Freie Universität Berlin

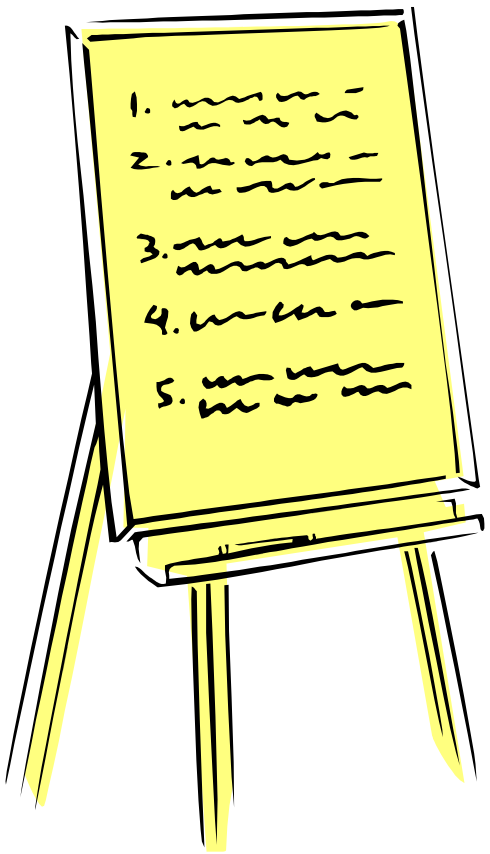
Winter term 2012/2013
Web-based Information Systems

Alexandru Todor
Freie Universität Berlin
Department of Computer Science
Corporate Semantic Web Workgroup
mailto: todor@inf.fu-berlin.de

Agenda

- Presentations
- Tutorial
  - RDF/S
  - SPARQL
  - Jena

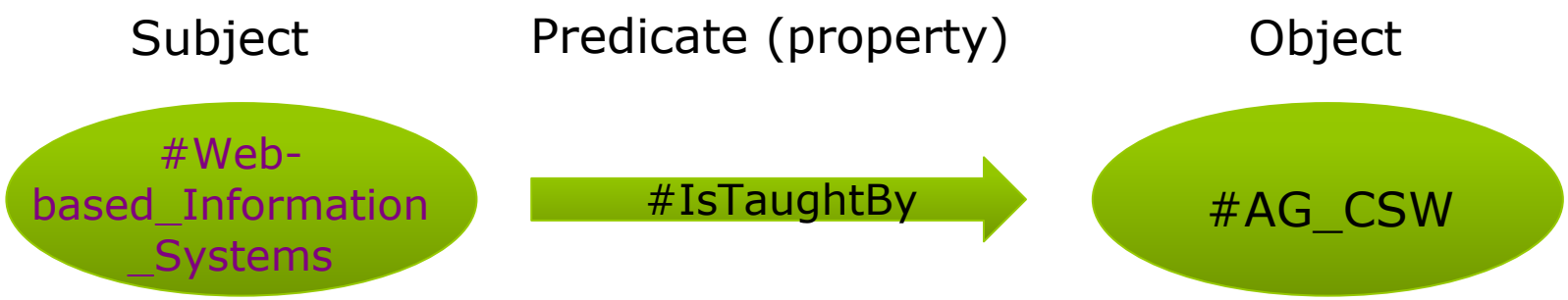
Presentations



- Present solutions of exercise 1
  - ?
- Set-up
  - 12 min sharp (!) presentation
    + 3 min discussion
  - Be prepared to comment

RDF

- RDF is made out of Triples (also called statements)
  - Each triple is made out of Subject – Predicate – Object pair



- Subjects are always Resources
- Objects can be resources or values

```
<rdf:Description rdf:about="subject">
      <predicate rdf:resource="object" />
      <predicate>literal value</predicate>
<rdf:Description>
```

RDF

- Simple Document

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

    <!– Body Code -->

</rdf:RDF>
```

- Statement

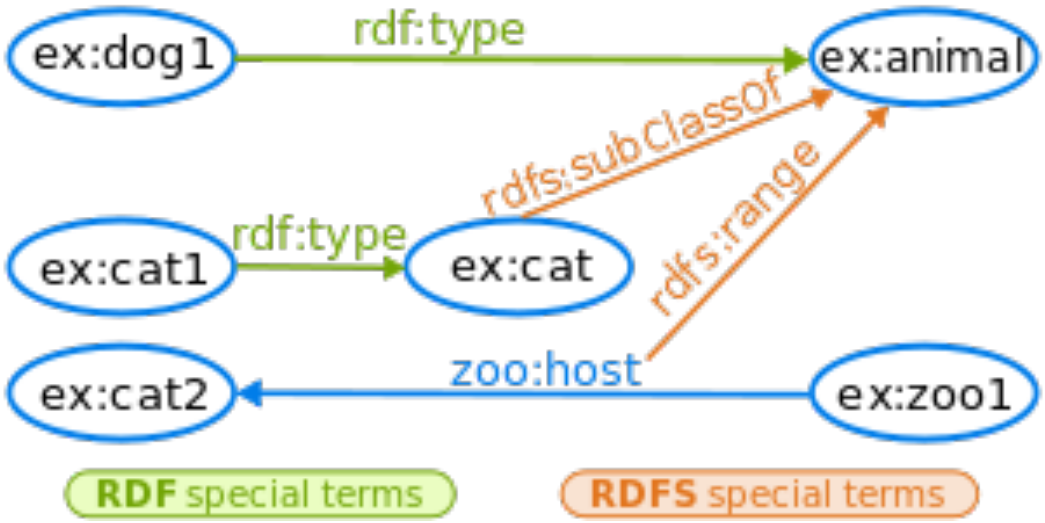
```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

    <rdf:Description rdf:about="http://www.inf.fu-berlin.de/en/groups/ag-csw/Teaching/Wintersemester_2011_12/Web-based_Information_Systems">

        <!-- Statement Code -->

    </rdf:Description>
</rdf:RDF>
```

- Properties (Predicates)

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:lectureprop=" http://www.inf.fu-berlin.de/properties#">

    <rdf:Description rdf:about="http://www.inf.fu-berlin.de/en/groups/ag-csw/Teaching/Wintersemester_2011_12/Web-based_Information_Systems ">

<leactureprop:isTaughtBy rdf:resource="http://www.corporate-semantic-web.de"/>

    </rdf:Description>

</rdf:RDF>
```

RDFS

- Defines classes of resources
  - rdfs:Class
  - rdfs:Resource
- Creates a hierarchy
- Adds more semantics to properties



CC Image from Wikipedia

RDF/RDFS

- Please read
  - http://www.w3schools.com/rdf/

SPARQL

- Example

  select **?**lecture where **{ <**http**:**//example.com/teacher/John><uni: isTeacherOf> ?lecture }

- Please read:

  http://www.cambridgesemantics.com/2008/09/sparql-by-example/

Jena

- Java Semantic Web Framework
  - Reading and writing RDF
  - An OWL API
  - Different storage engines
  - SPARQL query engine

Jena Examples

http://jena.sourceforge.net/tutorial/RDF_API

- Creating a model (from http://jena.sourceforge.net/tutorial/RDF_API/)

```
String personURI    = "http://somewhere/JohnSmith";
String givenName    = "John";
String familyName   = "Smith";
String fullName     = givenName + " " + familyName;


// create an empty Model
Model model = ModelFactory.createDefaultModel();


// create the resource
//   and add the properties cascading style
Resource johnSmith
  = model.createResource(personURI)
        .addProperty(VCARD.FN, fullName)
        .addProperty(VCARD.N,
                     model.createResource()
                          .addProperty(VCARD.Given, givenName)
                          .addProperty(VCARD.Family, familyName));
```

Jena - Sparql

Freie Universität Berlin

String queryString="select * where {<http://dbpedia.org/resource/Berlin> ?p ?o} limit 100"

```
// create a query object
Query query = QueryFactory.create(queryString);

// initialize the queryExecution factory with a remote service.
QueryExecution qexec = QueryExecutionFactory.sparqlService("http://dbpedia.org/sparql", query);

try {
    ResultSet results = qexec.execSelect();
    for (; results.hasNext();) {

    // Result processing is done here.
    }
}
finally {
   qexec.close();
}
```

Exercise 7

- Deadline 12.12.2012

End

- Questions ?