdfs:subClassOf rdf:resource="#BodyOfWater"/>
</rdfs:Class>
...

<rdf:Property rdf:ID="emptiesInto">
    <rdfs:range rdf:resource="#BodyOfWater"/>
    <rdfs:domain rdf:resource="#River"/>
</rdf:Property>
```
Using RDF schema with RDF

Basically, we describe now real Web resources according to the developed schema

...<River rdf:ID="Yangtze"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns="http://www.geodesy.org/water/naturally-occurring#">
   <emptiesInto
      rdf:resource="http://www.china.org/geography#EastChinaSea"/>
</River>

...
What are the advantages of applying RDF Schema to this small RDF graph?

Inference!

We can infer new facts from already existing facts and classes and relationships defined in the RDF schema.

River is a subclass of Stream → Yangtze is a stream

Stream is a subclass of WaterSource → Yangtze is a WaterSource

River emptiesInto BodyOfWater → EastChinaSea is a BodyOfWater
The newly inferred facts can be also used to improve information discovery.

E.g., you are searching for streams that empty into EastChinaSea.

You get the info about Yangtze.
Web Ontology Language - OWL (1/8)

OWL is built on the top of RDFS
  i.e., it extends RDFS

Work prior to OWL
  DAML (developed by DARPA)
    http://www.daml.org/
  OIL
    http://www.ontoknowledge.org/oil/index.shtml

OWL developed by the Web consortium (Recommendation)
    http://www.w3.org/2004/OWL/
OWL is an extension of RDFS

It allows you to define richer relationships than RDFS

Inferencing capabilities are more powerful

Consequently, you can improve information discovery even more
Using OWL to define rich properties

<owl:ObjectProperty rdf:ID="connectsTo">
  <rdf:type>
    rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"
  </rdf:type>
  <rdfs:domain rdf:resource="#WaterSource"/>
  <rdfs:range rdf:resource="#WaterSource"/>
</owl:ObjectProperty>
connectsTo relationship is symmetric

i.e., if A connectsTo B, then B connectsTo A

```xml
<River rdf:ID="Yangtze"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <connectsTo>
    <River rdf:about="http://www.china.org/rivers#Wu"/>
  </connectsTo>
</River>
```

Give me all rivers that connectTo Wu → Yangtze
Transitive property

<owl:ObjectProperty rdf:ID="containedIn">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
  <rdfs:domain rdf:resource="#Sea"/>
  <rdfs:range rdf:resource="#BodyOfWater"/>
</owl:ObjectProperty>

i.e., if A is containedIn B and B is contained in C then A is contained in C
<Sea rdf:ID="EastChinaSea"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <containedIn>
    <Sea rdf:about="http://www.china.gov#ChinaSea"/>
  </containedIn>
</Sea>
<Sea rdf:about="http://www.china.gov#ChinaSea"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns="http://www.geodesy.org/water/naturally-occurring#">
  <containedIn>
    <Ocean rdf:about="http://www.geodesy.org#PacificOcean"/>
  </containedIn>
</Sea>

EastChinaSea is containedIn PacificOcean

subClassOf is a transitive property
Other properties of relationships

- FunctionalProperty, there is at most one value for the property

- InverseProperty

  - containedIn → contains

- DatatypeProperty, the value of the property is a simple datatype

  - XML schema datatypes
Using OWL to improve information discovery while browsing

We have a typical directory of information

  e.g., similar to Yahoo categories

We can browse through categories and find the information we need
We can model categories as OWL classes

OWL classes are related with subClassOf relation

subClassOf relation is transitive

e.g., Class A is a subClassOf class B

i.e., an instance of class A is also an instance of class B

e.g., if a document belongs to a subcategory it also belongs to its super-category
We had a lot of learning resources on Computer Science

We wanted to categorize them and improve information discovery

Firstly, we created an ontology of Computer Science

i.e., categorization of computer science fields

e.g., information systems, algorithms, etc.

Finally, we added resources to the categories

When you open a category you see all resources including resources from subcategories
“SuchAlgorithmus” is a subclass of “Algorithmus”

“BinäreSuche” is an instance of “SuchAlgorithmus”

“BinäreSuche” is therefore an instance of “Algorithmus”

Additionally, a category can be a subclass of more than one superclass

Example

http://coronet.iicm.edu/navig
Using the system to personalize the information discovery

We define a personal class for each user

What happens if we declare the “Personal” class as a superclass of some class in Computer Science
e.g., Personal is a superclass of "Algorithmus"

All instances of "Algorithmus" are instances of "Personal"

We access these instances directly

I.e., we declare our interest in algorithms

A new instance in algorithms → we see it immediately

Example

http://coronet.iicm.edu/navig
RDF Tools/Links

Protege Editor
http://protege.stanford.edu/

IsaViz, visual authoring tool for RDF
http://www.w3.org/2001/11/IsaViz/

Jena RDF Framework, Java parser, Java API, RDQL
http://jena.sourceforge.net

W3C RDF Validator
http://www.w3.org/RDF/Validator/

As editor: any XML editor, emacs, etc.
Further Readings

- RDF Tutorial

- Intro to RDF
  [http://www.dlib.org/dlib/may98/miller/05miller.html](http://www.dlib.org/dlib/may98/miller/05miller.html)

- RDF Tutorial and examples

- Semantic Web articles from [http://www.xml.com](http://www.xml.com)

- What is RDF?

- Metadata articles from [http://www.xml.com](http://www.xml.com)