

Semantic Web Technologies

Introduction and RDF

Jos de Bruijn
jos.debruijn@deri.org

KRDB Research Group
Free University of Bolzano, Italy

February 20, 2007

1/58

Outline

Organization

Semantic Web

Limitations of the Web
Machine-processable data

Ontologies

Resource Description Framework

RDF Concepts
RDF Containers and Collections
RDF/XML Syntax

2/58

Organization

- ▶ Lecture with labs
- ▶ Lecture
 - ▶ Wednesdays 08:30 – 10:30
 - ▶ Exception: February 28, April 25
- ▶ Labs **bi-weekly**
 - ▶ Wednesdays 17:00 – 19:00
 - ▶ Dates: Mar 7, 14, 28, Apr 11, May 2, 16, 23
- ▶ Structure
 - ▶ 45 mins lecture
 - ▶ 15 mins break
 - ▶ 45 mins lecture
- ▶ Exams
 1. June 2007
 2. September 2007
 3. February 2008
- ▶ Web site: <http://www.debruijn.net/teaching/swt/>

4/58

Organization (cont'd)

- ▶ Questions about the lecture or exercises can be asked after the lecture or by email: jos.debruijn@deri.org
- ▶ Appointments can be made via email (only for Wednesdays)
- ▶ Solutions to labs to be sent to jos.debruijn@deri.org by following Tuesday
- ▶ Grading
 - ▶ Grade: $\max\{exam, 0.66 \times exam + 0.34 \times lab\}$
 - ▶ Labs are required(?) to be completed to get a grade
- ▶ Questions?

5/58

Course Material

- ▶ Presentation slides
- ▶ Additional reading on the slides and the Web site.
- ▶ (Grigori Antoniou, Frank van Harmelen: **A Semantic Web Primer**, MIT Press, USA, 2004. ISBN: 0-262-01210-3.)

6/58

Your Background

- ▶ First-Order Predicate Logic
- ▶ Description Logics
- ▶ Logic Programming
- ▶ Web Technology
- ▶ Web Services

7/58

The Web

- ▶ What is the Web?
 - ▶ HTTP (how to transfer data)
GET /index.html
 - ▶ URI (how to address data)
http://www.deri.org/
 - ▶ HTML (how to mark up data for human reader)
<html><head><title>.....

10/58

The Problem with the Web

- Billions of diverse documents online; problems in:
- ▶ Retrieving documents
 - ▶ Extracting relevant data from retrieved documents
 - ▶ Combining information from different sources to achieve a particular goal

11/58

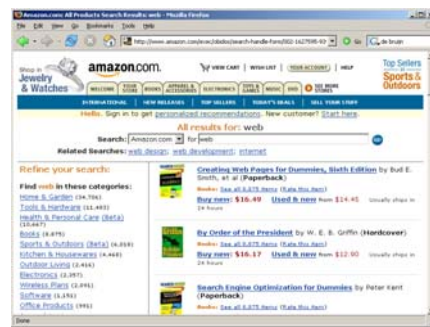
Retrieving documents



Where is Jos?

12/58

Extracting Information



13/58

Extracting Information



Which book is about the Web?

14/58

Extracting Information

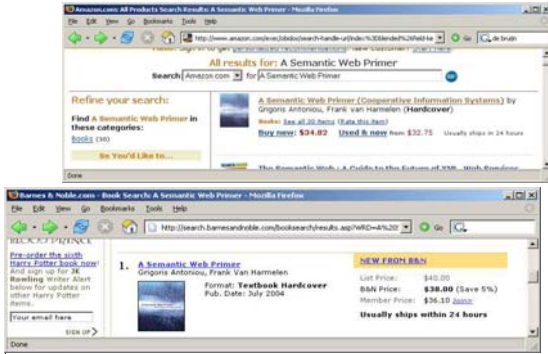


What is the price of the book?

15/58

Combining information

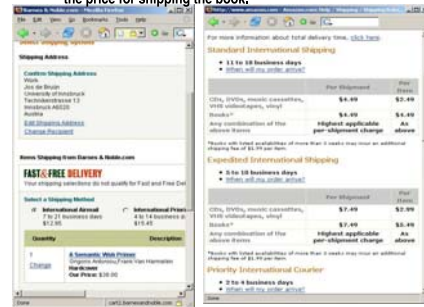
I want the cheapest copy of "A Semantic Web Primer".



16/58

Combining information (cont'd)

I want the cheapest copy of "A Semantic Web Primer"; taking into account the price for shipping the book.



On average 10 clicks to find out what the shipping rate is!

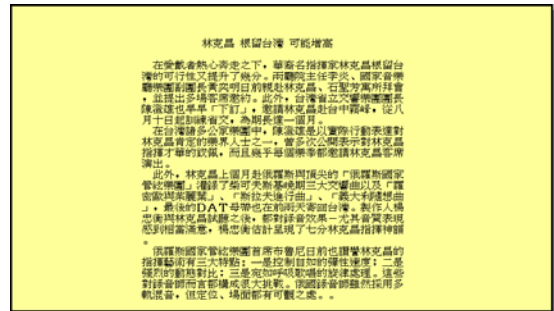
17/58

The solution

- ▶ Instead of publishing natural language, publish machine-processable data!
- ▶ Publish information in terms understandable for a machine
- ▶ Ask questions in terms understandable for a machine
- ▶ And: make sure all machines understand your terms!

19/58

What it's like to be a machine on the Web



Slide originally presented by Frank van Harmelen
<http://www.cs.vu.nl/X/EFrankh/spool/SemWebSlides/SemWeb-tour-Brussels.ppt>

20/58

Publishing and querying machine processable data

- ▶ Publishing:
 - ▶ B related-to A
 - ▶ C related-to A
 - ▶ D related-to C
- ▶ Querying (give me all things related to A):
 - ▶ ?x related-to A
- ▶ Answer:
 - ▶ ?x = B
 - ▶ ?x = C

21/58

Publishing and querying machine processable data (cont'd)

- ▶ Publishing (related-to is transitive):
 - ▶ B related-to A
 - ▶ C related-to A
 - ▶ D related-to C
 - ▶ ?x related-to ?y and ?y related-to ?z ⇒ ?x related-to ?z
- ▶ Querying (give me all things related to A):
 - ▶ ?x related-to A
- ▶ Answer:
 - ▶ ?x = B
 - ▶ ?x = C
 - ▶ ?x = D

22/58

Course Overview

- ▶ Ontologies
- ▶ Resource Description Framework (RDF)
- ▶ A query language for RDF: SPARQL
- ▶ Lightweight ontologies: RDF Schema
- ▶ Web Ontology Language (OWL)
- ▶ Ontologies and Rules in F-Logic
- ▶ Exchange of rules over the Web: RuleML
- ▶ Semantic Web Services

23/58

Ontologies

- ▶ **Formal**
- ▶ **explicit** specification of a
- ▶ **shared**
- ▶ **conceptualization**
- ▶ of a **domain**.

25/58

Ontologies I: Formal Ontologies

- ▶ Meaning of ontology is **unambiguous**;
- ▶ avoids **misunderstanding**;
- ▶ specification using **formal language**;
- ▶ enables **reasoning**: making implicit information explicit;
- ▶ Hampers **consensus**

26/58

Ontologies II: Explicit specifications

- ▶ Make domain assumptions **explicit**
 - ▶ for **reasoning**,
 - ▶ for **clarifying** understanding of domain.
- ▶ **Minimal ontological commitment**
 - ▶ Too much explicit ⇒ **no consensus**
 - ▶ Too little explicit ⇒ ontology **unusable**
 - ▶ Minimal ontological commitment = “make as little as explicit as possible, while keeping ontology useful”.

27/58

Ontologies III: Specification of a shared conceptualization of a domain

- ▶ Domain: specific part of the world
- ▶ Conceptualization
 - ▶ Forming **idea** of domain
 - ▶ in the **minds** of people
- ▶ **Shared** among its users
 - ▶ facilitates **accepting** the ontology.

28/58

Classes, Properties, Instances, Axioms

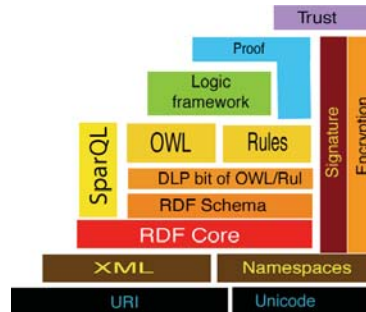
- ▶ Classes
 - ▶ Grouping of individuals
 - ▶ with common properties
 - ▶ cf. UML
 - ▶ **Intentional** classes
 - ▶ e.g. Persons, Cars, Universities, ...
- ▶ Relations
 - ▶ Connections **between** individuals
 - ▶ May be **attached** to classes
 - ▶ e.g. hasName, hasChild, hasColor, owns, ...
- ▶ Individuals
 - ▶ **Objects** in the domain
 - ▶ May be **instances** of classes
- ▶ Axioms
 - ▶ Additional statements about the domain
 - ▶ Specified in logical language
 - ▶ e.g. “hasName has one value”

29/58

Ontologies

- ▶ Form the **backbone** of the Semantic Web,
- ▶ define the **basic vocabulary** for the annotations,
- ▶ enable reasoning with **background knowledge**,
- ▶ based on **formal** languages,
- ▶ interweave meaning for
 - ▶ **humans** and
 - ▶ **machines**,
- ▶ are **shared**.

Semantic Web Layer Cake

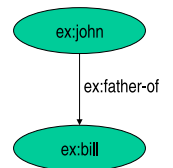


Principles of RDF

- ▶ Resources (identified by URIs)
 - ▶ A URI **identifies** a resources, but does not necessarily **point** to it
 - ▶ Correspond to nodes in a graph
 - ▶ E.g. `http://www.w3.org/`, `http://example.org/#john`, `http://www.w3.org/1999/02/22-rdf-syntax-ns#Property`
- ▶ Properties (identified by URIs)
 - ▶ Correspond to labels of edges in a graph
 - ▶ Binary relation between two resources
 - ▶ E.g. `http://www.example.org/#hasName`, `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`
- ▶ Literals
 - ▶ concrete data values
 - ▶ E.g. "John Smith", "1", "2006-03-07"

Principles of RDF (cont'd)

- ▶ Triple data model:
(*subject, predicate, object*)
 - ▶ Subject: Resource or blank node
 - ▶ Predicate: Property
 - ▶ Object: Resource, literal or blank node
 - ▶ Example:
(`ex:john`, `ex:father-of`, `ex:bill`)
- ▶ Labeled, directed graphs
 - ▶ Nodes: resources, literals
 - ▶ Labels: properties
 - ▶ Edges: statements



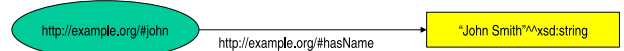
Resources

- ▶ A resource may be:
 - ▶ Web page (e.g. `http://www.w3.org/`)
 - ▶ A person (e.g. `http://www.debruijn.net/`)
 - ▶ A book (e.g. `urn:isbn:0-345-33971-1`)
 - ▶ Anything denoted with a URI!
- ▶ A URI is an **identifier** and **not** a location on the Web
- ▶ RDF allows making statements about resources (meta-data):
 - ▶ `http://www.w3.org/` **has the format** `text/html`
 - ▶ `http://www.debruijn.net/` **has first name** `Jos`
 - ▶ `urn:isbn:0-345-33971-1` **has the author** `Tolkien`

Literals

- ▶ Plain literals
 - ▶ E.g. "blabla"
 - ▶ Optional language tag, e.g. "Hello, how are you?"@en-GB
- ▶ Typed literals
 - ▶ E.g. "hello"^^xsd:string, "1"^^xsd:integer
 - ▶ Recommended datatypes: XML Schema datatypes
 - ▶ Datatype mechanism extensible
 - ▶ Type checking not in RDF
- ▶ Only as **object** of a triple, e.g.:

<`http://example.org/#john`>
<`http://example.org/#hasName`>, "John Smith"^^xsd:string)



Datatypes

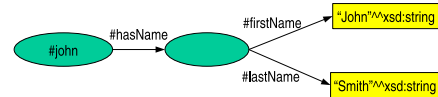
- ▶ One pre-defined datatype: rdf:XMLLiteral
- ▶ Recommended datatypes are XML Schema datatypes, e.g.:
 - ▶ xsd:string
 - ▶ xsd:integer
 - ▶ xsd:float
 - ▶ xsd:anyURI
 - ▶ xsd:boolean
- ▶ A datatype has (example is for xsd:boolean):
 - ▶ Lexical space, e.g. {"0", "1", "true", "false"}
 - ▶ Value space, e.g. {T, F}
 - ▶ Lexical-to-value mapping, e.g. {"true", T}, {"1", T}, {"0", F}, {"false", F}

38/58

Blank nodes

- ▶ Blank nodes are nodes **without** a URI
 - ▶ Unnamed resources
 - ▶ More complex constructs
- ▶ Representation of blank nodes is **syntax-dependent**
- ▶ For example:

```
<#john>, <#hasName>, ..:johnsname>
<.:johnsname, <#firstName>, "John"^^xsd:string>
<.:johnsname, <#lastName>, "Smith"^^xsd:string>
```



39/58

Reification

- ▶ Reification: statements about statements

Mary claims that John's name is "John Smith".

```
<#myStatement>, rdf:type, rdf:Statement>
<#myStatement>, rdf:subject, <#john>>
<#myStatement>, rdf:predicate, <#hasName>>
<#myStatement>, rdf:object, "John Smith">

<#mary>, <#claims>, <#myStatement>>
```

40/58

RDF Vocabulary

- ▶ RDF defines a number of resources and properties
- ▶ We have already seen: rdf:XMLLiteral, rdf:type, ...
- ▶ RDF vocabulary is defined in the namespace:


```
http://www.w3.org/1999/02/22-rdf-syntax-ns#
```
- ▶ Classes:
 - rdf:Property rdf:Statement rdf:XMLLiteral rdf:Seq rdf:Bag
 - rdf:Alt rdf:List
- ▶ Properties:
 - rdf:type rdf:subject rdf:predicate rdf:object rdf:first rdf:rest
 - rdf:_n rdf:value
- ▶ Resources:
 - rdf:nil

41/58

RDF Vocabulary (cont'd)

- ▶ Typing using rdf:type:
 - <A, rdf:type, B>
 - "A belongs to class B"
 - ▶ All properties belong to class rdf:Property:
 - <P, rdf:type, rdf:Property>
 - "P is a property"
- ```
<rdf:type, rdf:type, rdf:Property>
"rdf:type is a property"
```

42/58

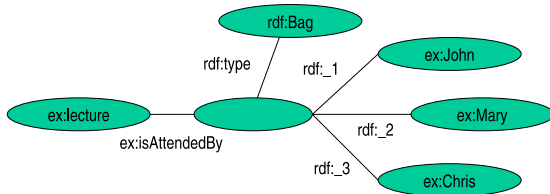
## RDF containers

- ▶ Grouping property values:
  - "The lecture is attended by John, Mary and Chris" Bag
  - "[RDF-Concepts] is edited by Graham and Jeremy (in that order)" Seq
  - "The source code for the application may be found at ftp1.example.org, ftp2.example.org, ftp3.example.org" Alt

44/58

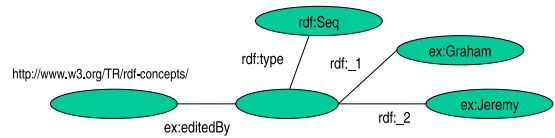
## RDF Containers: Bag

"The lecture is attended by John, Mary and Chris"



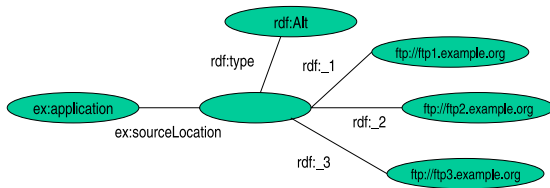
## RDF Containers: Seq

"[RDF-Concepts] is edited by Graham and Jeremy (in that order)"



## RDF Containers: Alt

"The source code for the application may be found at ftp1.example.org, ftp2.example.org, ftp3.example.org"

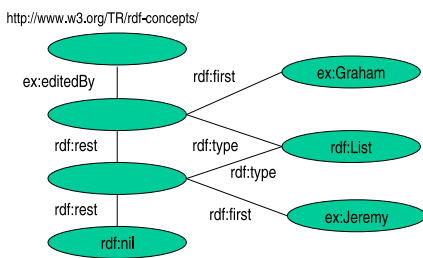


## RDF Containers

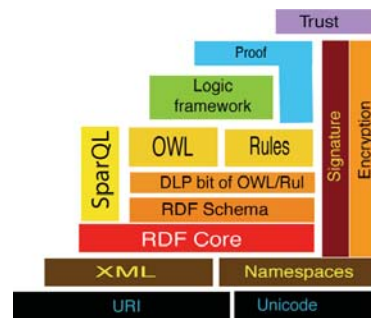
- ▶ Three types of containers:
  - ▶ `rdf:Bag` – unordered set of items
  - ▶ `rdf:Seq` – ordered set of items
  - ▶ `rdf:Alt` – set of alternatives
- ▶ Every container has a triple declaring the `rdf:type`
- ▶ Items in the container are denoted with `rdf:_1`, `rdf:_2`, ..., `rdf:_n`
- ▶ Limitations:
  - ▶ Semantics of the container is up to the application
  - ▶ What about closed sets? How do we know whether Graham and Jeremy are the only editors of [RDF-Concepts]?

## RDF Collections

"[RDF-Concepts] is edited by Graham and Jeremy (in that order) and **nobody else**"



## Semantic Web Layer Cake



## URIs and Namespaces

- ▶ Uniform Resource Identifier (URI):
  - ▶ Globally unique identifiers for resources
  - ▶ Not necessarily resources on the Web (e.g. 'urn:isbn:')
  - ▶ Thus: it is possible to make claims about the same resource in different locations!
- ▶ Namespaces: Syntactical space of XML names:
  - ▶ `< namespace, localname >`
  - ▶ Usually abbreviated using namespace prefixes: `prefix:localname`, e.g. `xs:integer`
  - ▶ Reused in RDF, but merely as abbreviations: `xs:integer` is equivalent to `http://www.w3.org/2001/XMLSchema#integer` and **not** `<"http://www.w3.org/2001/XMLSchema#", "integer">`

For more on XML, see Semantic Web Primer, Chapter 2

## RDF/XML

- ▶ Serializing RDF for the Web
  - ▶ XML as standardized interchange format:
    - ▶ Namespaces (e.g. `rdf:type`, `xsd:integer`, `ex:john`)
    - ▶ Encoding (e.g. UTF8, iso-8859-1)
    - ▶ XML Schema (e.g. datatypes)
    - ▶ DTD entities (e.g. `&xsd;`)
  - ▶ Reuse of existing XML tools:
    - ▶ Syntax checking (i.e. schema validation)
    - ▶ Transformation (via XSLT)
      - ▶ Different RDF representation
      - ▶ Layout (XHTML)
      - ▶ Different XML-based format
- ▶ Parsing and in-memory representation/manipulation (DOM/SAX)
- ▶ ...

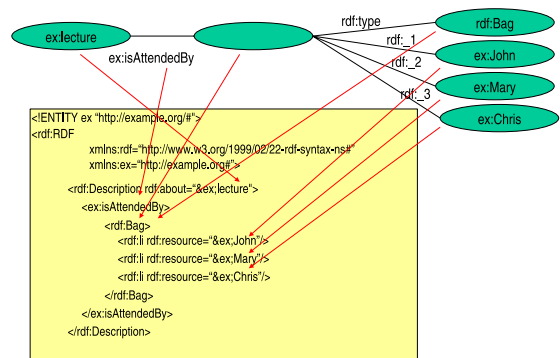
## RDF/XML (cont'd)

```
<#john,#hasName,"John">
<#john,#marriedTo,#mary>
```

```
<!ENTITY ex "http://example.org/#">
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:ex="http://example.org/#">
 <rdf:Description rdf:about="http://example.org/#john">
 <ex:hasName>John</ex:hasName>
 <ex:marriedTo rdf:resource="&ex;mary"/>
 </rdf:Description>
</rdf:RDF>
```

} Head  
} Body  
} Foot

## RDF/XML (cont'd)



## Conclusion

- ▶ Advantages:
  - ▶ Reuse existing standards/tools
  - ▶ Provides some structure for free (e.g. for containers)
  - ▶ Standard format
- ▶ Disadvantages:
  - ▶ Verbose
  - ▶ Reconstructing RDF graph non-trivial

## Summary

- Organization
- Semantic Web
  - Limitations of the Web
  - Machine-processable data
- Ontologies
- Resource Description Framework
  - RDF Concepts
  - RDF Containers and Collections
  - RDF/XML Syntax

### Required reading

- ▶ Ontology Development 101: [http://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html)
- ▶ RDF Primer: <http://www.w3.org/TR/rdf-primer/>, Chapters 1-4

### Further reading

- ▶ Semantic Web Primer, Chapters 1,2 and Sections 3.1, 3.2, 3.3.
- ▶ Berners-Lee, T.; Hendler, J.; Lassila, O.: The Semantic Web. Scientific American, May 2001. <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>
- ▶ Jos de Bruijn: **Using Ontologies**. <http://www.deri.org/publications/techpapers/documents/DERI-TR-2003-10-29.pdf>
- ▶ RDF Concepts and abstract syntax: <http://www.w3.org/TR/rdf-concepts/>
- ▶ RDF/XML syntax specification: <http://www.w3.org/TR/rdf-syntax-grammar/>