# Advanced Studies in IT
# CT433

Making the Web Machine- Readable:
The Semantic Web:

Lecture 4

www.deri.ie

science foundation ireland
fondúireacht eolaíochta éireann

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh

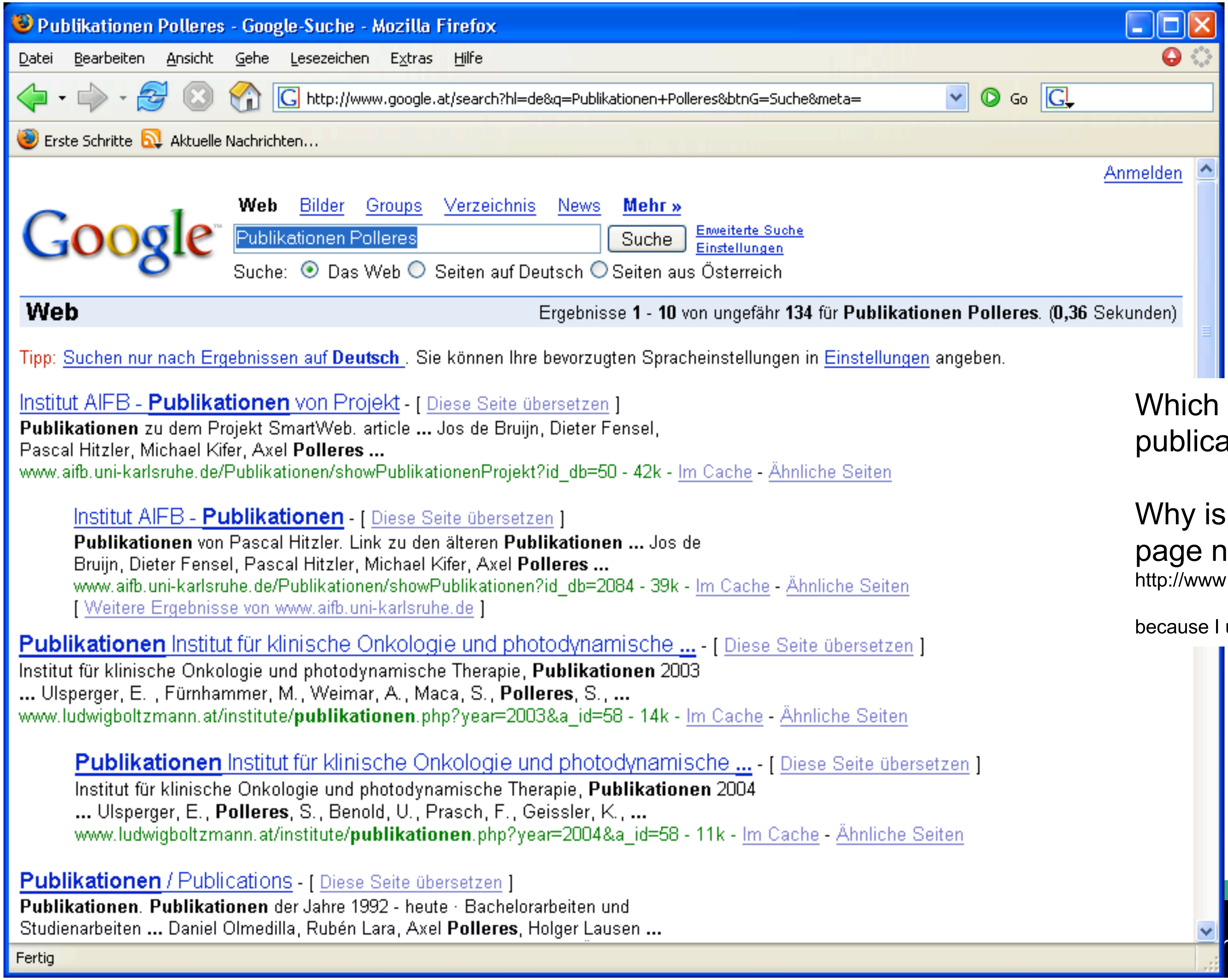
# Overview

- **What is the Semantic Web?**
- RDF
  - RDF/XML
  - RDF Turtle
- SPARQL - A Query language for the Semantic Web
- What’s next?
  - RDFS + OWL
  - Next Time: Outlook/Applications/Review of Assignments

# What is the „Semantic Web“?

- What is the „Web“?
  - Web =
    - HTTP (How to transfer data)
    - URI (How to address data?)
    - HTML (How to layout data?)
- The problem:
  - Millions of different documents online:
    - How to find the right documents?
    - How to extract relevant information?
    - How to combine information from different sources?
    - Why do I have to do this at all? Why doesn't my computer do it for me? Why can't I just query the Web like a database?

# How to find the right documents?


Publikationen Polleres - Google-Suche - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
Go
Erste Schritte
Aktuelle Nachrichten...
Anmelden
Google
Web Bilder Groups Verzeichnis News Mehr »
Suche
Erweiterte Suche
Einstellungen
Fertig

http://www.google.at/search?hl=de&q=Publikationen+Polleres&btnG=Suche&meta=

Publikationen Polleres

Suche: ● Das Web ○ Seiten auf Deutsch ○ Seiten aus Österreich

**Web** Ergebnisse **1** - **10** von ungefähr **134** für **Publikationen Polleres**. (**0,36** Sekunden)

Tipp: Suchen nur nach Ergebnissen auf **Deutsch**. Sie können Ihre bevorzugten Spracheinstellungen in Einstellungen angeben.

Institut AIFB - **Publikationen** von Projekt - [ Diese Seite übersetzen ]
**Publikationen** zu dem Projekt SmartWeb. article **...** Jos de Bruijn, Dieter Fensel, Pascal Hitzler, Michael Kifer, Axel **Polleres** **...**
www.aifb.uni-karlsruhe.de/Publikationen/showPublikationenProjekt?id_db=50 - 42k - Im Cache - Ähnliche Seiten

Institut AIFB - **Publikationen** - [ Diese Seite übersetzen ]
**Publikationen** von Pascal Hitzler. Link zu den älteren **Publikationen** **...** Jos de Bruijn, Dieter Fensel, Pascal Hitzler, Michael Kifer, Axel **Polleres** **...**
www.aifb.uni-karlsruhe.de/Publikationen/showPublikationen?id_db=2084 - 39k - Im Cache - Ähnliche Seiten
[ Weitere Ergebnisse von www.aifb.uni-karlsruhe.de ]

**Publikationen** Institut für klinische Onkologie und photodynamische ... - [ Diese Seite übersetzen ]
Institut für klinische Onkologie und photodynamische Therapie, **Publikationen** 2003 **...** Ulsperger, E. , Fürnhammer, M., Weimar, A., Maca, S., **Polleres**, S., **...**
www.ludwigboltzmann.at/institute/**publikationen**.php?year=2003&a_id=58 - 14k - Im Cache - Ähnliche Seiten

**Publikationen** Institut für klinische Onkologie und photodynamische ... - [ Diese Seite übersetzen ]
Institut für klinische Onkologie und photodynamische Therapie, **Publikationen** 2004 **...** Ulsperger, E., **Polleres**, S., Benold, U., Prasch, F., Geissler, K., **...**
www.ludwigboltzmann.at/institute/**publikationen**.php?year=2004&a_id=58 - 11k - Im Cache - Ähnliche Seiten

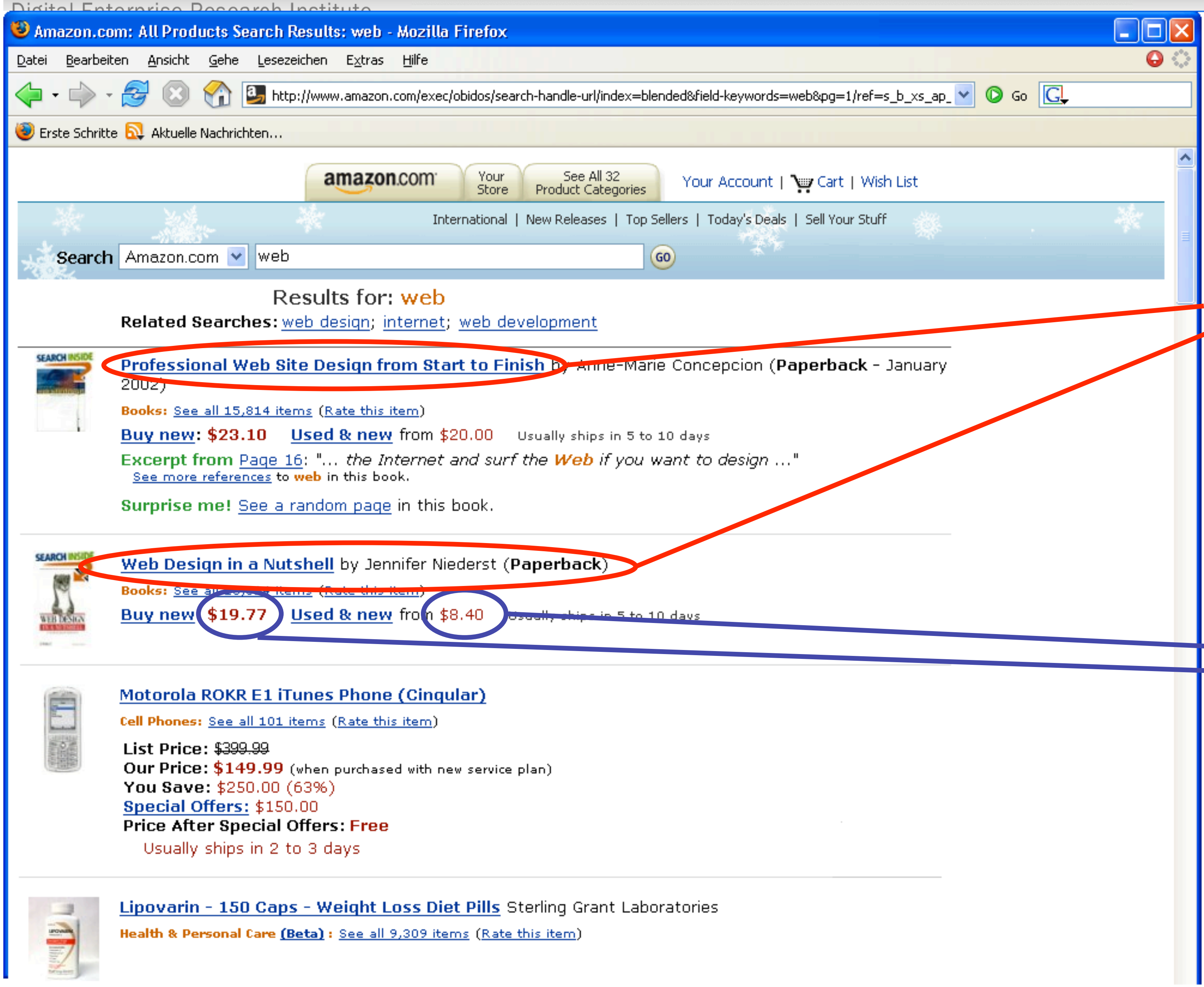
**Publikationen** / Publications - [ Diese Seite übersetzen ]
**Publikationen**. **Publikationen** der Jahre 1992 - heute · Bachelorarbeiten und Studienarbeiten **...** Daniel Olmedilla, Rubén Lara, Axel **Polleres**, Holger Lausen **...**

Which links show publications of Axel?

Why is my publications page not found?
http://www.polleres.net/publications.html

because I used German :-/

# How to extract relevant information?


Amazon.com: All Products Search Results: web - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://www.amazon.com/exec/obidos/search-handle-url/index=blended&field-keywords=web&pg=1/ref=s_b_xs_ap_
Go
Erste Schritte Aktuelle Nachrichten...
amazon.com
Your Store
See All 32 Product Categories
Your Account | Cart | Wish List
International | New Releases | Top Sellers | Today's Deals | Sell Your Stuff
Search Amazon.com web GO
SEARCH INSIDE
SEARCH INSIDE

Results for: web

**Related Searches:** web design; internet; web development

**Professional Web Site Design from Start to Finish** by Anne-Marie Concepcion (**Paperback** - January 2002)

**Books:** See all 15,814 items (Rate this item)

**Buy new: $23.10** **Used & new** from $20.00 Usually ships in 5 to 10 days

**Excerpt from** Page 16: "... *the Internet and surf the* ***Web*** *if you want to design* ..."
See more references to **web** in this book.

**Surprise me!** See a random page in this book.

**Web Design in a Nutshell** by Jennifer Niederst (**Paperback**)

**Books:** See all 15,814 items (Rate this item)

**Buy new: $19.77** **Used & new** from $8.40 Usually ships in 5 to 10 days

**Motorola ROKR E1 iTunes Phone (Cingular)**

**Cell Phones:** See all 101 items (Rate this item)

**List Price:** ~~$399.99~~
**Our Price: $149.99** (when purchased with new service plan)
**You Save:** $250.00 (63%)
**Special Offers:** $150.00
**Price After Special Offers: Free**
Usually ships in 2 to 3 days

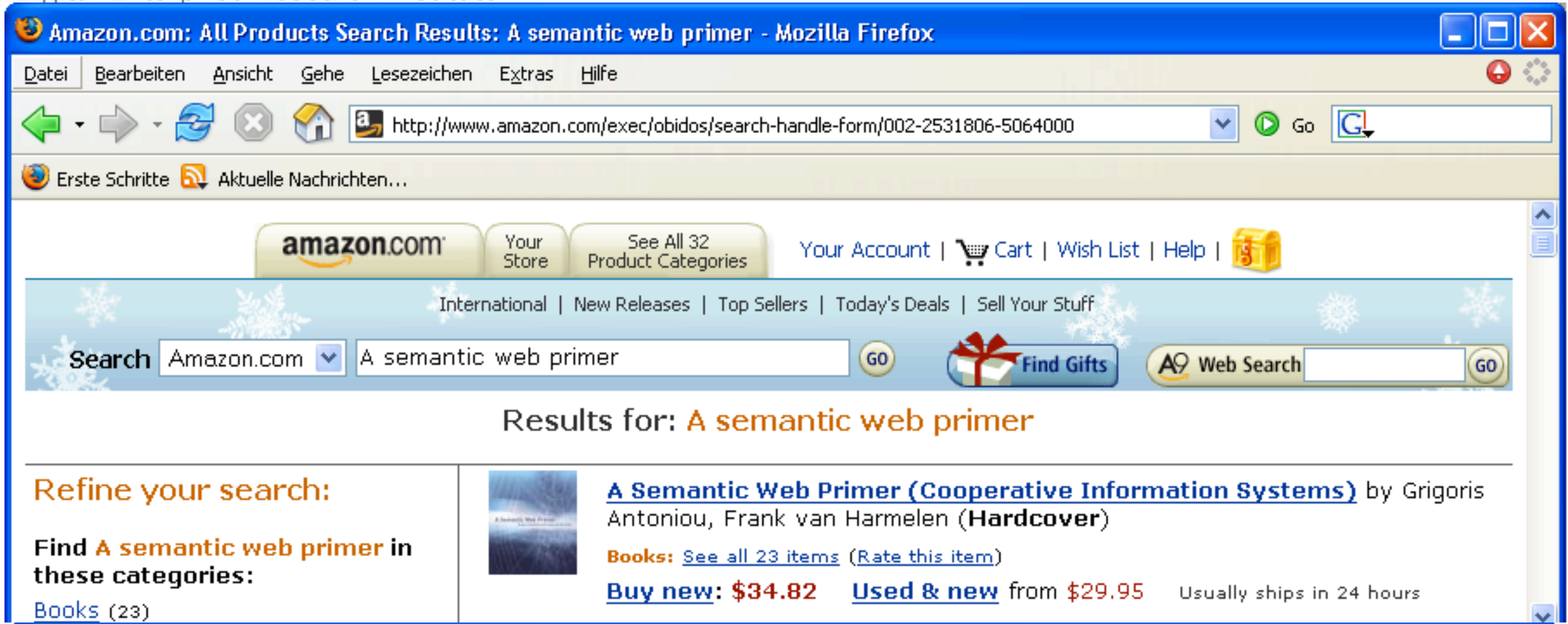
**Lipovarin - 150 Caps - Weight Loss Diet Pills** Sterling Grant Laboratories

**Health & Personal Care (Beta):** See all 9,309 items (Rate this item)

What here is a book about the Web?

What is the price?

# How to combine information from different sources?


Amazon.com: All Products Search Results: A semantic web primer - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://www.amazon.com/exec/obidos/search-handle-form/002-2531806-5064000
Go
Erste Schritte
Aktuelle Nachrichten...
amazon.com
Your Store
See All 32 Product Categories
Your Account | Cart | Wish List | Help |
International | New Releases | Top Sellers | Today's Deals | Sell Your Stuff
Search Amazon.com
A semantic web primer
GO
Find Gifts
Web Search
GO

Results for: A semantic web primer

Refine your search:

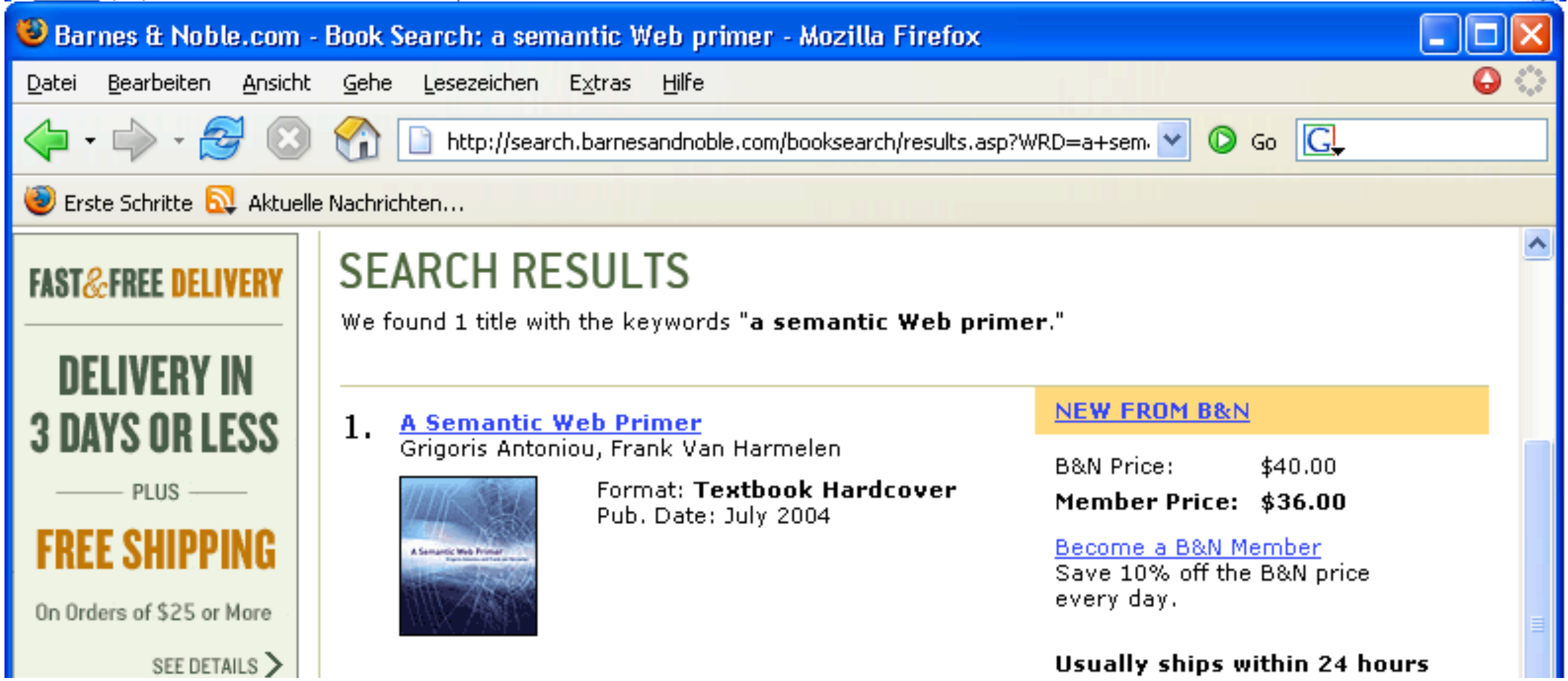
Find A semantic web primer in these categories:

Books (23)

A Semantic Web Primer (Cooperative Information Systems) by Grigoris Antoniou, Frank van Harmelen (**Hardcover**)

**Books:** See all 23 items (Rate this item)

**Buy new: $34.82** Used & new from $29.95 Usually ships in 24 hours


Barnes & Noble.com - Book Search: a semantic Web primer - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://search.barnesandnoble.com/booksearch/results.asp?WRD=a+sem
Go
Erste Schritte
Aktuelle Nachrichten...
FAST&FREE DELIVERY
DELIVERY IN 3 DAYS OR LESS
PLUS
FREE SHIPPING
On Orders of $25 or More
SEE DETAILS

SEARCH RESULTS

We found 1 title with the keywords "**a semantic Web primer**."

1. A Semantic Web Primer
Grigoris Antoniou, Frank Van Harmelen

Format: **Textbook Hardcover**
Pub. Date: July 2004

NEW FROM B&N

B&N Price: $40.00
**Member Price: $36.00**

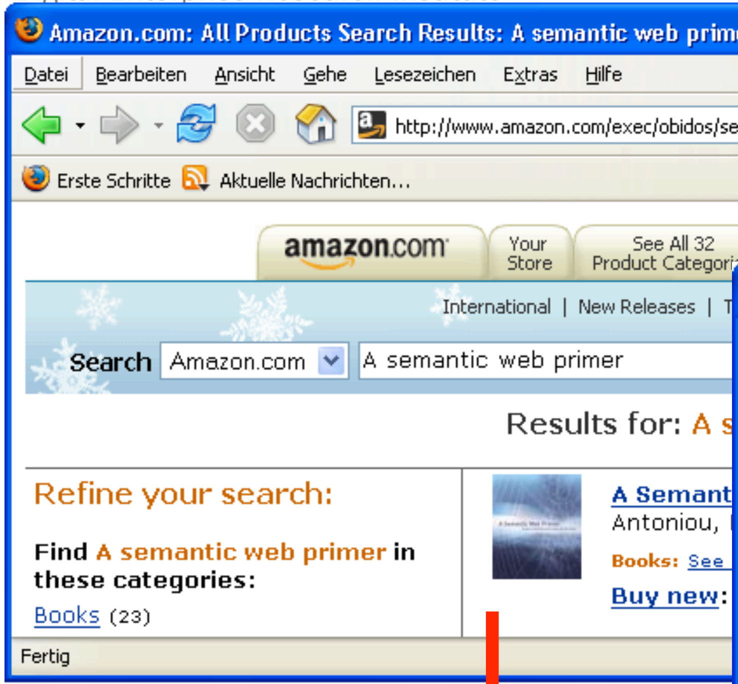
Become a B&N Member
Save 10% off the B&N price every day.

**Usually ships within 24 hours**

I want the cheapest offer for „A Semantic Web Primer“?

# How to combine information from different sources?

Why do I have to do this at all? Why doesn't my computer do it for me?



Amazon.com: All Products Search Results: A semantic web primer - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
Go
Erste Schritte Aktuelle Nachrichten...
amazon.com Your Store See All 32 Product Categori
Your Account | Cart | Wish List | Help |
International | New Releases | T
Search

http://www.amazon.com/exec/obidos/search-handle-form/002-2531806-5064000

Search: Amazon.com | A semantic web primer

Results for: A s

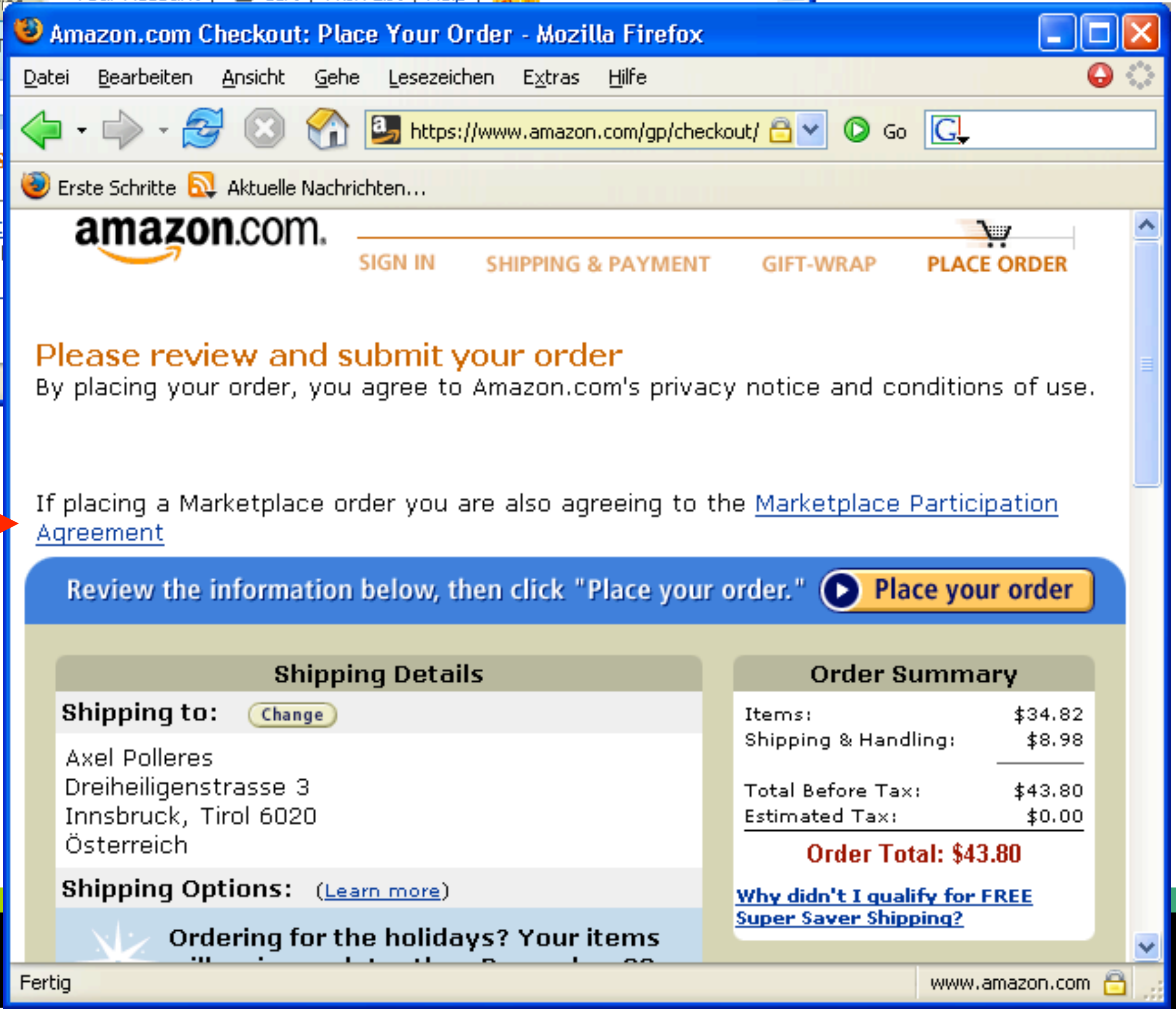
Refine your search:

Find A semantic web primer in these categories:

Books (23)

A Semant
Antoniou,
Books: See
Buy new:

Fertig



Amazon.com Checkout: Place Your Order - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
Go
Erste Schritte Aktuelle Nachrichten...
amazon.com
SIGN IN
SHIPPING & PAYMENT
GIFT-WRAP
PLACE ORDER

https://www.amazon.com/gp/checkout/

Please review and submit your order

By placing your order, you agree to Amazon.com's privacy notice and conditions of use.

If placing a Marketplace order you are also agreeing to the Marketplace Participation Agreement

Review the information below, then click "Place your order." Place your order

Shipping Details

Shipping to: Change

Axel Polleres
Dreiheiligenstrasse 3
Innsbruck, Tirol 6020
Österreich

Shipping Options: (Learn more)

Ordering for the holidays? Your items

Order Summary

| | |
|---|---|
| Items: | $34.82 |
| Shipping & Handling: | $8.98 |
| Total Before Tax: | $43.80 |
| Estimated Tax: | $0.00 |

Order Total: $43.80

Why didn't I qualify for FREE Super Saver Shipping?

Fertig www.amazon.com

I want the cheapest offer for „A Semantic Web Primer“ including shipping costs?

- 10 interaction „clicks“ until I get to the actual shipping costs... Services!

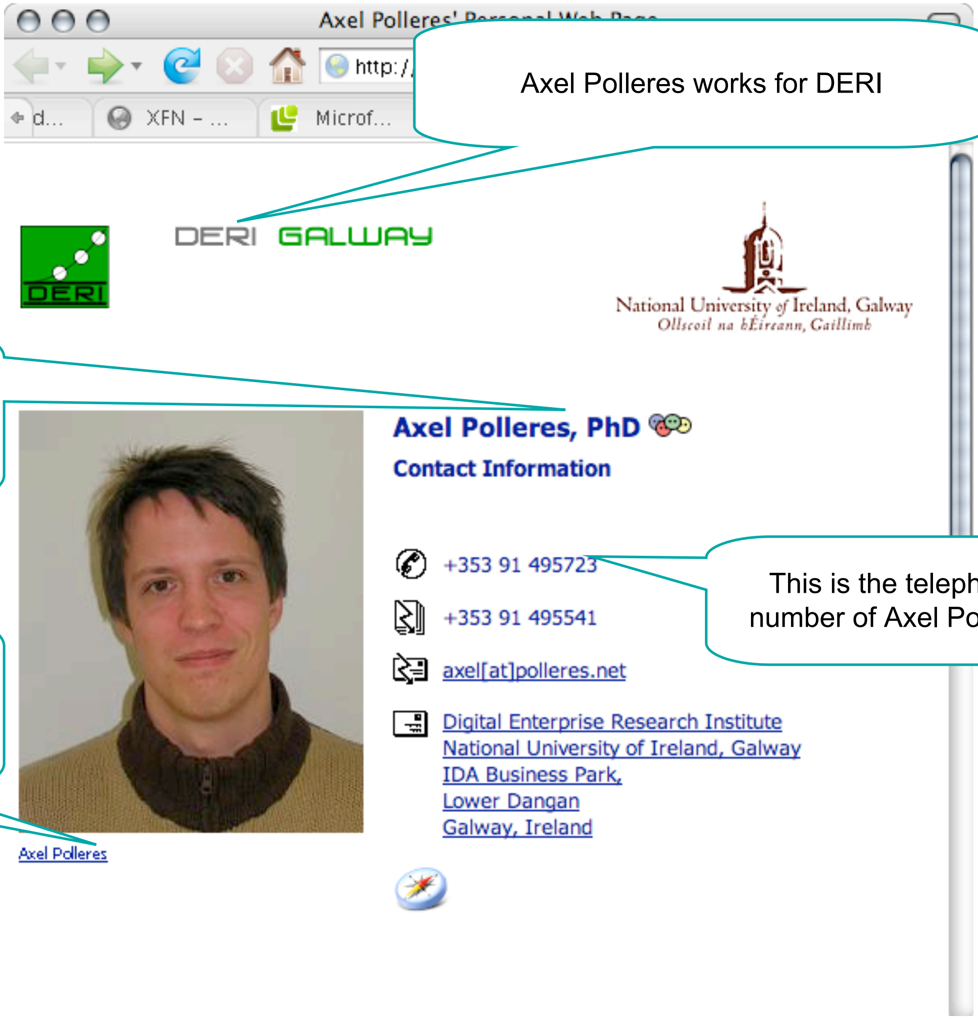
# Semantic Web: The idea

- The Web (HTML) presents knowledge in
  - natural language (Spanish, German, English, Hungarian, Chinese,…)
  - Grafics, Multimedia, Layout
- Humans can (combining these pages with their **background knowledge**):
  - deduce facts facts from incomplete information
  - draw connections, detect similarities
- But…
  - Computers cannot do this:
    - incomplete information is useless
    - drawing connections only possible if rules are avaialable as a program
    - Problems with combining information:
      - » Is <foo:creator> the same as <bar:author>?
      - » How do I combine data from differentHTML docs or XML Trees?

→ Idea: Make machine understandable meta-information explicit!!! Effect: If everybody does that, the Web can be uses as a huge database!

# Data and Meta-data:

**Data:**



Axel Polleres' Personal Web Page

http://

d...

XFN - ...

Microf...

DERI GALWAY

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh

Axel Polleres

**Axel Polleres, PhD**

**Contact Information**

+353 91 495723

+353 91 495541

axel[at]polleres.net

Digital Enterprise Research Institute
National University of Ireland, Galway
IDA Business Park,
Lower Dangan
Galway, Ireland

Axel Polleres works for DERI

Axel Polleres is a researcher

This is the telephone number of Axel Polleres

This page was created by Axel Polleres

**Meta-Data:**

Researchers working at DERI know about the Semantic Web.

If someone knows about the Semantic Web, he/she also knows about RDF.

**A possible query to the Semantic Web:**

We look for someone who knows about RDF

**We need the Data and Meta-Data available in machine-readable form!**

# What is the „Semantic Web“?

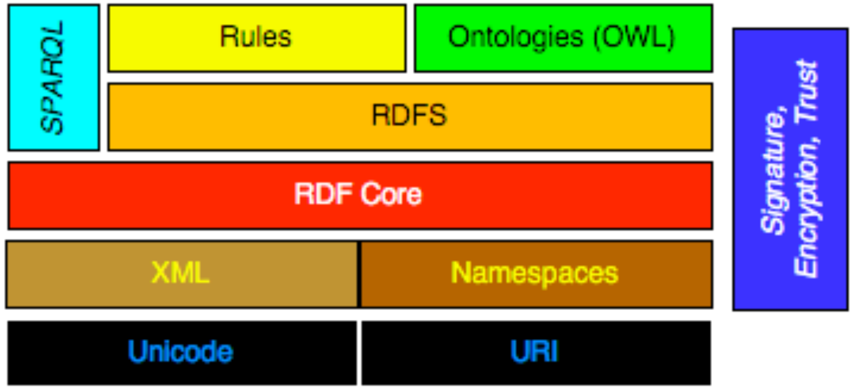
- Semantic Web = Web plus machine-readable "meaning".
- *"If HTML and the Web made all the online documents look like one huge **book**, RDF, schema and inference languages will make all the data in the world look like one huge **database**„*
  - *Tim Berners-Lee, Weaving the Web, 1999*
- *„The "Semantic Web" is  a meta-data based infrastructure to draw conclusions from the Web. It **extends** the Web but will not replace it.“*
  - Klaus Birkenbihl, W3C German-Austrian office
- That means:
  - Machine readable annotations, which describes the content of Websites, Databases, Libraries by **standardised vocabularies** (RDF)
  - Ontologies, i.e. Meta-Data, which describes the structure and connections of this Data (RDFS/OWL)
  - Intelligent inference machines on top (something like a "database engine for Meta-data")

# What is the „Semantic Web“ not (yet)?

- “Artificial Intelligence for the Web”
  - Although it uses logics and inference techniques…
  - … more down-to-earth
  - It deals with proper representation, description and processing of metadata
  - But: AI techniques *can use* SW metadata! (next step)

- “A purely academic research area”
  - Many applications already exist (see examples later)
  - „big players“ in industry use Semantic Web Technologies (Oracle, Sun, Adobe, HP, IBM, Nokia, Vodaphone, …)
  - E.g. EU-Projects with lots of industrial partners (SAP, IBM, BT, etc.), or
  - Google starting to crawl and use RDF (Social Networks API, etc.)

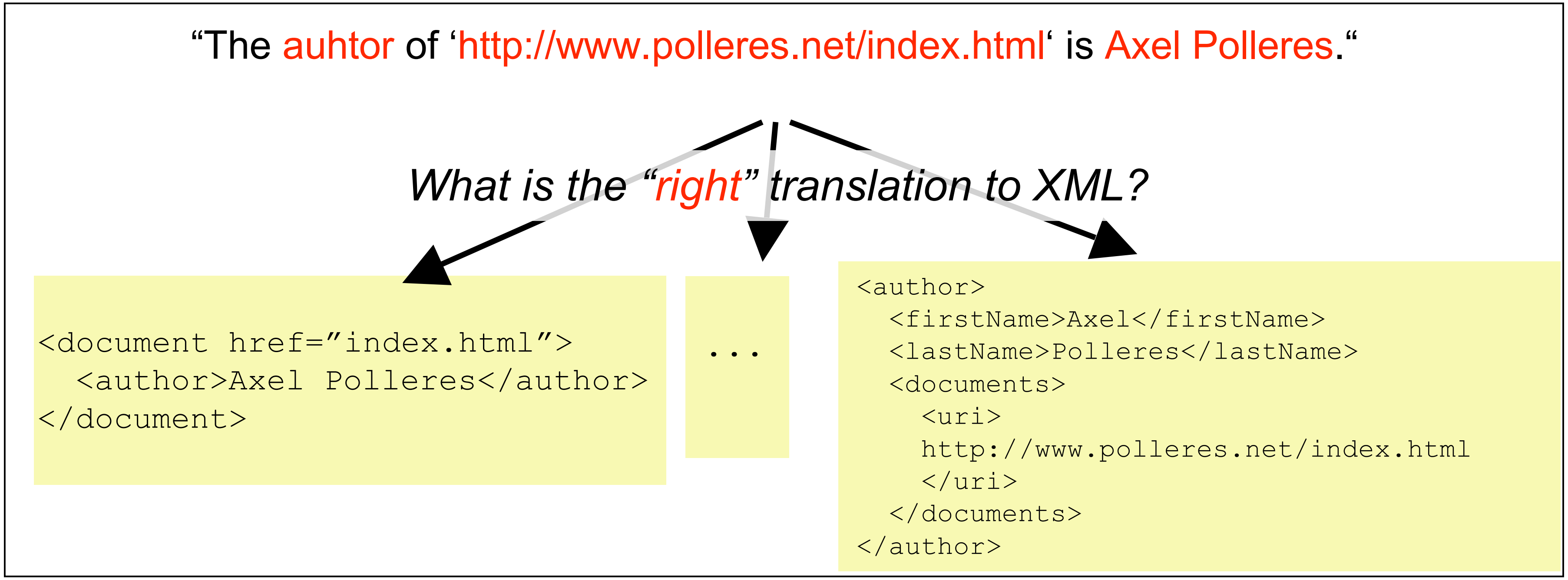
# Semantic Web Architecture:



- A “layered” architecture of standards, developed in various working groups of the W3C.

# Why are XML/XPath/XSLT not enough?

- In XML, metadata can be expressed by element and attribute names... what more would I need?
- Problem:
  - XML is ambiguous
  - XML doesn't prescribe an agreed convention for how to describe data.
  - Parsing depends on the structure (expressed in DTD or XML Schema)
  - I cannot describe meta information (e.g. every <foo:creator> is a <bar:author>)



“The auhtor of ‘http://www.polleres.net/index.html‘ is Axel Polleres.“

*What is the “right” translation to XML?*

```
<document href=”index.html”>
  <author>Axel Polleres</author>
</document>
```

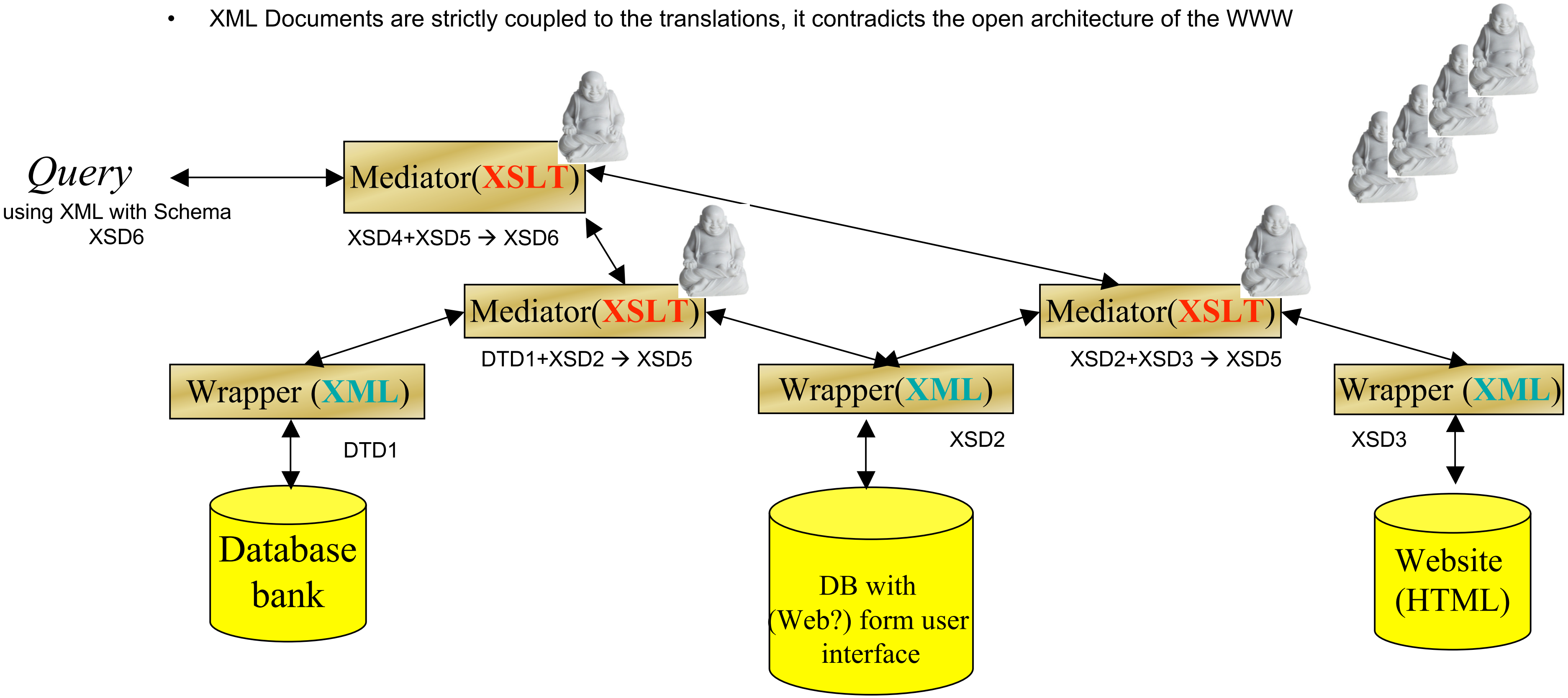
...

```
<author>
  <firstName>Axel</firstName>
  <lastName>Polleres</lastName>
  <documents>
    <uri>
    http://www.polleres.net/index.html
    </uri>
  </documents>
</author>
```

How could a semantic Web based on pure XML look like?

## Possible "engineering" solution for the Web integration problem:

- Develop wrappers and mediators, using XML+XSLT
- ie. wrappers and XSLT-mediators for ANY POSSIBLE combination of data sources!
- Advantage:
  - unified format (XML)
  - simple language to describe transformations (XSLT)
- Disadvantage:
  - Doesn't scale to the Web!.
  - XML Documents are strictly coupled to the translations, it contradicts the open architecture of the WWW



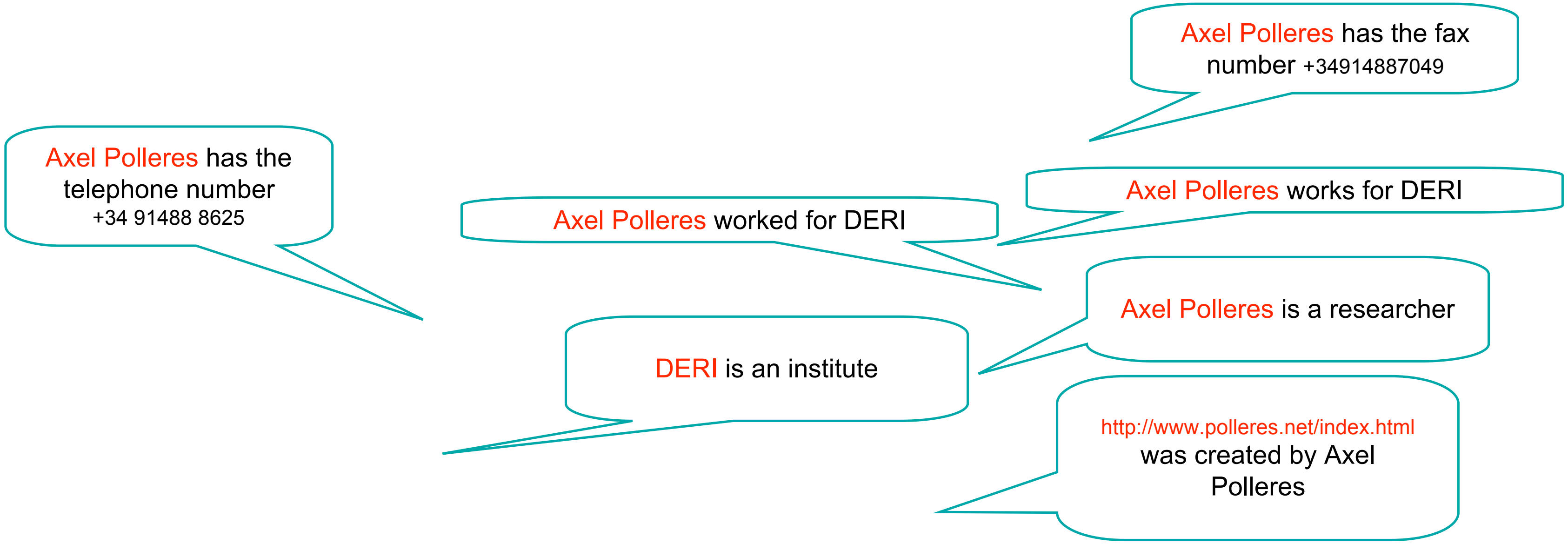
# Overview

- What is the Semantic Web?
- **RDF**
  - RDF/XML
  - RDF Turtle
- SPARQL - A Query language for the Semantic Web
- What's next?
  - RDFS + OWL
  - Next Time: Outlook/Applications/Review of Assignments

# RDF – The Resource Description Framework

- Allows description of „Resources“
- A resource is not necessarily a web accessible document, but could also be e.g. a person, a company, a service, etc.
- Meta-data in RDF consists of a set of „**statements**“
- Each statement is about a **resource**

- Our example from before: meta-data to describe a website, person, company, etc:



Axel Polleres has the fax number +34914887049

Axel Polleres has the telephone number +34 91488 8625

Axel Polleres works for DERI

Axel Polleres worked for DERI

Axel Polleres is a researcher

DERI is an institute

http://www.polleres.net/index.html was created by Axel Polleres

# RDF Basics

- Statements can (mathematically) be expressed by triples:

  Notation: **s p o.**

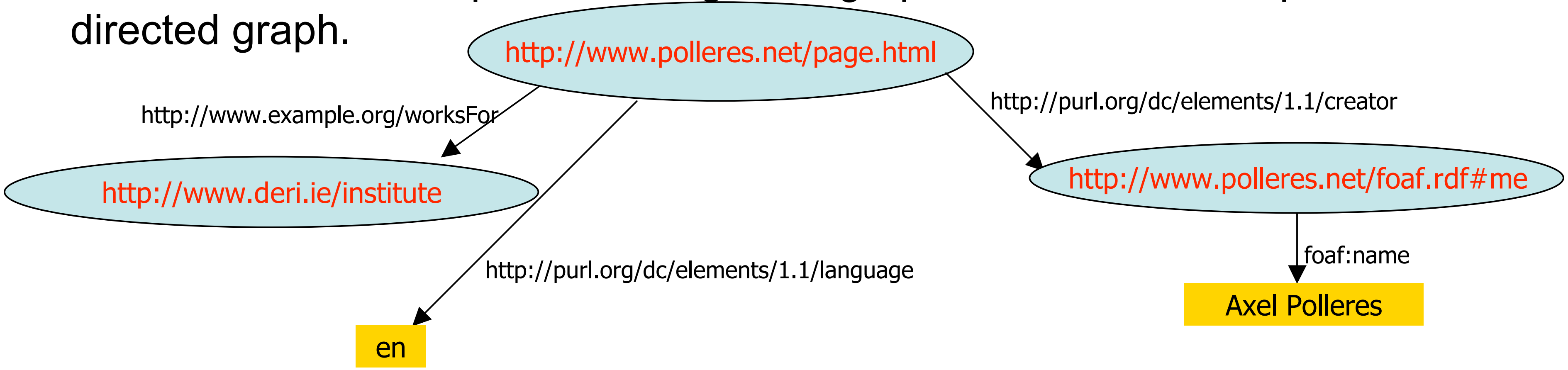
  für Subject, Predicat,e Object, e.g.

  AxelPolleres worksFor DERI.

  You can also think of a triple as an attribute/value pair for a resource

- *RDF* is a general model for statements:
  - … with machine-readable syntax (e.g., RDF/XML, Turtle, n3, …)
  - RDF/XML is the “official” Metadata-format promoted by the (W3C)
  - thus on the same level as standards like HTML,XML!

- You can view each triple as an edge in a graph… and a set of triples as a directed graph.


http://www.polleres.net/page.html
http://www.example.org/worksFor
http://www.deri.ie/institute
http://purl.org/dc/elements/1.1/creator
http://www.polleres.net/foaf.rdf#me
http://purl.org/dc/elements/1.1/language
foaf:name
en
Axel Polleres

# RDF data model:

<http://www.polleres.net/page.html> dc:creator <http://www.polleres.net/foaf.rdf#me.>
<http://www.polleres.net/foaf.rdf#me> foaf:name “Axel Polleres“.

- **Resources**
  - objects identified by a URI (**U**niform **R**esource **I**dentifier)
  - all things can have a URI, not only Web documents!
  - E.g. “http://www.deri.org/deri”, “http://www.w3c.org/RDF”, “http://www.polleres.net/index.html” , “http://www.polleres.net/foaf.rdf#me ”
  - Namespaces/URIs help to guarantee that people use unique ids for objects they describe.

- **Properties**:
  - attributes or relations, a property can itself be a resource
  - e.g. ex2:Lives-In, ex1:hasColor, org:worksFor, rdf:type, xyz:includedIn, dc:creator, foaf:name *

- **Satements: triple** <s,p,o> for subject, predicate, Object, where
  - **S: resource** (or blank node)
  - **P: property**
  - **O: resource or literal** (or blank node)
    - literals are strings, numeric values, or XML-fragments

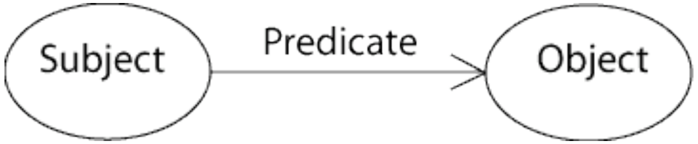
      e.g. "Axel Polleres", "1"^^xsd:integer, `<article><artNr>01024</artNr><Name>chocolate</Name></article>`

* The notation prefix:name is an abbreviation for the full URI in RDF, e.g.. dc = http://purl.org/dc/elements/1.1/ , i.e. dc:creator means in reality: http://purl.org/dc/elements/1.1/creator ... this is similar, to XML qnames (=namespace prefixed elements and attributes)

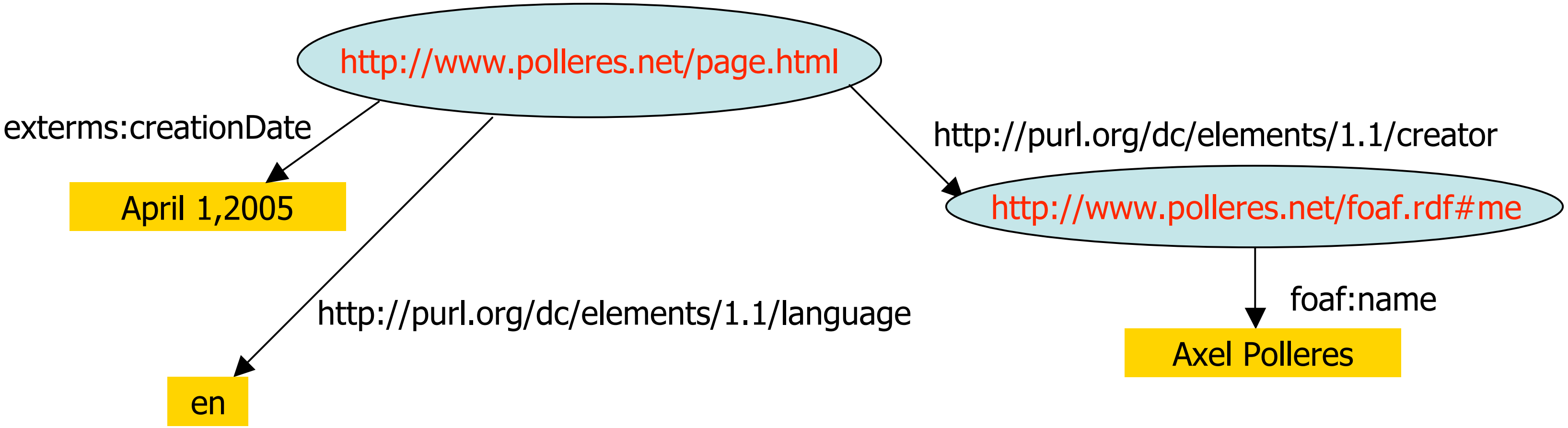
# RDF as a graph:

```
<http://www.polleres.net/page.html>     dc:creator      <http://www.polleres.net/foaf.rdf#me> .
<http://www.polleres.net/foaf.rdf#me>   foaf:name        “Axel Polleres“.
<http://www.polleres.net/page.html>     dc:creationDate  “April 1,2004”
<http://www.polleres.net/page.html>   <http://purl.org/dc/elements/1.1/language> “en”.
```

- You can view each triple as an edge in a graph…



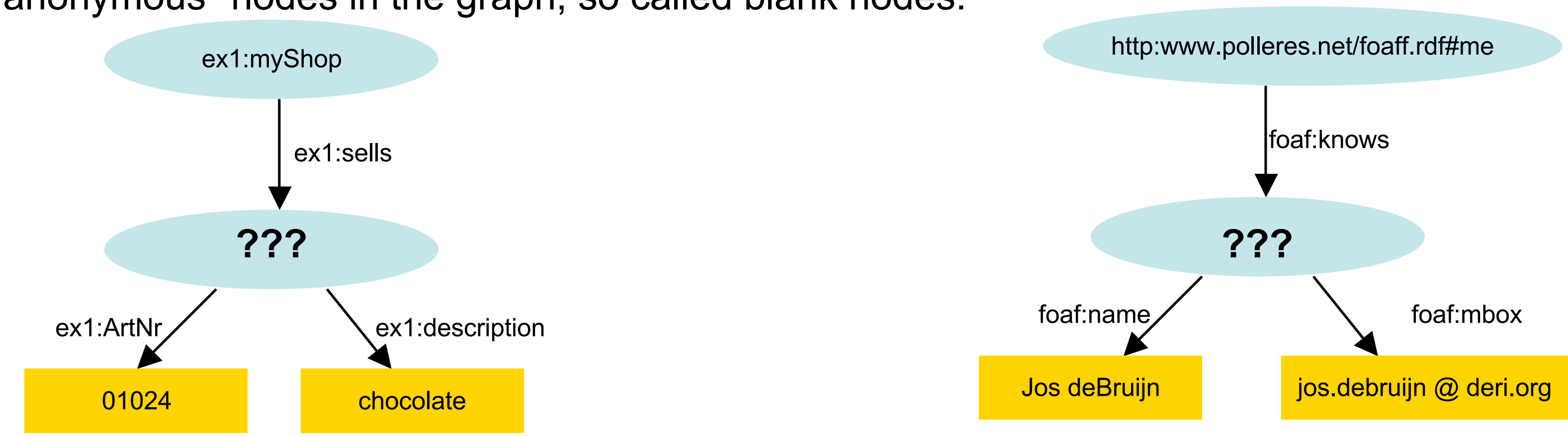
- … and a set of triples as a graph.
- Graph notation is often more intuitive especially for tools:


http://www.polleres.net/page.html
exterms:creationDate
April 1,2005
http://purl.org/dc/elements/1.1/creator
http://www.polleres.net/foaf.rdf#me
foaf:name
Axel Polleres
http://purl.org/dc/elements/1.1/language
en

# RDF Blank Nodes:

- z.B. statement with incomplete information:
  - „myShop sells the artikel with number 01024 and name 'chocolate' “
  - „Axel Polleres knows a person whose name is Jos DeBruijn and who has the email-address jos.debruijn@deri.org “

You cannot write such statements as single triples, but as a set of triples with "anonymous" nodes in the graph, so called blank nodes:



In triple-notation you have to give the anonymous nodes a label, here the prefix „_:“ is used:

```
<http:www.polleres.net/foaf.rdf#me>          foaf:knows   _:jos.
_:jos                                         foaf:name    “Jos de Bruijn“.
_:jos                                         foaf:mbox    jos.debruijn@deri.org.
```

# 2 RDF syntaxes:

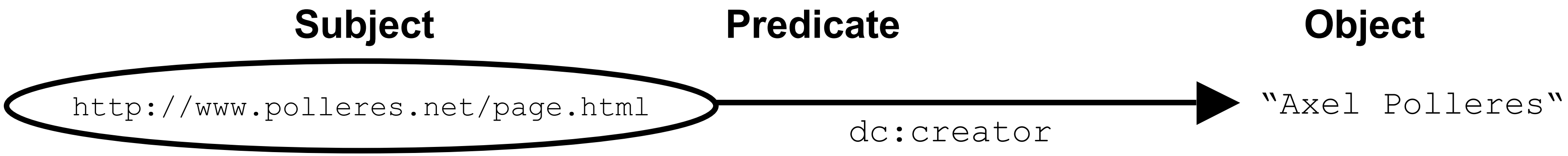
- A machine readable XML exchange syntax: RDF/XML
  - http://www.w3.org/TR/rdf-syntax-grammar/

- Turtle - Terse RDF Triple Language
  - http://www.dajobe.org/2004/01/turtle/

  (for tools, editors, queries etc. writing down explaining RDF)

# RDF/XML:

- RDF written down as Triples or graph notation cannot be integrated in a webpage in a machine readable way…
- machine-readable exchange format is necessary: RDF/XML
- Example for RDF/XML Syntax:



```
<?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/”>

  <rdf:Description rdf:about = “http://www.polleres.net/page.html”>
    <dc:creator>Axel Polleres</dc:creator>
  </rdf:Description>

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

# RDF/XML Structure

XML-Namespaces:

- references a web-resource with unambiguous vocabulary
- external term definitions (ontologies like here Dublin Core) can be referenced

Head

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

Body (RDF statements)

```
  <rdf:Description rdf:about=“http://www.Batman.org/Robin/”>
    <mySchema:employedBy rdf:resource=“#Batman”/>
  </rdf:Description>

  <rdf:Description rdf:id=“Batman”>
    <mySchema:Name>Bruce Wayne</mySchema:Name>
    <mySchema:Email rdf:resource=“mailto:boss@batman.org“/>
  </rdf:Description>
```

End

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

described Resource

Reference to Identifier

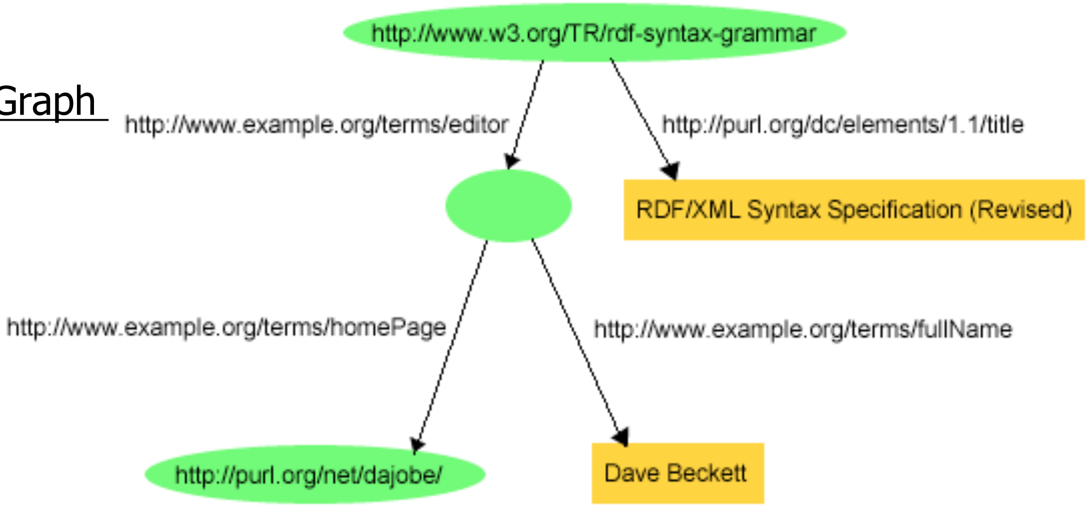
# RDF/XML – several statements:

Several  statements on the same subject can be written within a single <rdf:Description> :

```
<?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.polleres.net/index.html">
      <exterms:creationDate>April 1, 2005</exterms:creation-date>
      <dc:language>en</dc:language>
      <dc:creator rdf:resource="http://www.polleres.net/foaf.rdf#me"/>
   </rdf:Description>
   ...
 </rdf:RDF>
```

# RDF/XML Syntax – Blank Nodes

RDF Graph


http://www.w3.org/TR/rdf-syntax-grammar
http://www.example.org/terms/editor
http://purl.org/dc/elements/1.1/title
RDF/XML Syntax Specification (Revised)
http://www.example.org/terms/homePage
http://www.example.org/terms/fullName
http://purl.org/net/dajobe/
Dave Beckett

RDF/XML Syntax

```
<?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="x"/>
  </rdf:Description>

  <rdf:Description rdf:nodeID="x">
     <exterms:fullName>Dave Beckett</exterms:fullName>
     <exterms:homePage rdf:resource="http://purl.org/net/dajobe/"/>
  </rdf:Description>
</rdf:RDF>
```

These so-called. **nodeIDs** are referenzes only valid in the scope of one RDF/XML documents, but which don't have a unique URI (cf. the scope of an existential variable in a formula!)

# Some more features of RDF:

- Typed nodes
- rdf:IDs
- Containers
- Collections
- Reification

# RDF/XML Syntax – typed nodes:

RDF has a special property for **is-a** relations: rdf:Type
RDF triple Notation…

```
<http://www.polleres.net/foaf.rdf#me> rdf:type ex:Person.
```

…in RDF/XML Syntax:

```
<?xml version="1.0"?>
 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:foaf="http://xmlns.com/foaf/0.1//“>

   <rdf:Description rdf:about="http://www.polleres.net/foaf.rdf#me">
     <rdf:type rdf:resource=“http://www.example.org/Person”/>
      ... other predicates for axel ...
   </rdf:Description>
</rdf:RDF>
```

There is an abbreviated notion for this in RDF/XML:

```
<?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/“
          xmlns:ex="http://www.example.org/”>

   <ex:Person rdf:about="http://www.example.org/staffid/85740">
            ... other predicates for 85740 ...
   </ex:Person>

</rdf:RDF>
```

# RDF/XML Syntax – Literals with Datatypes

## RDF triple notation…

```
ex:index.html exterms:creation-date “1999-08-16"^^xsd:date .
```

## …in RDF/XML Syntax:

```
<?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:creationdate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
       1999-08-16
     </exterms:creation-date>
   </rdf:Description>

</rdf:RDF>
```

***Note:*** `http://www.w3.org/2001/XMLSchema#` *can not directly be referred via a namespace prefix here because it is inside the attribute value, but you could use a DTD entity for abbreviation…*

# RDF/XML syntax - IDs

```
<?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.com/terms/"
          xml:base="http://www.example.com/2002/04/products">

 <rdf:Description rdf:ID="item10245">
   <rdf:type rdf:resource="http://www.example.com/terms/Mountainbike"/>
   <exterms:model rdf:datatype="&xsd;string">Rockclimber</exterms:model>
   <exterms:weight rdf:datatype="&xsd;decimal">9.4</exterms:weight>
   <exterms:packedSize rdf:datatype="&xsd;integer">784</exterms:packedSize>
 </rdf:Description>

...other product descriptions...
 </rdf:RDF>
```

An rdf:ID can be referenced in another document relative to base document (similar to IDs in HTML, XML):

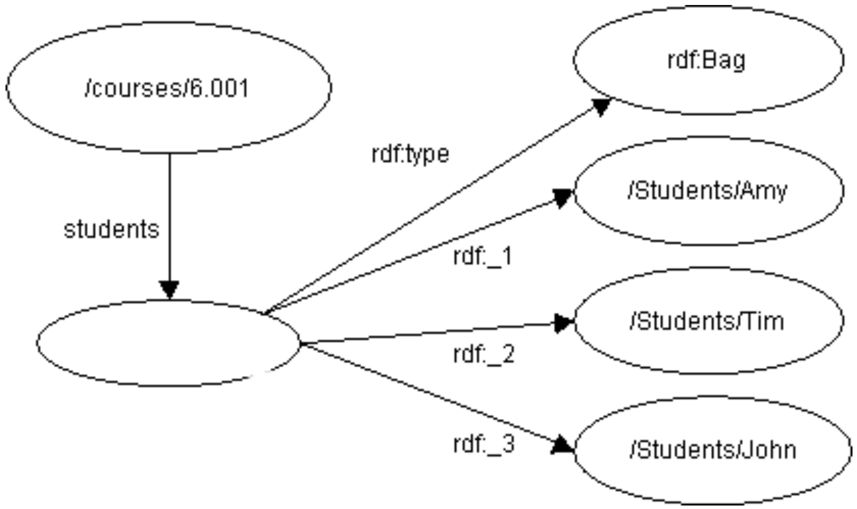
```
<rdf:Description rdf:about=“http://www.polleres.net/foaf.rdf#me”>
 <blabla:owns rdf:resource=“http://www.example.com/2002/04/products#item10245”>
</rdf:Description>
```

# RDF Containers

- If you want to give a set as attribute value, RDF provides 3 types of **Containers**:
  rdf:Bag Unordered list of resources or literals as multiple values of a Property.
  rdf:Sequence Sequence, ordered list of resources or literals as multiple values of a Property.
  rdf:Alt Alternative, list of resources or literals that are alternatives for the (single) value of a Property.

- **Beispiel (rdf:Bag) :**
  Die Studenten im Kurs 6.001 sind Amy, Tim, and John. *

RDF Model



RDF/XML Syntax

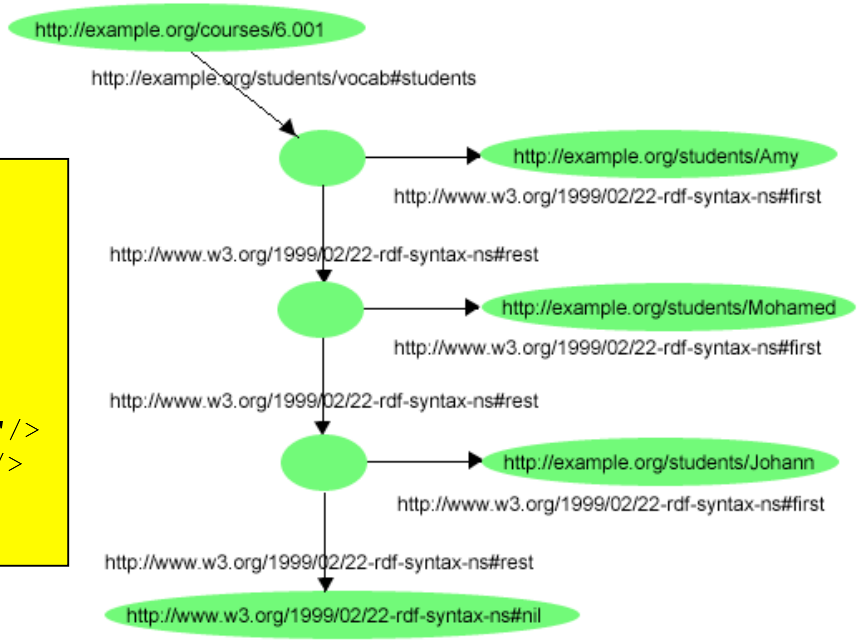
```
<rdf:RDF>
<rdf:Description
           about="http://.../courses/6.001">
 <s:students>
  <rdf:Bag>
    <rdf:li resource=“http://.../students/Amy"/>
    <rdf:li resource="http://.../students/Tim"/>
    <rdf:li resource="http://.../students/John"/>
  </rdf:Bag>
 </s:students>
</rdf:Description>
</rdf:RDF>
```

***: Beispiel von: REC-rdf-syntax-19990222**
http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/#containers

# RDF Collection

Similar to RDF containers, but resolves collection in LISP-style lists:

RDF Model


http://example.org/courses/6.001
http://example.org/students/vocab#students
http://example.org/students/Amy
http://www.w3.org/1999/02/22-rdf-syntax-ns#first
http://www.w3.org/1999/02/22-rdf-syntax-ns#rest
http://example.org/students/Mohamed
http://www.w3.org/1999/02/22-rdf-syntax-ns#first
http://www.w3.org/1999/02/22-rdf-syntax-ns#rest
http://example.org/students/Johann
http://www.w3.org/1999/02/22-rdf-syntax-ns#first
http://www.w3.org/1999/02/22-rdf-syntax-ns#rest
http://www.w3.org/1999/02/22-rdf-syntax-ns#nil

RDF/XML Syntax

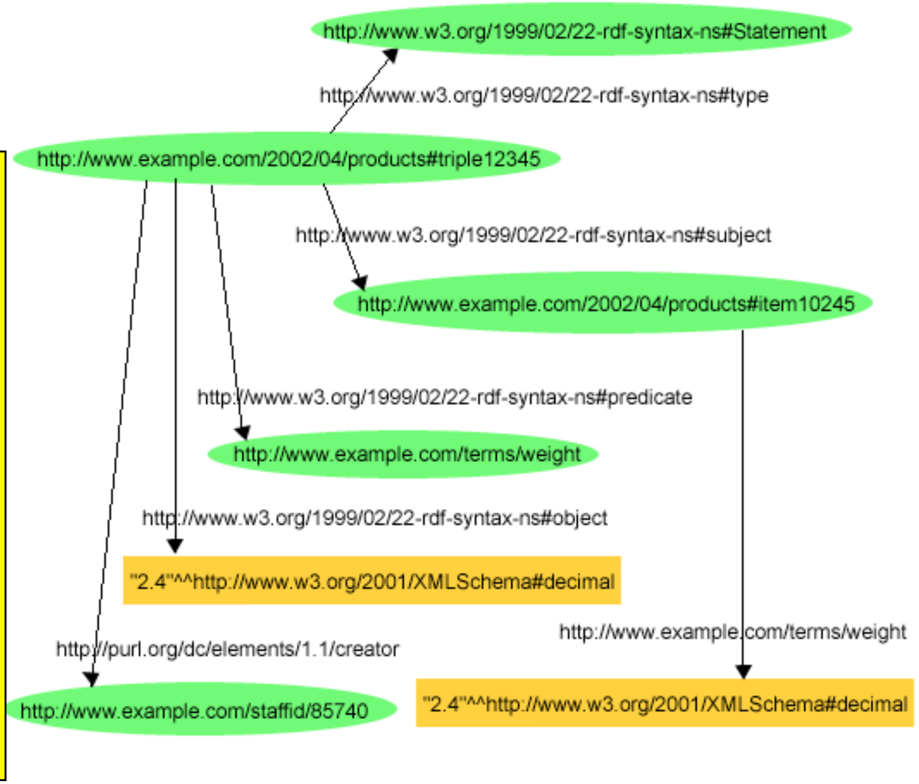
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:s="http://example.org/students/vocab#">

 <rdf:Description rdf:about="http://example.org/courses/6.001">
  <s:students rdf:parseType="Collection">
    <rdf:Description rdf:about="http://example.org/students/Amy"/>
    <rdf:Description rdf:about="http://example.org/students/Mohamed"/>
    <rdf:Description rdf:about="http://example.org/students/Johann"/>
  </s:students> </rdf:Description>
</rdf:RDF>
```

**examples taken from: REC-rdf-primer-20040210**
http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

# Reification

- In RDF you can make statements about statements, e.g. you might want to express when a certain statement has been made, who made it, etc.

RDF Graph:


http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.example.com/2002/04/products#triple12345
http://www.w3.org/1999/02/22-rdf-syntax-ns#subject
http://www.example.com/2002/04/products#item10245
http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate
http://www.example.com/terms/weight
http://www.w3.org/1999/02/22-rdf-syntax-ns#object
"2.4"^^http://www.w3.org/2001/XMLSchema#decimal
http://purl.org/dc/elements/1.1/creator
http://www.example.com/terms/weight
http://www.example.com/staffid/85740
"2.4"^^http://www.w3.org/2001/XMLSchema#decimal

RDF/XML Syntax

```
<?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:dc="http://purl.org/dc/elements/1.1/"
         xmlns:exterms="http://www.example.com/terms/"
         xml:base="http://www.example.com/2002/04/products">

<rdf:Description rdf:ID="item10245">
  <exterms:weight rdf:datatype="&xsd;decimal">2.4</exterms:weight>
</rdf:Description>

<rdf:Statement rdf:about="#triple12345">
  <rdf:subject
   rdf:resource="http://www.example.com/2002/04/products#item10245"/>
  <rdf:predicate rdf:resource="http://www.example.com/terms/weight"/>
  <rdf:object rdf:datatype="&xsd;decimal">2.4</rdf:object>
  <dc:creator rdf:resource="http://www.example.com/staffid/85740"/>
</rdf:Statement>

</rdf:RDF>
```

***Remark:*** *Here we use the abbreviated notation from slide 26*

# Alternative: Use of rdf:ID for Reification

In an rdf:Description one can give an ID for triples...

```
<rdf:Description rdf:ID="item10245">
   <exterms:weight rdf:ID="triple12345‟ rdf:datatype="&xsd;decimal">
   2.4</exterms:weight>
</rdf:Description>

<rdf:Description rdf:about="#triple12345">
  <dc:creator rdf:resource="http://www.example.com/staffid/85740"/>
</rdf:Description>
```

… which can be reused in other meta-statements later on!

# Overview

- What is the Semantic Web?
- RDF
  - RDF/XML
  - **RDF Turtle**
- SPARQL - A Query language for the Semantic Web
- What’s next?
  - RDFS + OWL
  - Next Time: Outlook/Applications/Review of Assignments

# Turtle:

- RDF/XML language hard to read

- Turtle = Terse RDF Tripl LanguagE

- Plain text syntax for RDF
  - Based on Unicode
- Mechanisms for namespace abbreviation
- Allows grouping of triples according to subject
- Shortcuts for collections
- In short:
  - Takes good things of RDF/XML
  - leaves out angle brackets

# Prefixes

Mechanism for namespace abbreviation

- Syntax:

```
@prefix abbr : <URI > .
```

- Example:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

- Default Namespace:

```
@prefix : <URI > .
```

- Example:

```
@prefix : <http://example.org/myOntology#> .
```

# Identifiers in Turtle

- URIs: `<URI >`

  `<http://www.w3.org/1999/02/22-rdf-syntax-ns#>`
- QNames: `namespace-abbr ?:localname`

  `rdf:type dc:title :hasName`
- Literals: `"string "(@lang )?(^^type )?`

  `"John"   "Hello"@en-GB   "1.4"^^xs:decimal`
- Typed literal shortcuts:
  - integer: `2   45`
  - decimal: `2.4   5.67`
  - boolean: `true false`

# Triples in Turtle

- Simple triple: `subject predicate object .`

```
 :john rdf:label "John” .
```

- Grouping triples:

```
subject predicate object ; predicate object … .
subject predicate object , object … .
```

  - Examples:

```
:john rdf:label "John" ;
     rdf:type foaf:Person , ex:Employee ;
     ex:homePage <http://example.org/johnspage/> .
```

- Shortcut for `rdf:type “a” :`

```
:john a ex:Person .
:john a ex:Employee .
```

# Blank Nodes in Turtle

- Simple blank node: `[]`

```
      :john ex:hasFather [] .
```

- Blank node as subject: `[ predicate object ; predicate object ... ] .`

```
[ ex:hasName "John"] . [ ex:authorOf :lotr ; ex:hasName "Tolkien"] .
```

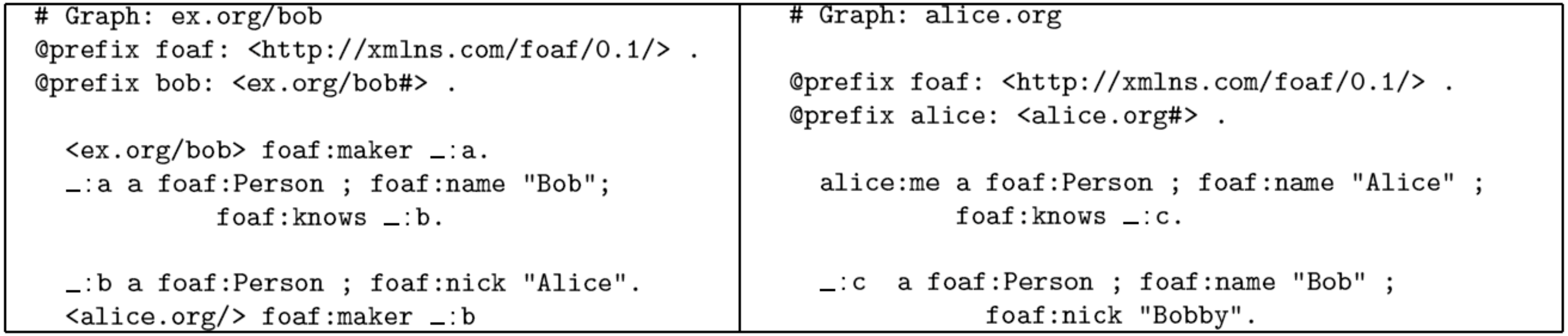
- Collections: `( object1 ... objectn )`

```
      :doc1 ex:hasAuthor (:john :mary) .
```

is short for:

```
      :doc1 ex:hasAuthor
               [ rdf:first :john; rdf:rest
                          [ rdf:first :mary; rdf:rest rdf:nil ]
            ] .
```

# 2 Example RDF graphs in Turtle:



```
# Graph: ex.org/bob
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix bob: <ex.org/bob#> .

  <ex.org/bob> foaf:maker _:a.
  _:a a foaf:Person ; foaf:name "Bob";
            foaf:knows _:b.

  _:b a foaf:Person ; foaf:nick "Alice".
  <alice.org/> foaf:maker _:b
```

```
# Graph: alice.org

@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix alice: <alice.org#> .

  alice:me a foaf:Person ; foaf:name "Alice" ;
           foaf:knows _:c.

  _:c  a foaf:Person ; foaf:name "Bob" ;
             foaf:nick "Bobby".
```

# Overview

- What is the Semantic Web?
- RDF
  - RDF/XML
  - RDF Turtle
- **SPARQL - A Query language for the Semantic Web**
- What's next?
  - RDFS + OWL
  - Next Time: Outlook/Applications/Review of Assignments

# SPARQL - a query language for the Semantic Web

- After having created a Data model,
- A query language is the next important step to make the Semantic Web real:

- SPARQL (SPARQL Protocol and RDF Query Language)
  - “SQL for RDF”
  - A W3C standard Recommendation since 15 January 2008

# SPARQL: “SQL Look-and-feel on top of RDF”

- Basic concept “graph Pattern Matching”
- The two most important forms
  - SELECT
  - CONSTRUCT

- We start with SELECT:

```
SELECT      Variables
FROM        Dataset
WHERE       Pattern
```

# SPARQL

- Simple Protocol and RDF Query Language
  - Basic Graph Patterns (Conjunctive queries)
  - UNIONS
  - GRAPH Patterns
  - OPTIONAL Patterns
  - FILTERs

# SPARQL Queries

- 3 basic forms
  - SELECT
  - ASK
  - CONSTRUCT

- We start with SELECT:

```
SELECT      Variables
FROM        Dataset
WHERE       Pattern
```

# Basic Graph Patterns (Conjunctive queries)

*"select persons and their names"*

```
SELECT ?X ?Y
FROM <http://alice.org>
FROM <http://ex.org/bob>
WHERE { ?X a foaf:Person . ?X foaf:name ?Y . }
```

| ?X | ?Y |
|---|---|
| _:a | "Bob" |
| _:c | "Bob" |
| alice:me | "Alice" |

# UNIONs

*"select Persons and their names* **or** *nicknames"*

```
SELECT ?X ?Y
FROM ...
WHERE { { ?X foaf:name ?Y . }
         UNION { ?X foaf:nick ?Y .} }
```

| ?X | ?Y |
|---|---|
| _:a | "Bob" |
| _:c | "Bob" |
| alice:me | "Alice" |
| _:b | "Alice" |
| _:c | "Bobby" |

# GRAPH patterns and named graphs

*"select creators of graphs and the persons they know"*

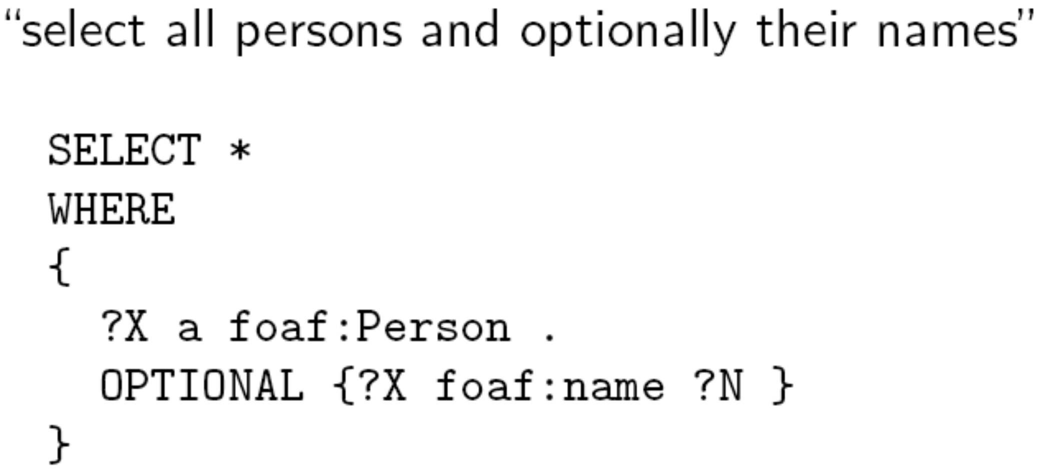
```
SELECT ?X ?Y
FROM <alice.org>
FROM <ex.org/bob>
FROM NAMED <ex.org/bob>
WHERE { ?G foaf:maker ?X .
         GRAPH ?G { ?X foaf:knows ?Y. } }
```

| ?X | ?Y |
|---|---|
| _:a | _:b |

Note: Since only `ex.org/bob` is in the named graphs ( `FROM NAMED` clause), the `foaf:knows` triples from the graph `alice.org` are not in the solutions!

# OPTIONAL

- Optional matching for incomplete matches… leaves unmatchable variables unbound:



“select all persons and optionally their names”

```
SELECT *
WHERE
{
   ?X a foaf:Person .
   OPTIONAL {?X foaf:name ?N }
}
```

| ?X | ?N |
|---|---|
| _:a | “Bob” |
| _:b | |
| _:c | “Bob” |
| alice:me | “Alice” |

# FILTERs

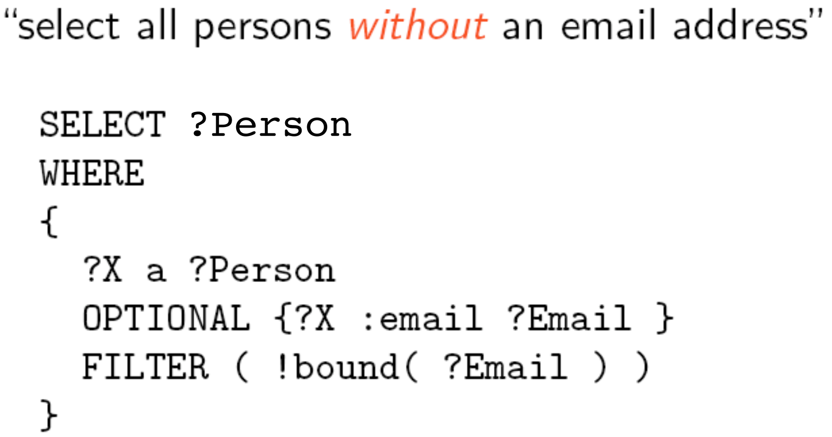
- By means of FILTERs, one can filter out undesired solutions, e.g.

  "select persons older than 30"

```
SELECT ?X
WHERE { ?X a foaf:person .
              ?X ex:age ?Y .
                  FILTER (?Y > 30)
               }
```

  - FILTERs can be complex boolean combinations ( &&, ||, !)
  - Special FILTER functions allowed, e.g. "BOUND(*Var*)"

- FILTERs can be used to emulate set difference (or negation as failure):



"select all persons *without* an email address"

```
SELECT ?Person
WHERE
{
  ?X a ?Person
  OPTIONAL {?X :email ?Email }
  FILTER ( !bound( ?Email ) )
}
```

- Attention: FILTERs can NOT bind new variables!

"select Persons and their age+1" ????

```
SELECT ?X ?Y
WHERE { ?X ex:age ?Z .
     FILTER (?Y = ?Z + 1 ) }
```

will not produce results, since "unbound = 33+1" gives an error.

# CONSTRUCT

- allows to create new triples …

```
CONSTRUCT{ :me foaf:knows ?X }
FROM <http://www.deri.ie/about/team>
WHERE { ?X a foaf:Person.  }
```

- Tricky: blank nodes in CONSTRUCT

```
CONSTRUCT { :me foaf:knows _:x .
            _:x foaf:name ?X}
FROM <http://www.deri.ie/about/team>
WHERE { _:y foaf:name ?X . }
```

*Will in the result RDF Graph produce a new blankNode Id instead of _:x **for each** binding of ?X*

# That’s all!

- Very simple, let’many useful extensions still missing, e.g.
  - calculating new bindings
  - Aggregates
- Let’s try out a view queries!

# Our own prototype SPARQL engine available:

- http://www.polleres.net/dlvhex-sparql/
- Based on dlvhex Logic Programming engine

Joint work with Thomas Krennwallner, Roman Schindlauer, Gennaro Frazzingaro

dlvhex
SPARQL

- Please let us know! Any found bug brings extra-points in the assignments

- “Backup”: Jena + ARQ (HP Labs) http://jena.sourceforge.net/ARQ/

# Overview

- What is the Semantic Web?
- RDF
  - RDF/XML
  - RDF Turtle
- SPARQL - A Query language for the Semantic Web
- **What's next?**
  - RDFS + OWL
  - Next Time: Outlook/Applications/Review of Assignments

# Now where is the metadata? RDFS and OWL

- RDF is completely unstructured
- SPARQL queries over a graph, not over a fixed relation or schema.
- How do I describe schema information on the Semantic Web?

- How do we describe the meaning of triples?
- I.e., how do we describe meta-level information such as

→ RDFS, OWL

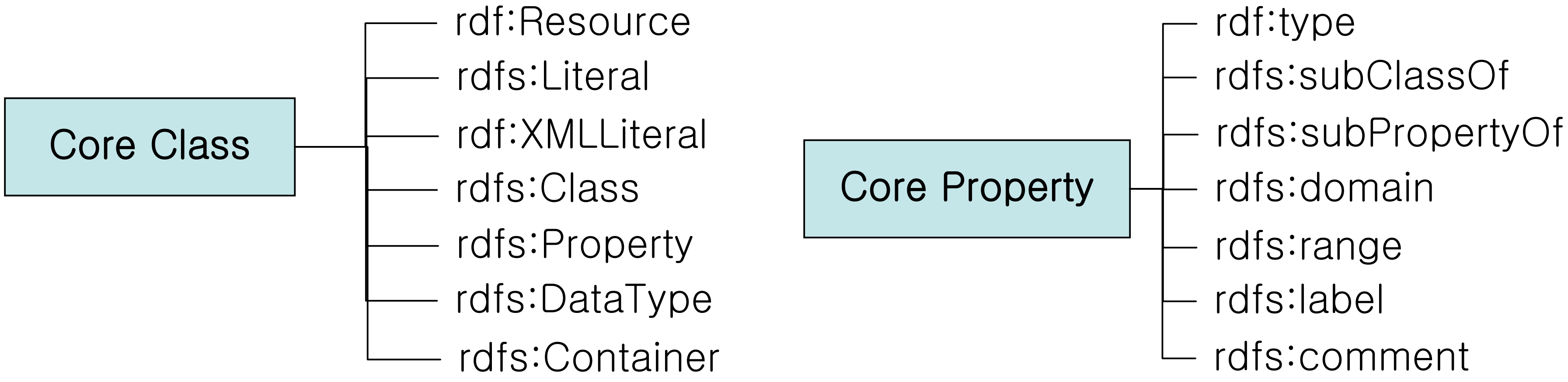
All researchers working at DERI know about the Semantic Web.

Persons who know about Semantic Web technnologies, know about RDF.

# RDFS – a simple Ontology language

- With **rdf:type** we can express what is the Type of a resource.
- **RDF Schema** (RDFS) describes relations between the used Types and properties inRDF
- RDFS thus is a “Meta-Meta-description language” for describing the structure of knowledge, i.e. with RDFS we can describe the used **Vocabulary: Classes and Properties**
- Uses and extends RDF:

The most important RDFS constructs:



Core Class

rdf:Resource
rdfs:Literal
rdf:XMLLiteral
rdfs:Class
rdfs:Property
rdfs:DataType
rdfs:Container

Core Property

rdf:type
rdfs:subClassOf
rdfs:subPropertyOf
rdfs:domain
rdfs:range
rdfs:label
rdfs:comment

# RDFS Examples:

Usage of rdf:type, rdfs:subclassOf, rdfs:subPropertyOf, rdfs:domain, rdfs:range:

„Axel Polleres is a researcher“

<http://www.polleres.net/foaf.rdf#me> rdf:type ex1:Researcher.

„People who are experts in Semantic Web, are experts in RDF.“

ex1:SemanticWebExpert rdfs:subclassof ex1:RDFExpert.

„Professors are employees of the university.“

deri:prof rdfs:subClassOf deri:employee.

„If A is the father of B, then A is also a parent of B.“

ex1:fatherOf rdfs:subPropertyOf ex:parentOf.

„If A is a parent of B, then A is a Person.“

ex1:parentOf rdfs:domain foaf:Person.

„If A is parent of B , then B is a Person.“

ex1:parentOf rdfs:range foaf:Person.

All RDFS statements are again RDF statement!

Difference: With additional RDFS information you can **derive** new information!!!

# Infering new Information from Meta-Data in RDF+RDFS:

Axel Polleres is a researcher

Axel Polleres works for DERI

- <http://www.polleres.net/foaf.rdf#me> rdf:type ex1:researcher.
- <http://www.polleres.net/foaf.rdf#me> ex1:worksFor <http:www.deri.org/deri>.

… Let‘s assume DERI defines a new class deri:researcher in their corporate RDF Schema:

- <http://www.polleres.net/foaf.rdf#me> rdf:type deri:researcher.

We have additional RDFS information:

- deri:researcher rdf:subclassOf ex1:SemanticWebExpert

ex1:SemanticWebExpert rdfs:subclassof ex1:RDFExpert.

All researchers at DERI are Semantic Web experts.

Persons who know about Semantic Web technnologies, know about RDF.

Now the following **inference rules** are valid in RDFS:

- If ***A*** rdfs:subclassOf ***B.*** and ***B*** rdfs:subclassOf ***C.*** then also
  ***A*** rdfs:subclassOf ***C.***
- If ***R*** rdf:type ***A.*** and ***A*** rdfs:subclassOf ***B.*** then also
  ***R*** rdf:type ***B***

***An RDFS aware application can infer from this:***

<http://www.polleres.net/foaf.rdf#me> rdf:type ex1:RDFExpert.
Thus: „Axel Polleres is an expert for RDF “

We need someone who knows about RDF

# RDFS vs logical inference

- This looked a lot like logics, didn't it?
- Most of the RDFS inference rules can be written down as Prolog style rules!

- If ***A*** rdfs:subclassOf ***B.*** and ***B*** rdfs:subclassOf ***C.*** then also ***A*** rdfs:subclassOf ***C.***

```
triple(A,rdfs:subClassOf,C)   :-          triple(A,rdfs:subClassOf,B),
                                          triple(B,rdfs:subClassOf,C).
```

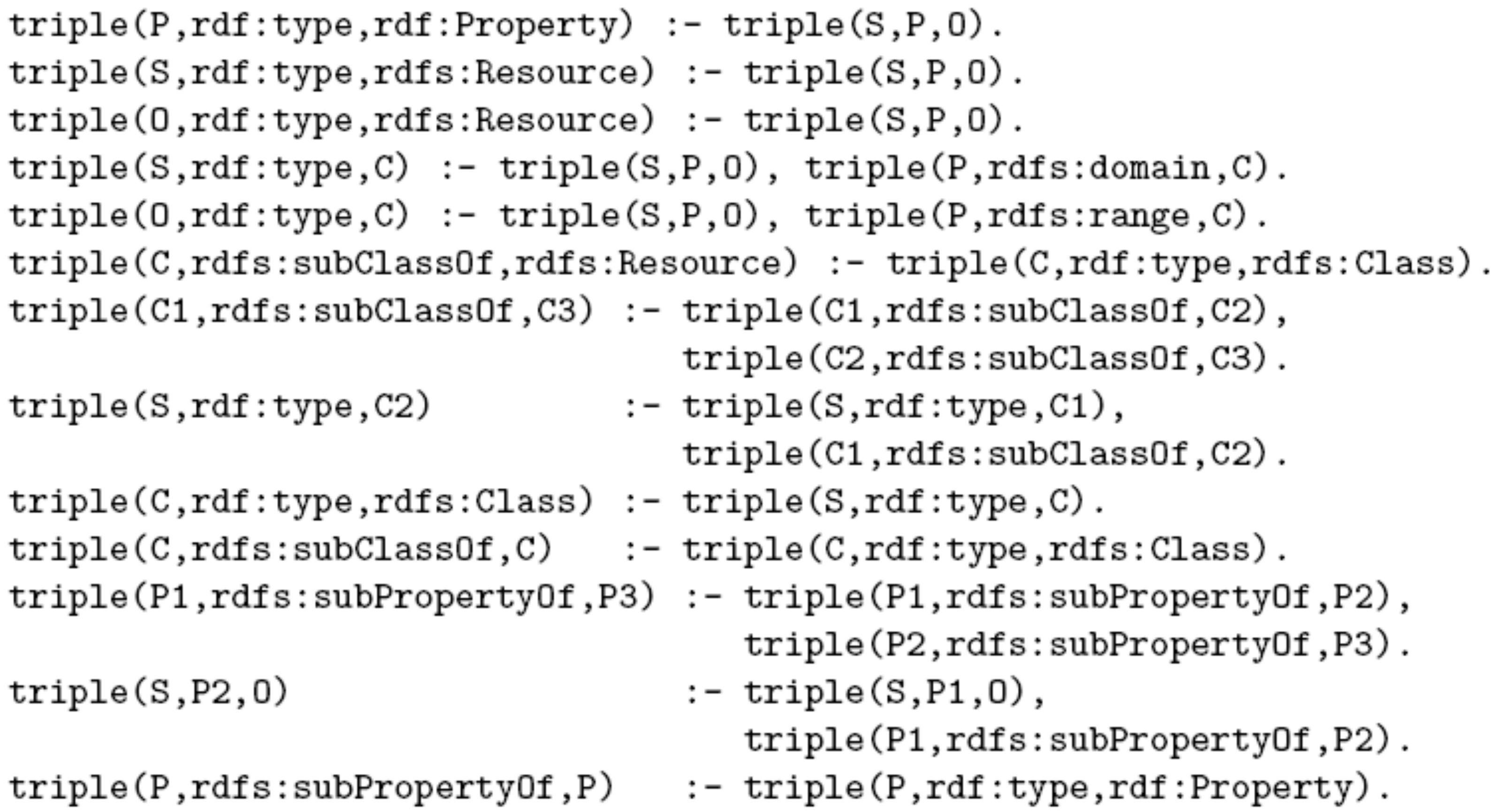
- If ***R*** rdf:type ***A.*** and ***A*** rdfs:subclassOf ***B.*** then also ***R*** rdf:type ***B***

```
triple(R,rdf:type,B)   :-     triple(R,rdf:type,A),
                                          triple(A,rdfs:subClassOf,B).
```

- However, the logics of RDF/RDFS is not very expressive…
- A list of RDFS inference rules can be found in the RDF semantics document:

http://www.w3.org/TR/rdf-mt/ Sections 3.1 and 4.1

The essence of RDF and RDFS semantics is expressible in PROLOG style rules:



```
triple(P,rdf:type,rdf:Property) :- triple(S,P,O).
triple(S,rdf:type,rdfs:Resource) :- triple(S,P,O).
triple(O,rdf:type,rdfs:Resource) :- triple(S,P,O).
triple(S,rdf:type,C) :- triple(S,P,O), triple(P,rdfs:domain,C).
triple(O,rdf:type,C) :- triple(S,P,O), triple(P,rdfs:range,C).
triple(C,rdfs:subClassOf,rdfs:Resource) :- triple(C,rdf:type,rdfs:Class).
triple(C1,rdfs:subClassOf,C3) :- triple(C1,rdfs:subClassOf,C2),
                                 triple(C2,rdfs:subClassOf,C3).
triple(S,rdf:type,C2)         :- triple(S,rdf:type,C1),
                                 triple(C1,rdfs:subClassOf,C2).
triple(C,rdf:type,rdfs:Class) :- triple(S,rdf:type,C).
triple(C,rdfs:subClassOf,C)   :- triple(C,rdf:type,rdfs:Class).
triple(P1,rdfs:subPropertyOf,P3) :- triple(P1,rdfs:subPropertyOf,P2),
                                    triple(P2,rdfs:subPropertyOf,P3).
triple(S,P2,O)                   :- triple(S,P1,O),
                                    triple(P1,rdfs:subPropertyOf,P2).
triple(P,rdfs:subPropertyOf,P)   :- triple(P,rdf:type,rdf:Property).
```

# RDFS & SPARQL?

- Recall some slides before:
  - Some common features missing:
    - no aggregates (sum, count, etc.)
    - no views.
    - Even worse: Most current SPARQL engines do not (yet) support RDFS

# What‘s missing in RDFS?

- Some things cannot be expressed in RDFS e.g. in the example before:

http://www.polleres.net/foaf.rdf#me rdf:type ex1:researcher.
http://www.polleres.net/foaf.rdf#me ex1:worksFor http:www.deri.org/deri.

All researchers at DERI are Semantic Web experts.

Here we did a trick and „invented“ a new class.
The Web Ontology Language **OWL allows to say more about classes and properties**, e.g. one can define classes not only by subclass relationships, but also by their attribute Values…

```
<owl:Class>
  <owl:intersectionOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#worksFor" />
        <owl:hasValue rdf:resource="http:www.deri.org/deri" />
      </owl:Restriction>
      <owl:Class rdf:about="#Researcher" />
  </owl:intersectionOf>
  <owl:subclassOf rdf:about=“” rdf:resource=“#SemanticWebExpert”/>
</owl:Class>
```

# Web Ontology Language - OWL

- RDFS is useful, but doesn‘t solve all problems
- Complex applications need more expressivity:
  - More complex inferences by Terms:
    - E.g. transitive Properties can not be described in RDFS: “if «A» is ancestor of «B» and «B» is ancestor of «C», then «A» is ancestor of «C»?"

      ex:ancestorOf rdf:type owl:TransitiveProperty .
  - Equivalences of individuals and classes

    <http://www.polleres.net/> owl:sameAs <http://platon.escet.urjc.es/~axel/>.
  - *Build classes from other classes, describe them by union, intersection, attribute values, cardinality of attributes*, etc.
- OWL (Web Ontology Language) addresses these and other problems.

- Based on works in the theoretical fundament of *Description Logics rather than (deductive) Databases*
- Combinations with SPARQL not entirely clear, ongoing research

- … and OWL is not the end of the story!

  Further extensions of OWL/RDFS such as complex rules languages are currently under development in the W3C -> RIF Working Group

# Common Data Formats:

- Predefined vocabularies in (RDFS) are beginning to be adopted. Examples:
  - Dublin Core
  - FOAF
  - RSS/SIOC
- Apart from that, easier to use microformats are gaining momentum (machine readable formats directly embedded in HTML)
  - hCalendar
  - hCard
  - XFN
  - etc

# Dublin Core Metadata Set

*The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to promoting the widespread adoption of interoperable metadata standards and developing specialized metadata vocabularies for describing resources that enable more intelligent information discovery systems.*

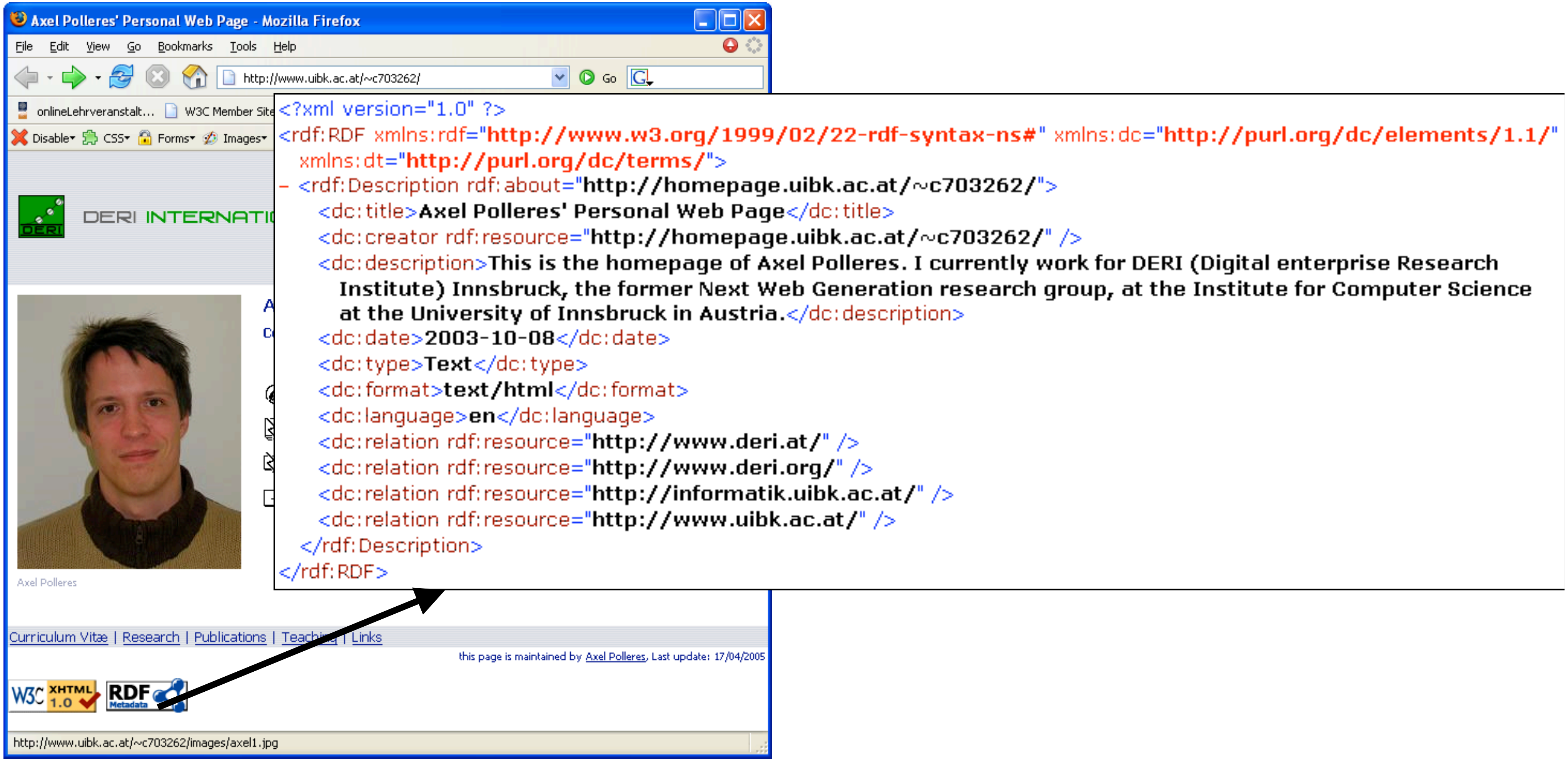
(http://dublincore.org/)

# Dublin Core

- The Dublin Core Metadata Set [DublinCore] is a standard vocabulary for describing resources (such as web resources, pictures, etc.) in RDF syntax, with keywords such as:
  - title
  - description
  - author
  - creator
  - format
  - date
  - type
  - relation
  - …

# Dublin Core an example:

- My homepage: http://www.polleres.net/metadata.rdf



Axel Polleres' Personal Web Page - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.uibk.ac.at/~c703262/
Go
onlineLehrveranstalt... W3C Member Site
Disable CSS Forms Images
DERI INTERNATI
Axel Polleres
Curriculum Vitæ | Research | Publications | Teaching | Links
this page is maintained by Axel Polleres, Last update: 17/04/2005
W3C XHTML 1.0
RDF Metadata
http://www.uibk.ac.at/~c703262/images/axel1.jpg

```
<?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:dt="http://purl.org/dc/terms/">
- <rdf:Description rdf:about="http://homepage.uibk.ac.at/~c703262/">
    <dc:title>Axel Polleres' Personal Web Page</dc:title>
    <dc:creator rdf:resource="http://homepage.uibk.ac.at/~c703262/" />
    <dc:description>This is the homepage of Axel Polleres. I currently work for DERI (Digital enterprise Research
      Institute) Innsbruck, the former Next Web Generation research group, at the Institute for Computer Science
      at the University of Innsbruck in Austria.</dc:description>
    <dc:date>2003-10-08</dc:date>
    <dc:type>Text</dc:type>
    <dc:format>text/html</dc:format>
    <dc:language>en</dc:language>
    <dc:relation rdf:resource="http://www.deri.at/" />
    <dc:relation rdf:resource="http://www.deri.org/" />
    <dc:relation rdf:resource="http://informatik.uibk.ac.at/" />
    <dc:relation rdf:resource="http://www.uibk.ac.at/" />
  </rdf:Description>
</rdf:RDF>
```

All DC vocabulary, see: http://dublincore.org/documents/dcmi-terms/

# FOAF: Friend of a friend

*“The Friend of a Friend (FOAF) project is about creating a Web of machine-readable homepages describing people, the links between them and the things they create and do. ”* (http://www.foaf-project.org/)

- Annotation vocabulary for linking semantic information about people to achieve a social network

- Everybody can provide/link his/her own FoaF file on his/her own webpage.

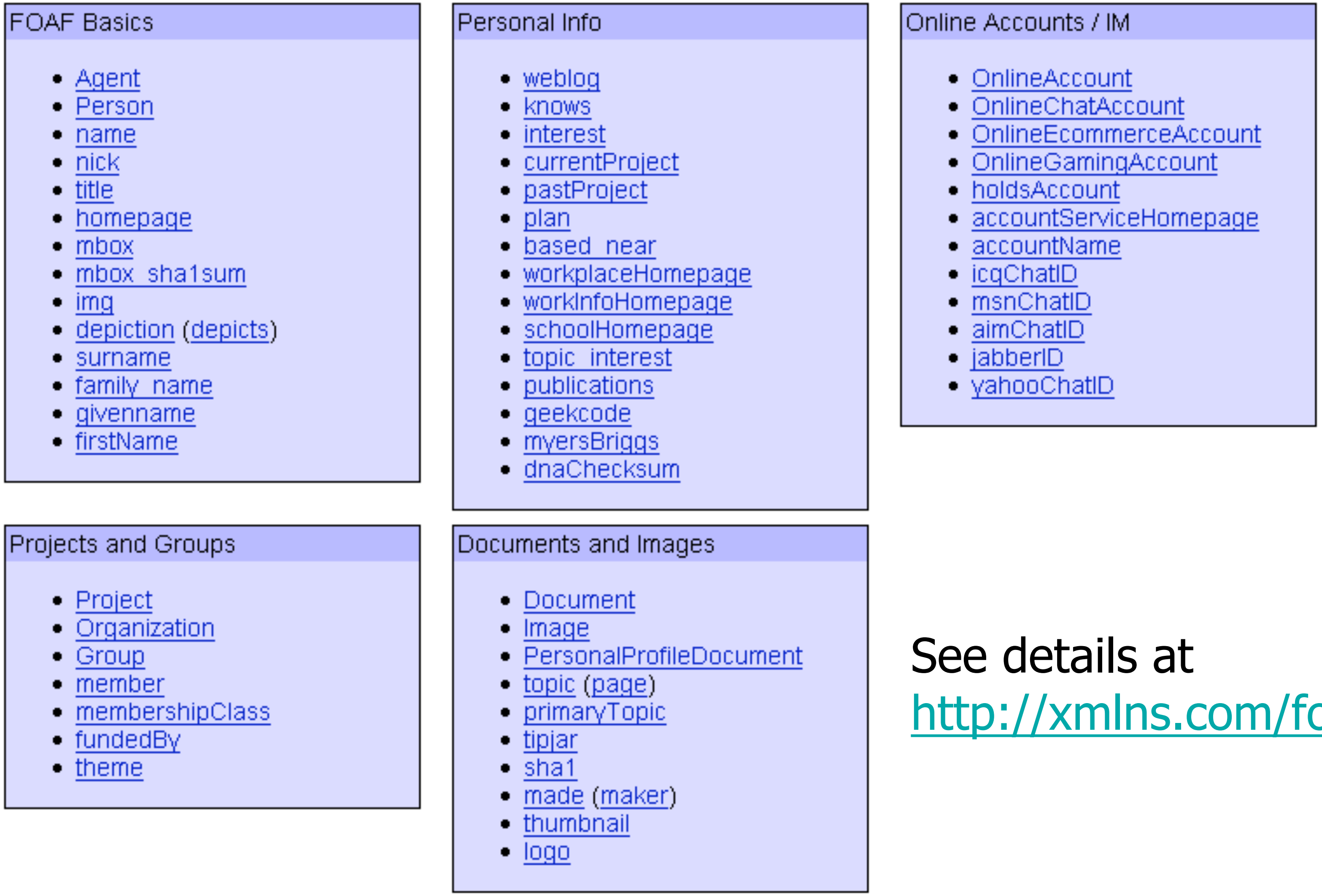
- Related to the success of “social networks”: FaceBook, LinkedIn etc.

  But: these use disclosed applications/databases/portals instead whereas FOAF is intended to work decentralized and based on RDF,

  but: some start to export RDF already

→ LiveJournal is exporting FOAF
→ Google is indexing FOAF
(nice Social Network API recently released)

# FOAF Vocabulary:



FOAF Basics

- Agent
- Person
- name
- nick
- title
- homepage
- mbox
- mbox_sha1sum
- img
- depiction (depicts)
- surname
- family_name
- givenname
- firstName

Personal Info

- weblog
- knows
- interest
- currentProject
- pastProject
- plan
- based_near
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- topic_interest
- publications
- geekcode
- myersBriggs
- dnaChecksum

Online Accounts / IM

- OnlineAccount
- OnlineChatAccount
- OnlineEcommerceAccount
- OnlineGamingAccount
- holdsAccount
- accountServiceHomepage
- accountName
- icqChatID
- msnChatID
- aimChatID
- jabberID
- yahooChatID

Projects and Groups

- Project
- Organization
- Group
- member
- membershipClass
- fundedBy
- theme

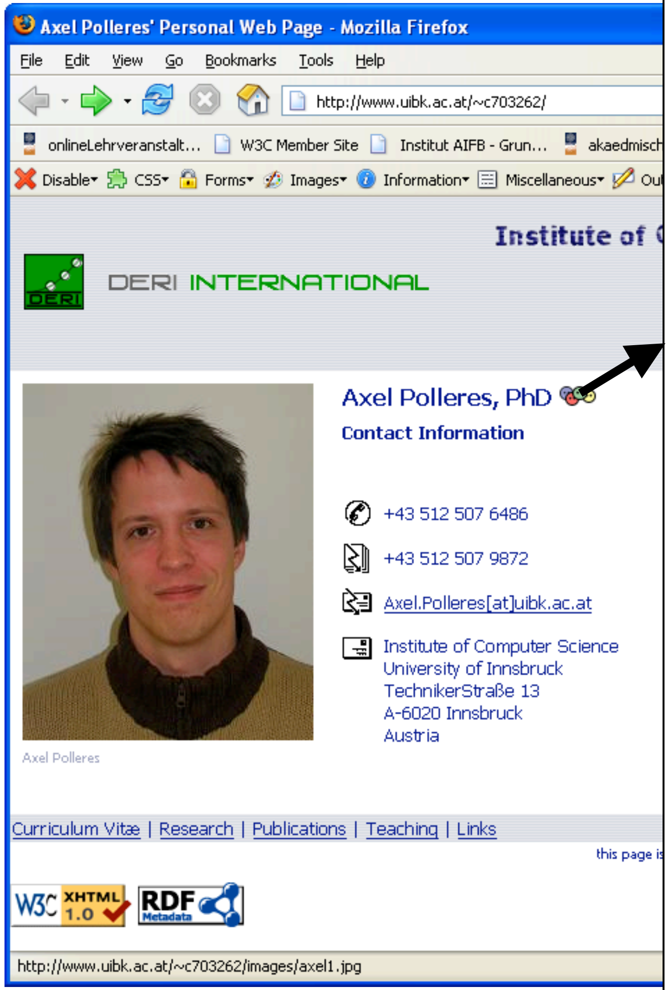
Documents and Images

- Document
- Image
- PersonalProfileDocument
- topic (page)
- primaryTopic
- tipjar
- sha1
- made (maker)
- thumbnail
- logo

See details at http://xmlns.com/foaf/0.1/

# FOAF example:

- Axel's homepage: http://www.polleres.net/foaf.rdf


Axel Polleres' Personal Web Page - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.uibk.ac.at/~c703262/
onlineLehrveranstalt... W3C Member Site Institut AIFB - Grun... akaedmisch
Disable CSS Forms Images Information Miscellaneous Ou
Institute of
DERI INTERNATIONAL
this page i
W3C XHTML 1.0
RDF Metadata
http://www.uibk.ac.at/~c703262/images/axel1.jpg

Axel Polleres, PhD

Contact Information

☎ +43 512 507 6486

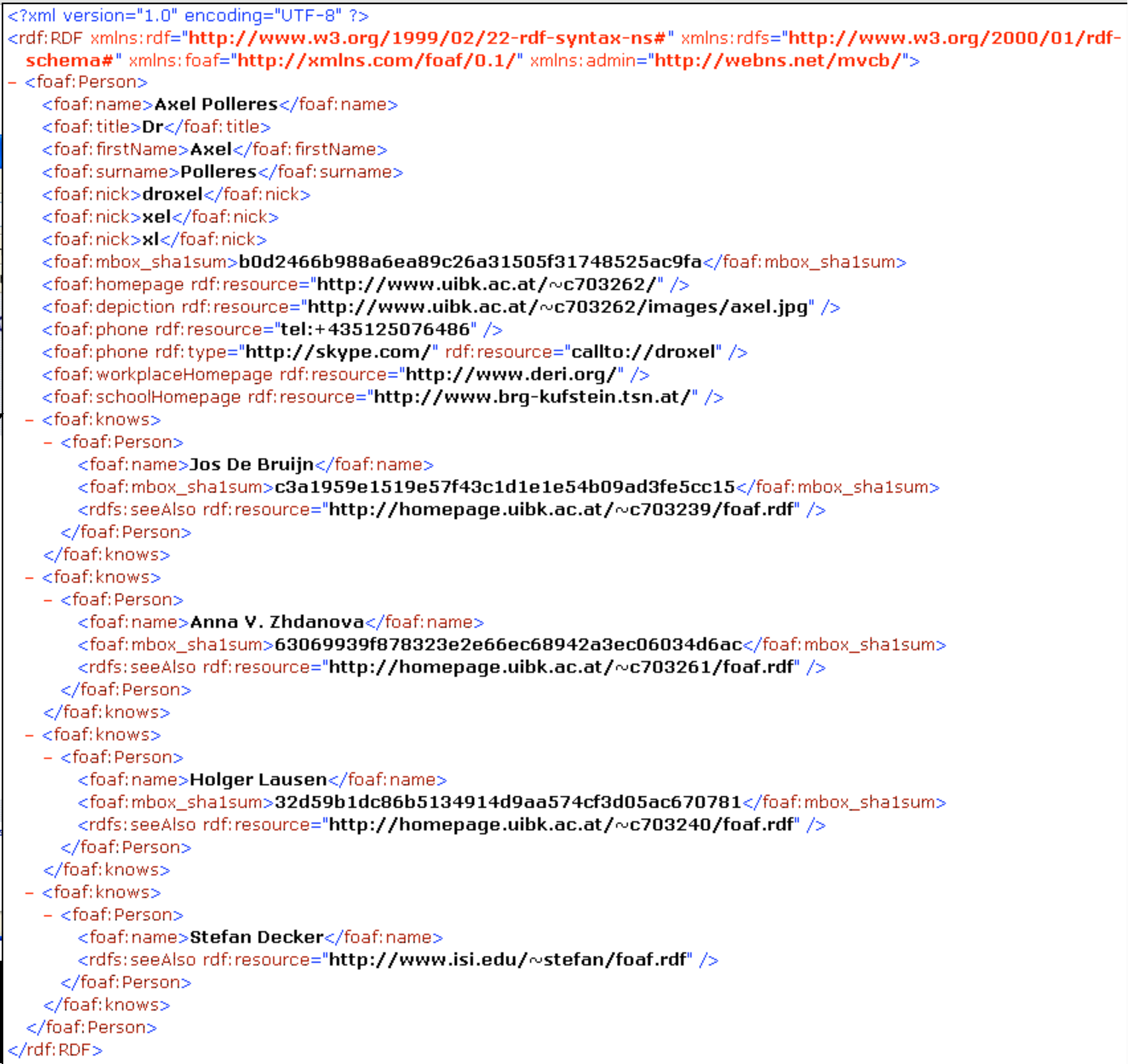
+43 512 507 9872

Axel.Polleres[at]uibk.ac.at

Institute of Computer Science
University of Innsbruck
TechnikerStraße 13
A-6020 Innsbruck
Austria

Axel Polleres

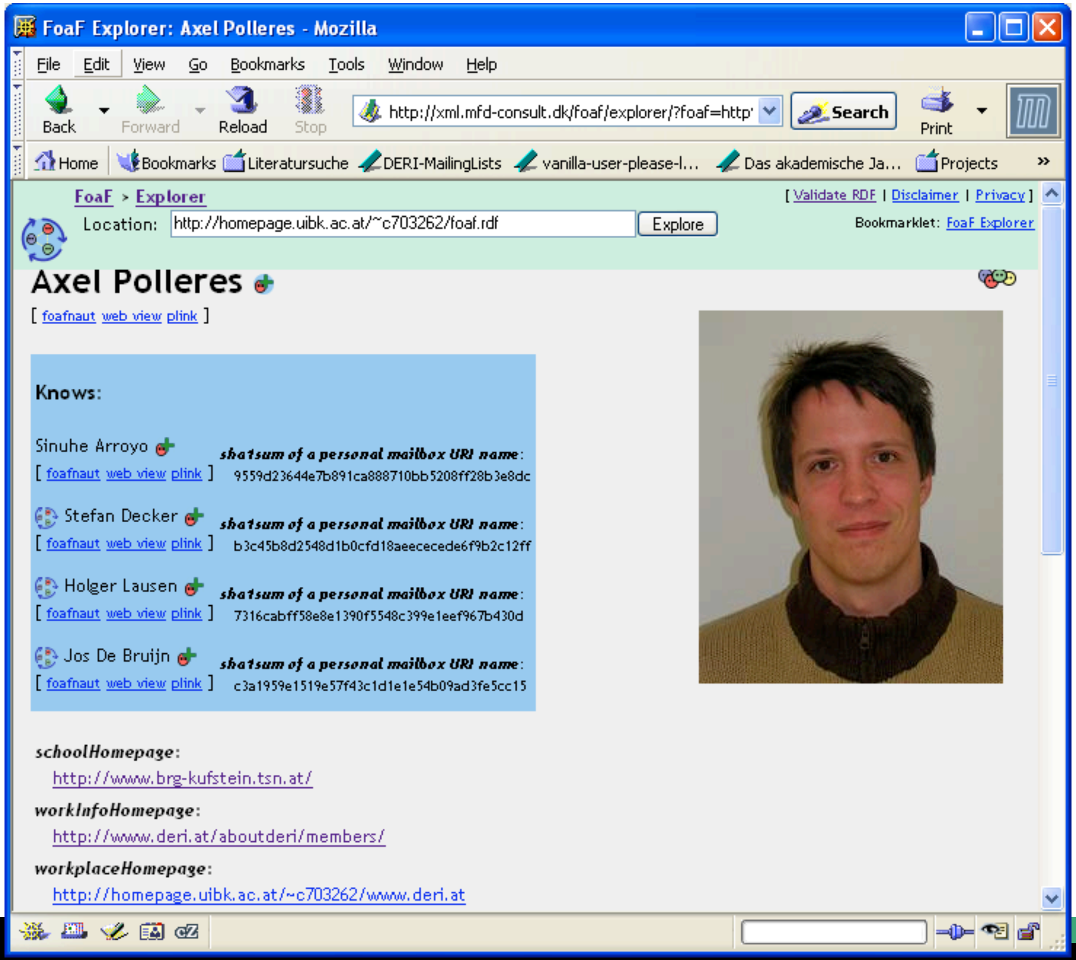
Curriculum Vitæ | Research | Publications | Teaching | Links



```
<?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:foaf="http://xmlns.com/foaf/0.1/" xmlns:admin="http://webns.net/mvcb/">
- <foaf:Person>
    <foaf:name>Axel Polleres</foaf:name>
    <foaf:title>Dr</foaf:title>
    <foaf:firstName>Axel</foaf:firstName>
    <foaf:surname>Polleres</foaf:surname>
    <foaf:nick>droxel</foaf:nick>
    <foaf:nick>xel</foaf:nick>
    <foaf:nick>xl</foaf:nick>
    <foaf:mbox_sha1sum>b0d2466b988a6ea89c26a31505f31748525ac9fa</foaf:mbox_sha1sum>
    <foaf:homepage rdf:resource="http://www.uibk.ac.at/~c703262/" />
    <foaf:depiction rdf:resource="http://www.uibk.ac.at/~c703262/images/axel.jpg" />
    <foaf:phone rdf:resource="tel:+435125076486" />
    <foaf:phone rdf:type="http://skype.com/" rdf:resource="callto://droxel" />
    <foaf:workplaceHomepage rdf:resource="http://www.deri.org/" />
    <foaf:schoolHomepage rdf:resource="http://www.brg-kufstein.tsn.at/" />
  - <foaf:knows>
    - <foaf:Person>
        <foaf:name>Jos De Bruijn</foaf:name>
        <foaf:mbox_sha1sum>c3a1959e1519e57f43c1d1e1e54b09ad3fe5cc15</foaf:mbox_sha1sum>
        <rdfs:seeAlso rdf:resource="http://homepage.uibk.ac.at/~c703239/foaf.rdf" />
      </foaf:Person>
    </foaf:knows>
  - <foaf:knows>
    - <foaf:Person>
        <foaf:name>Anna V. Zhdanova</foaf:name>
        <foaf:mbox_sha1sum>63069939f878323e2e66ec68942a3ec06034d6ac</foaf:mbox_sha1sum>
        <rdfs:seeAlso rdf:resource="http://homepage.uibk.ac.at/~c703261/foaf.rdf" />
      </foaf:Person>
    </foaf:knows>
  - <foaf:knows>
    - <foaf:Person>
        <foaf:name>Holger Lausen</foaf:name>
        <foaf:mbox_sha1sum>32d59b1dc86b5134914d9aa574cf3d05ac670781</foaf:mbox_sha1sum>
        <rdfs:seeAlso rdf:resource="http://homepage.uibk.ac.at/~c703240/foaf.rdf" />
      </foaf:Person>
    </foaf:knows>
  - <foaf:knows>
    - <foaf:Person>
        <foaf:name>Stefan Decker</foaf:name>
        <rdfs:seeAlso rdf:resource="http://www.isi.edu/~stefan/foaf.rdf" />
      </foaf:Person>
    </foaf:knows>
  </foaf:Person>
</rdf:RDF>
```

# Tools for FOAF:

- FoaF-a-Matic (http://www.ldodds.com/foaf/foaf-a-matic.html a FoaF generator)
- FoaF-Explorer (http://xml.mfd-consult.dk/foaf/explorer/) a Foaf Browser,
- now also Google, see e.g. (http://code.google.com/apis/socialgraph/ ), indexes FOAF and XFN microformat


FoaF Explorer: Axel Polleres - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://xml.mfd-consult.dk/foaf/explorer/?foaf=http'
Search
Print
Home Bookmarks Literatursuche DERI-MailingLists vanilla-user-please-l... Das akademische Ja... Projects
FoaF > Explorer
[ Validate RDF | Disclaimer | Privacy ]
Location:
Explore
Bookmarklet: FoaF Explorer

Location: http://homepage.uibk.ac.at/~c703262/foaf.rdf

## Axel Polleres

[ foafnaut web view plink ]

**Knows:**

Sinuhe Arroyo
[ foafnaut web view plink ]
***sha1sum of a personal mailbox URI name***: 9559d23644e7b891ca888710bb5208ff28b3e8dc

Stefan Decker
[ foafnaut web view plink ]
***sha1sum of a personal mailbox URI name***: b3c45b8d2548d1b0cfd18aeececede6f9b2c12ff

Holger Lausen
[ foafnaut web view plink ]
***sha1sum of a personal mailbox URI name***: 7316cabff58e8e1390f5548c399e1eef967b430d

Jos De Bruijn
[ foafnaut web view plink ]
***sha1sum of a personal mailbox URI name***: c3a1959e1519e57f43c1d1e1e54b09ad3fe5cc15

***schoolHomepage***:
http://www.brg-kufstein.tsn.at/

***workInfoHomepage***:
http://www.deri.at/aboutderi/members/

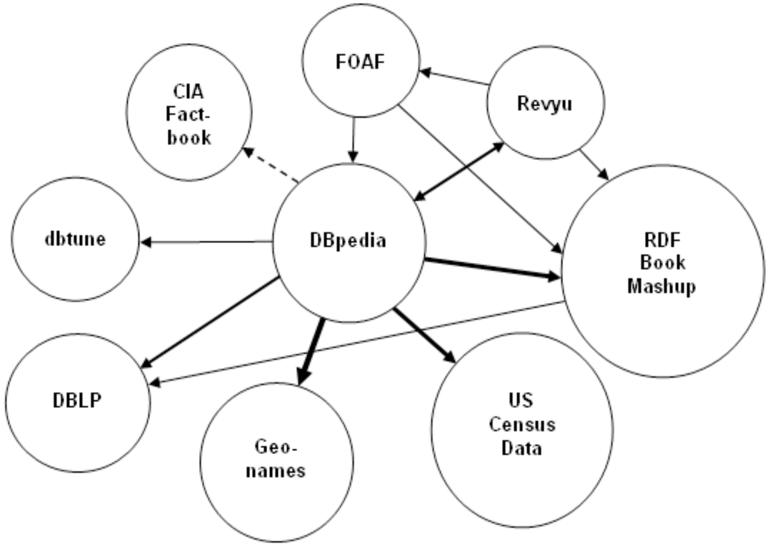
***workplaceHomepage***:
http://homepage.uibk.ac.at/~c703262/www.deri.at

- Format to annotate and link Blogs, blogposts, mailinglists, forums, etc.
- Reuses FOAF, Dublin Core, etc.

- Useful to extract and merge blogposts, mailinglist forums, etc

# Open Linked Data

- DBPedia: FU Berlin and others export huge ammounts of metadata from Wikipedia, DBLP, and other sources
- Reuters allows export RDF! Calais API http://slashdot.org/article.pl?sid=08/02/10/2041235&from=rss
- Google indexing foaf and XFN
- Microformats are gaining momentum
- W3C founded a working group for getting RDF out of XML or HTML (GRDDL): First proposals based on XSLT.



# Open Linked Data:
# The Semantic Web as a "Quad" space:

- A semantic model (RDF) can be published on the web or be accessible via a (GRDDL) wrapper, at a specific web location (URL):

  Resolve(http://g1o.net/foaf.rdf) → rdf/xml

- The collection of all the RDF models published on the web is today referred to by some as being *the* **"Semantic Web"**

- The Semantic Web made up of all this data can be seen as a potentially infinite set of quadruples:

  ***L****ocation* **S**ubject **O**bject **P**redicate

- The Web therefore hosts an infinite, shared ownership quad store:

  - Most of the graphs are immediately readable
  - Writing is possible, but limitation on the **L** (e.g. one's homepage)

# Tools:

- Browsers: Tabulator, Disco (browsers) etc..
- Centralized services: SWSE (for full queries) Andreas Harth - DERI, Sindice (find the sources, query yourself) Giovanni Tummarello - DERI
- Aggregators/processors: Semantic Web Pipes (see following slides)