

## RDF Examples

Tutorial Slides  
Steffen Staab  
Karlsruhe University

# Tutorial for RDF & RDFS

## HTML Form für Crawl

```
<form method="get" action="http://aifbhades.aifb.uni-  
karlsruhe.de/iswww-serv/GatherAFile">
```

```
<input type="text" name="url" value="http://www.aifb.uni-  
karlsruhe.de/~sst/Teaching/Intelligente%20System%20im%  
20WWW%20SS%202000/test" size="60">
```

```
<input type="submit" value="submit" name="crawlbutton">  
</form>
```

# Tutorial for RDF & RDFS

## Simple Gather Servlet



```
/* Prints the first line of a crawled file as HTML Output */
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.net.*;
```

Testbeispiel auf Übungshomepage!

Vergleiche servlet "GatherAFile" unter /home/iswww/servlets

Slide 3

# Tutorial for RDF & RDFS

## Simple Gather Servlet II



```
public class GatherAFile extends HttpServlet
{
    public void doGet (
        HttpServletRequest request,
        HttpServletResponse response
    ) throws ServletException, IOException
    {
        String urlString =
            request.getParameter("url");
        PrintWriter out;
        String title = "Simple
Servlet Output";
        String
        output = "";
        // set content type and other response header
        // fields first
        response.setContentType("text/html");

        // then write the data of the response
        out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>");
        out.println(title);
        out.println("</TITLE></HEAD><BODY>");
        out.println("<H1>" + title + "</H1>");
        out.println("<P>This is output from The
GatherAFile - an extremely simple
servlet.</P>");
        try {
            output = loadFromURL(urlString);
        }
        catch (Exception e)
        {
            System.out.println(e.toString());
        }
        out.println("The first line of the crawled file
is:" + output);
        out.println("</BODY></HTML>");
        out.close();
    }
}
```

Slide 4

# Tutorial for RDF & RDFS

## Simple Gather Servlet III

```

public static String loadFromURL(String
    urlString) throws Exception{
    URL url;
    try {
        url = new URL(urlString);
    } catch (MalformedURLException e) {
        System.out.println(e.toString());
        return null;
    }
    InputStreamReader isr = new
    InputStreamReader(url.openStream());
    BufferedReader br = new
    BufferedReader(isr);
    return br.readLine();
}

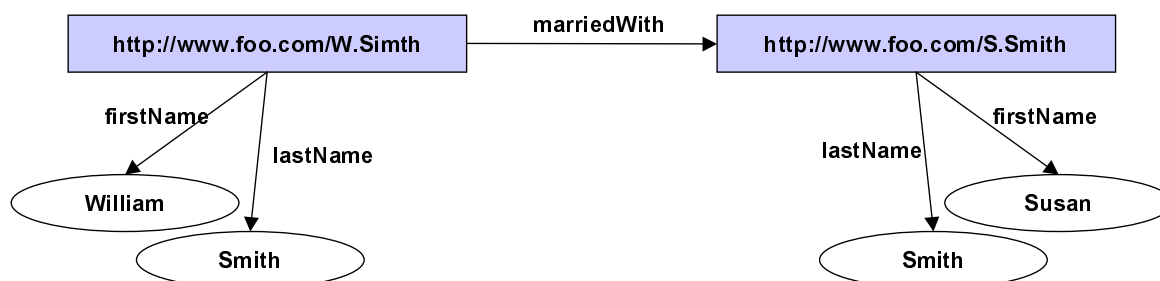
public String getServletInfo()
{
    return "servlet to gather a file and
    print first line";
}

```

Slide 5

# Tutorial for RDF & RDFS

## Two people being married - data model



Slide 6

## Tutorial for RDF & RDFS

### Two people being married - serialization II

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22/rdf-syntax-ns">
  <rdf:description about="http://www.foo.com/W.Smith">
    <firstName>William</firstName>
    <lastName>Smith</lastName>
    <marriedWith>
      <rdf:description about = "http://www.foo.com/S.Smith">
        <firstName>Susan</firstName>
        <lastName>Smith</lastName>
        <rdf:description>
          </marriedWith>
        </rdf:description>
      </marriedWith>
    </rdf:description>
  </rdf:RDF>
```

## Tutorial for RDF & RDFS

### Two people being married - serialization II

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22/rdf-syntax-ns">
  <rdf:Description about="http://www.foo.com/W.Smith">
    <firstName>William</firstName>
    <lastName>Smith</lastName>
    <marriedWith rdf:resource="http://www.foo.com/S.Smith">
  </rdf:Description>
  <rdf:Description about = "http://www.foo.com/S.Smith">
    <firstName>Susan</firstName>
    <lastName>Smith</lastName>
  </rdf:Description>
</rdf:RDF>
```

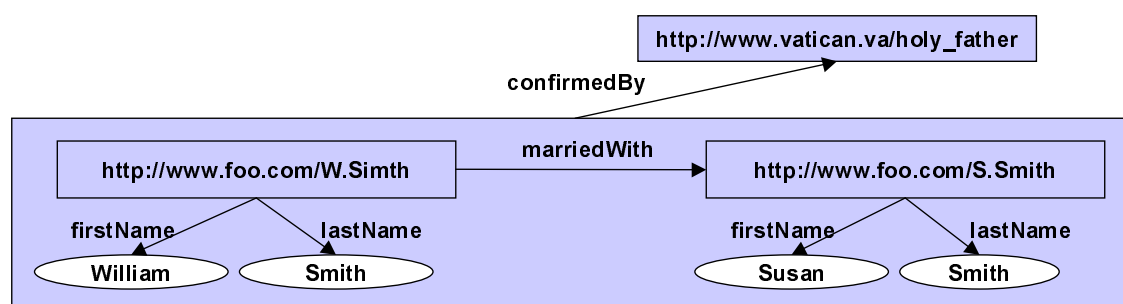
# Tutorial for RDF & RDFS

## Strict principle

Always alternate between resource and property description when going deeper into the tree!

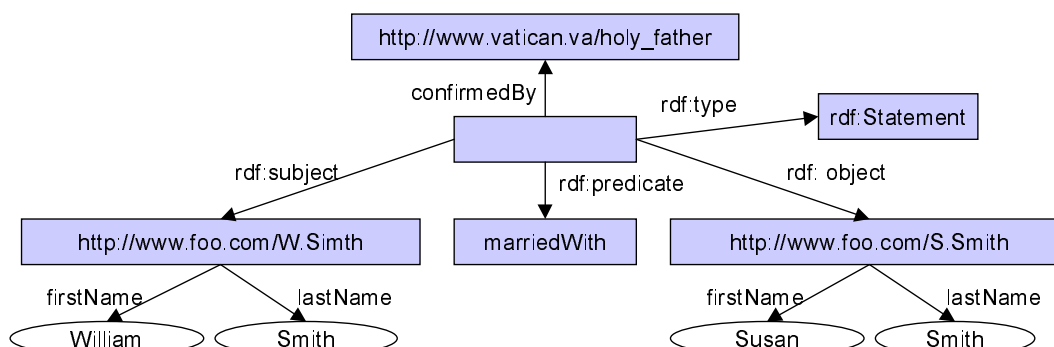
# Tutorial for RDF & RDFS

## Catholic marriage - data model



# Tutorial for RDF & RDFS

## Catholic marriage - equivalent data model



Slide 11

# Tutorial for RDF & RDFS

## Catholic marriage - serialization I

.....

```
<rdf:Description about="">
```

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

```
  <confirmedBy rdf:resource="http://www.vatican.va/holy_father" />
```

```
  <rdf:subject about="http://www.foo.com/W.Smith" />
```

```
  <rdf:predicate rdf:resource="http://www.vatican.va/marriedWith" />
```

```
  <rdf:object>
```

```
    <rdf:Description about="http://www.foo.com/S.Smith">
```

```
      <firstName>Susan</firstName>
```

```
      <lastName>Smith</lastName>
```

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

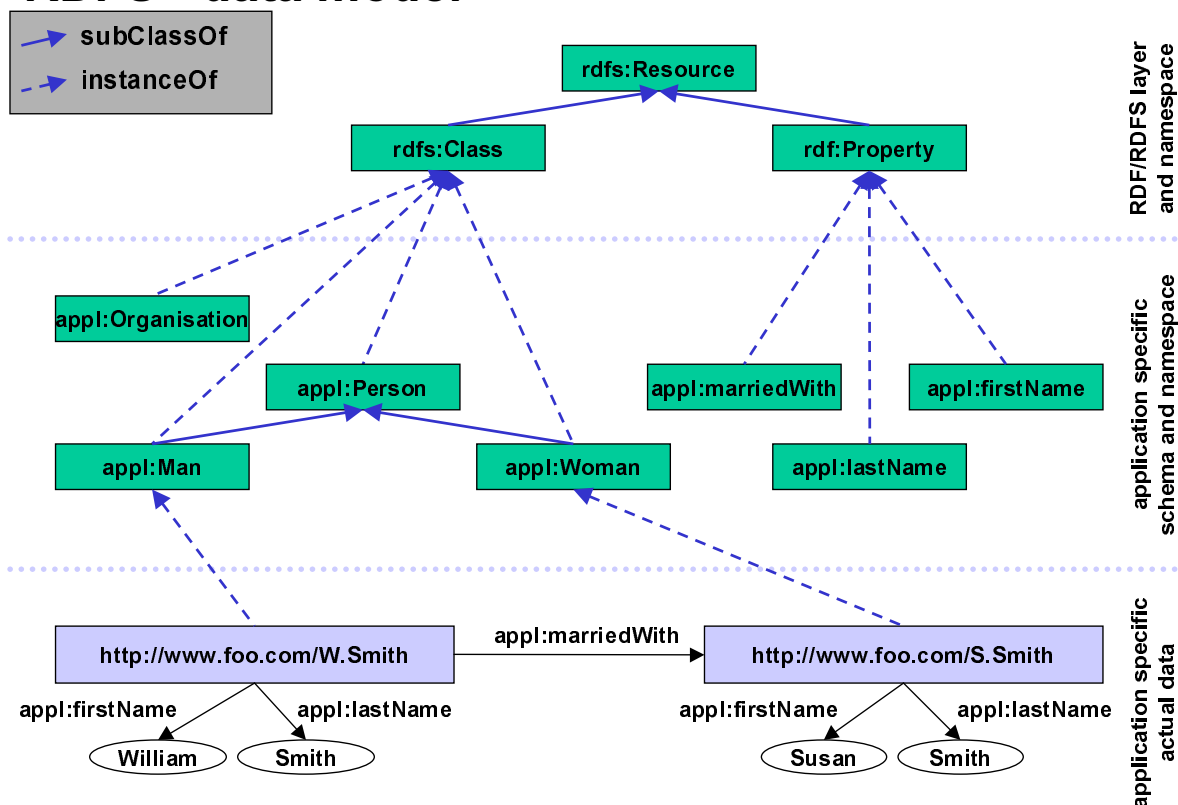
```
  </rdf:object>
```

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

Slide 12

# Tutorial for RDF & RDFS

## RDFS - data model



Slide 13

# Tutorial for RDF & RDFS

## RDFS example serialization 1a

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22/rdf-syntax-ns"
  xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303">

<rdf:Description ID="Person">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <rdfs:subClassOf
    rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
</rdf:Description>

<rdf:Description ID="Man">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <rdfs:subClassOf rdf:resource="#Person"/>
</rdf:Description>

```

Slide 14

## Tutorial for RDF & RDFS

### RDFS example serialization 1b

```
<rdf:Description ID="Woman">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-
    19990303#Class"/>
  <rdfs:subClassOf rdf:resource="#Person"/>
</rdf:Description>
```

```
<rdf:Description ID="Organisation">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-
    19990303#Class"/>
  <rdfs:subClassOf
    rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
</rdf:Description>
```

## Tutorial for RDF & RDFS

### RDFS example serialization 1b

```
<rdf:Description ID="firstName">
  <rdf:type
    resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Property"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="http://www.w3.org/TR/xmlschema-2/#string"/>
</rdf:Description>
```

```
<rdf:Description rdf:ID="marriedWith">
  <rdf:type
    resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Property"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Person"/>
</rdf:Description>
```

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

## Tutorial for RDF & RDFS

### Ontology Metadata

- Title
- Authors
- Subjects
- Version
- ....

Dublin Core -

Set of standard metadata descriptions for librarians

- Title
- Authors
- ....

Slide 17

## Tutorial for RDF & RDFS

### Ontology Metadata with DC und ODOC Ia

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc = "http://purl.oclc.org/dc"
        xmlns:odoc = "http://ontoserver.aifb.uni-karlsruhe.de/schema/ontodoc">
<rdf:Description about = "">
  <dc>Title>An Example Ontology</dc>Title>
  <dc:creator>
    <rdf:Bag>
      <rdf:li>Steffen Staab</rdf:li>
      <rdf:li>Michael Erdmann</rdf:li>
      <rdf:li>Alexander Maedche</rdf:li>
      <rdf:li>Stefan Decker</rdf:li>
    </rdf:Bag>
  </dc:creator>
  <dc:date>2000-02-29</dc:date>
```

Slide 18

## Tutorial for RDF & RDFS

### Ontology Metadata with DC und ODOC Ib



```
<dc:format>text/xml</dc:format>
```

```
<dc:description>
```

```
    An example ontology modeled for this small application
```

```
</dc:description>
```

```
<dc:subject>Ontology, RDF</dc:subject>
```

```
<odoc:url>http://ontoserver.aifb.uni-
```

```
    karlsruhe.de/schema/example.rdf</odoc:url>
```

```
<odoc:version>2.1</odoc:version>
```

```
<odoc:last_modification>2000-03-01</odoc:last_modification>
```

```
<odoc:ka_technique>semi-automatic text knowledge  
    acquisition</odoc:ka_technique>
```

```
.....
```

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

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

Slide 19

## Tutorial for RDF & RDFS

### Inferencing with RDF



Environment

```
/home/iswww
```

SiLRI (Simple Logic-based Rdf Inference engine)

```
/home/iswww/oie
```

Execute example on shell

```
/home/iswww/t.bat
```

```
java com.ontoprise.rdfie.SiLRI ora.q ora.rule -rdf ora.rdf  
    query rule rdf-input
```

Slide 20

# Tutorial for RDF & RDFS

## Übung für nächste Stunde

- Fertige HTML-Form an mit Übergabeparameter
  - Textfeld für 2 URLs
    - für File mit RDF-Part
    - für File mit F-Logic Regelbeschreibungen
  - Textfeld für F-Logic Query
- Ablauf:
  - File at URLs werden gecrawled
  - `<rdf:RDF> ... <rdf:RDF>` Teil wird herausgeschnitten
  - F-Logic Query wird mit gecrawlten Regeln, und gecrawltem RDF-Teil an SiLRI übergeben
  - Antwort wird als Text ausgegeben