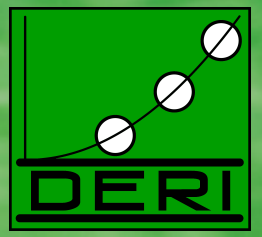


Introduction to Semantic Web: ideas, platforms and APIs

Dr. Giovanni Tummarello
DERI, Galway

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh

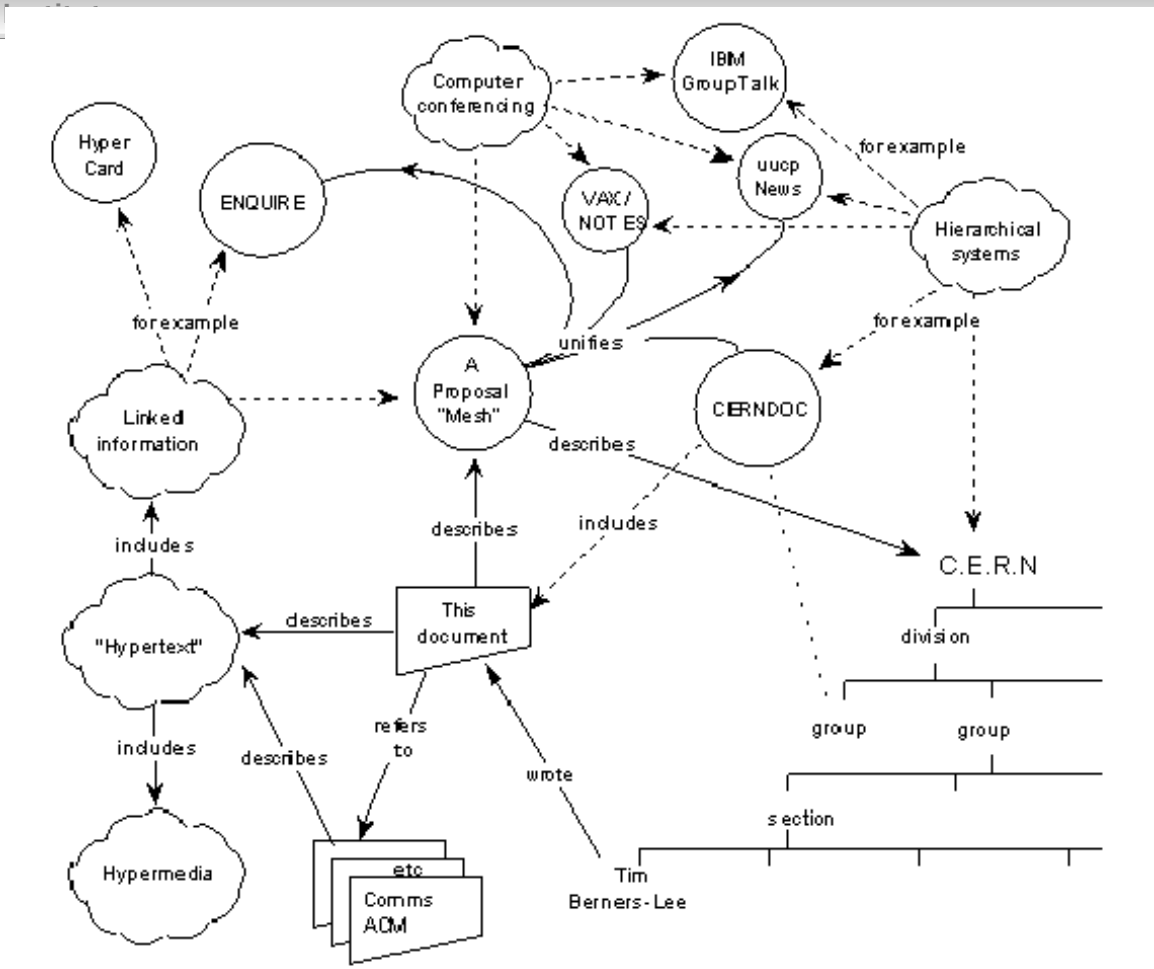
sfi
science foundation ireland
fondúireacht eolaíochta éireann

Introduction

- On the web today:
  - over 10 billions reusable, pieces of information
  - offered across 1+ billion web locations.
- Enabled by RDF and by Microformats
- Called the Semantic Web, or Web of Data, or 3.0
- Just the tip of what is to come.
  - See recent Yahoo announcement for Microformat Support
  - See google Social API (supports Microformats + RDF), Reuters OpenCalais (RDF), Myspace (announced), Linkedin, LastFM, BBC
  - New platforms: Twine (RDF), FreeBase, etc..

Part 1: crash course on the W3C Semantic Web technology stack

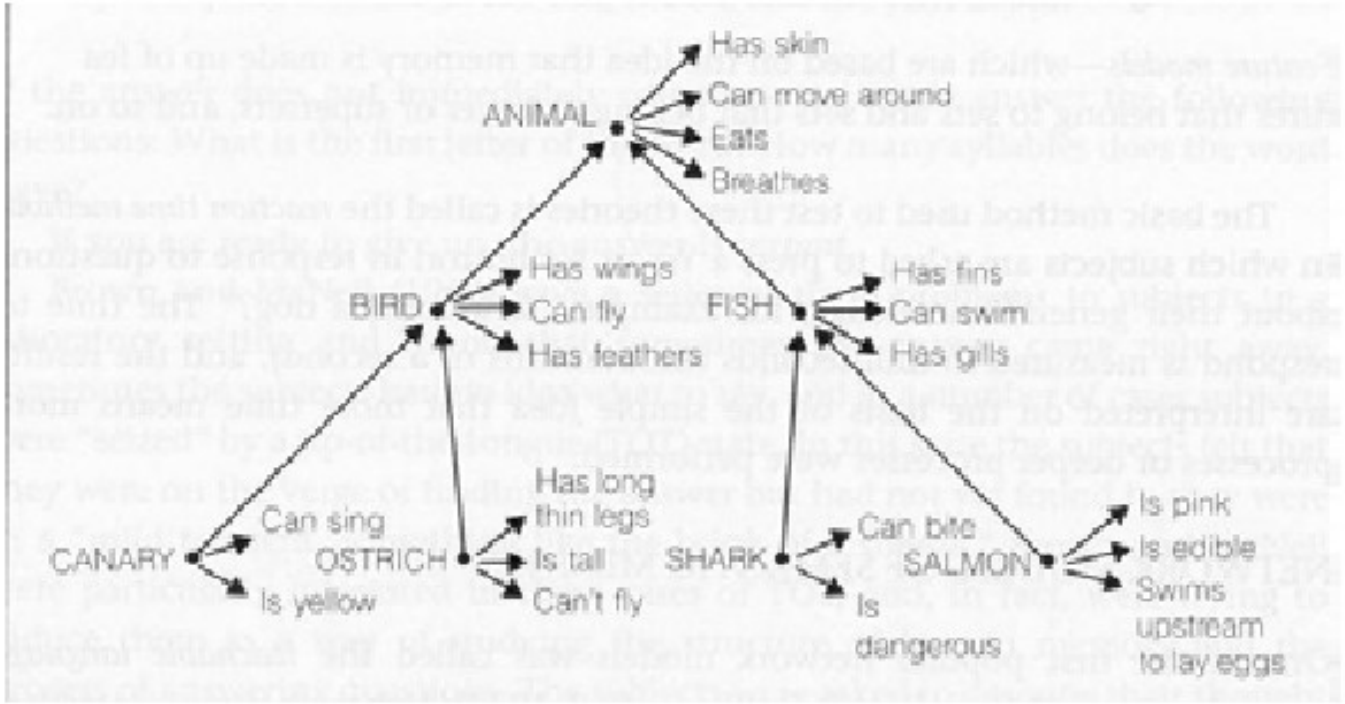
A quick look at the semantic web:



The “origins”: Tim Berners-Lee, CERN
March 1989 Information Management: A Proposal

Much earlier: semantic networks

Example of a taxonomic tree (Collins & Quillian)



The “Web” in the “Semantic Web”

Mentioning “web” implies a whole scenario:

- Distributed, Machine accessible
  - http://g1o.net/foaf.rdf
- Based on open standards, not implementations
- Technically imperfect (but live with it – 404!)
- Potentially unfriendly (live with it and/or take measures)

Basic instruments: URIs

- **U**niform **R**esource **I**dentifiers
  - To identify “things” that will be subject, object or predicate of our statements.
- More than “URLs”, e.g. :
  - A book:

    ISBN:1234-123443-1234343
  - Some bytes:

    SHA1:56547383A6747B667F443C4D
  - My cat:

    TAG:giovanni@wup.it/25-12-2007/mycat

RDF – expressing relationships between resources

Hypothesis: we can express knowledge as a directed and labeled graph, the basic unit being a STATEMENT (a triple) linking (usually) RESOURCES:

Subject → Predicate → Object

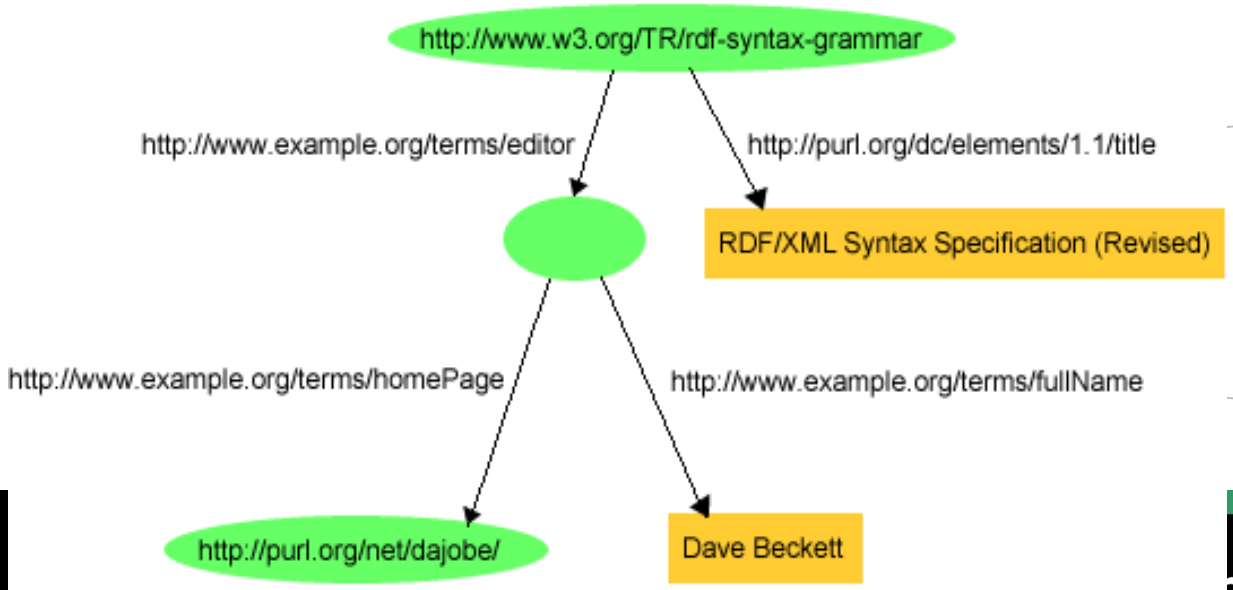
Observation:

- Seems nice and uniform
- More complex statements can be expressed as aggregations of simpler ones.

Thinking in RDF


```
<?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.example.org/index.html"> <exterms:creation-date>August 16,
      1999</exterms:creation-date>
      </rdf:Description>
      <rdf:Description rdf:about="http://www.example.org/index.html">
            <dc:language>en</dc:language>
      </rdf:Description>
      <rdf:Description rdf:about="http://www.example.org/index.html">
            <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
      </rdf:Description>
</rdf:RDF>
```

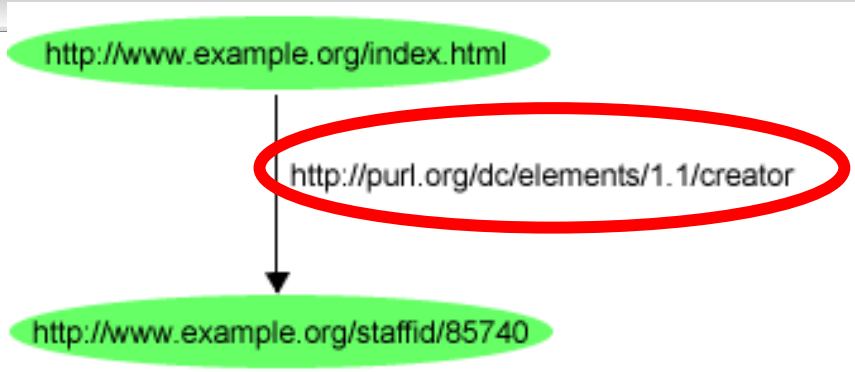
…forget the XML serialization, RDF is a graph:



Looks powerful already!

- If we liked XML (Tree structures are a superset of plain (DB) tables)
- ..We got to love RDF (Graphs are supersets of trees)
- Standard serializability
- Actually simpler (no “attributes” like things)

Vocabularies, schemas, ontologies



- Properties are resources themselves, so they have a URI!
  - *..to give a precise meaning to a relation*
  - *..to allow others to reuse the very same relation*
  - *..rules to specify how to use such relations*
- *Resources are usually typed (belong to a specific Class)*
- The definition of Classes and Properties are specified in an Ontology
- Ontologies are written in RDF themselves
- Ontologies can be put online and automatically imported by applications

Namespaces and URI refs

http://example.org/ontology#Concept1

Can be written as ex:Concept1(URI ref)

Where ex = http://example.org/ontology#

- Used to specify which vocabulary (ontology) the term belongs to

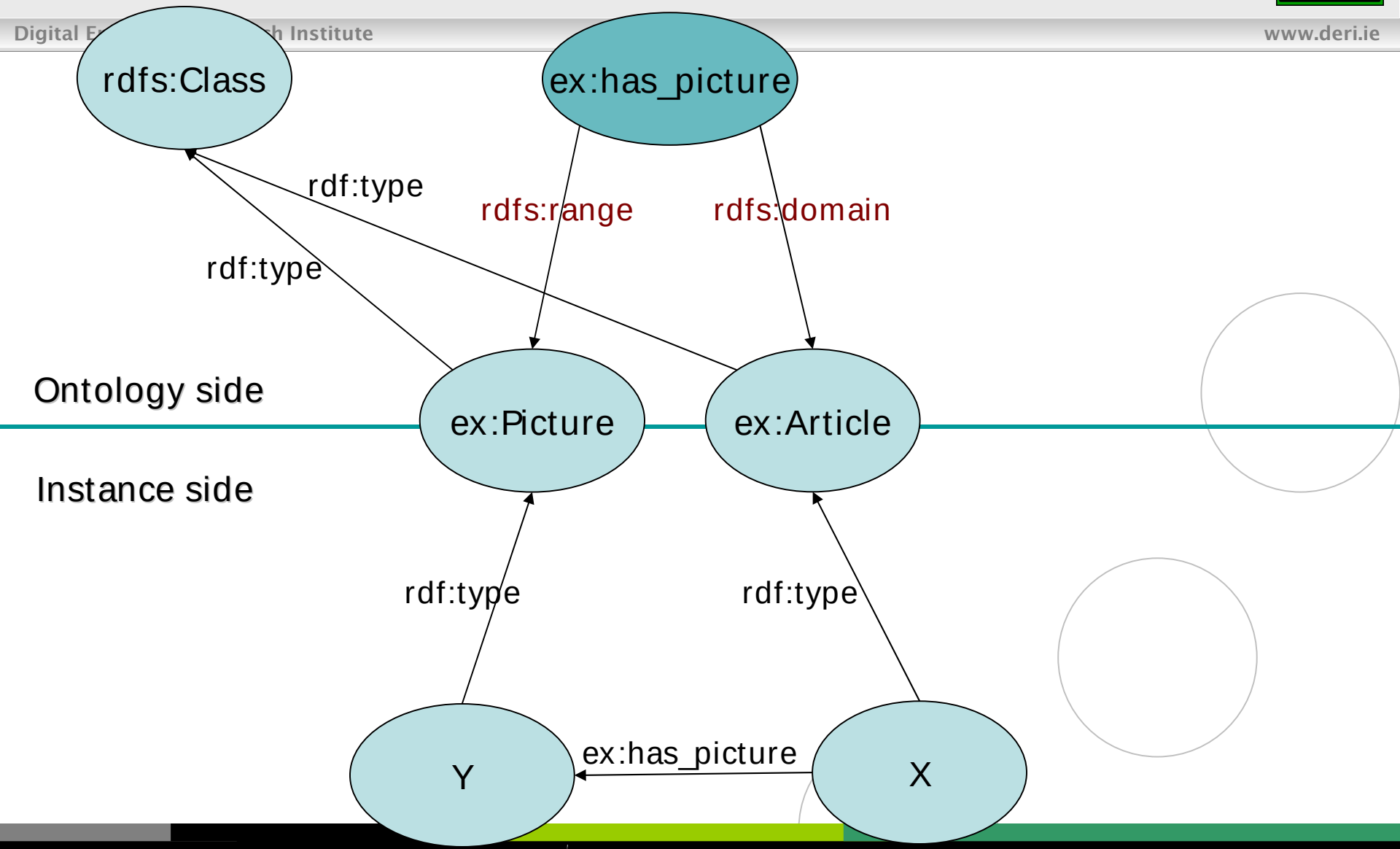
RDF Schema (RDFS)

- Simple language for writing Ontologies

RDF Schema (RDFS)

- Classes definition
  <disc:Philosopher rdf:type rdfs:Class>
- Properties definition
  <disc:collaborates rdf:type rdfs:Property>
- Hierarchies of Classes
  <disc:Philosopher rdfs:subClassOf foaf:Person>
- Property's domain and range
  <disc:collaborates rdfs:domain disc:Philosopher>
  <disc:collaborates rdfs:range disc:Philosopher>
- Hiearchies of Properties
  <disc:collaborates rdfs:subPropertyOf foaf:knows>

RDF and RDFSchema together



Ontology Web Language (OWL)

- Based on Description Logics
- Can be written in RDF as well
- More powerful constructs
- Not always conpatible with RDFS
  - Classes and instances are strictly disjoint

Ontology Web Language (OWL)

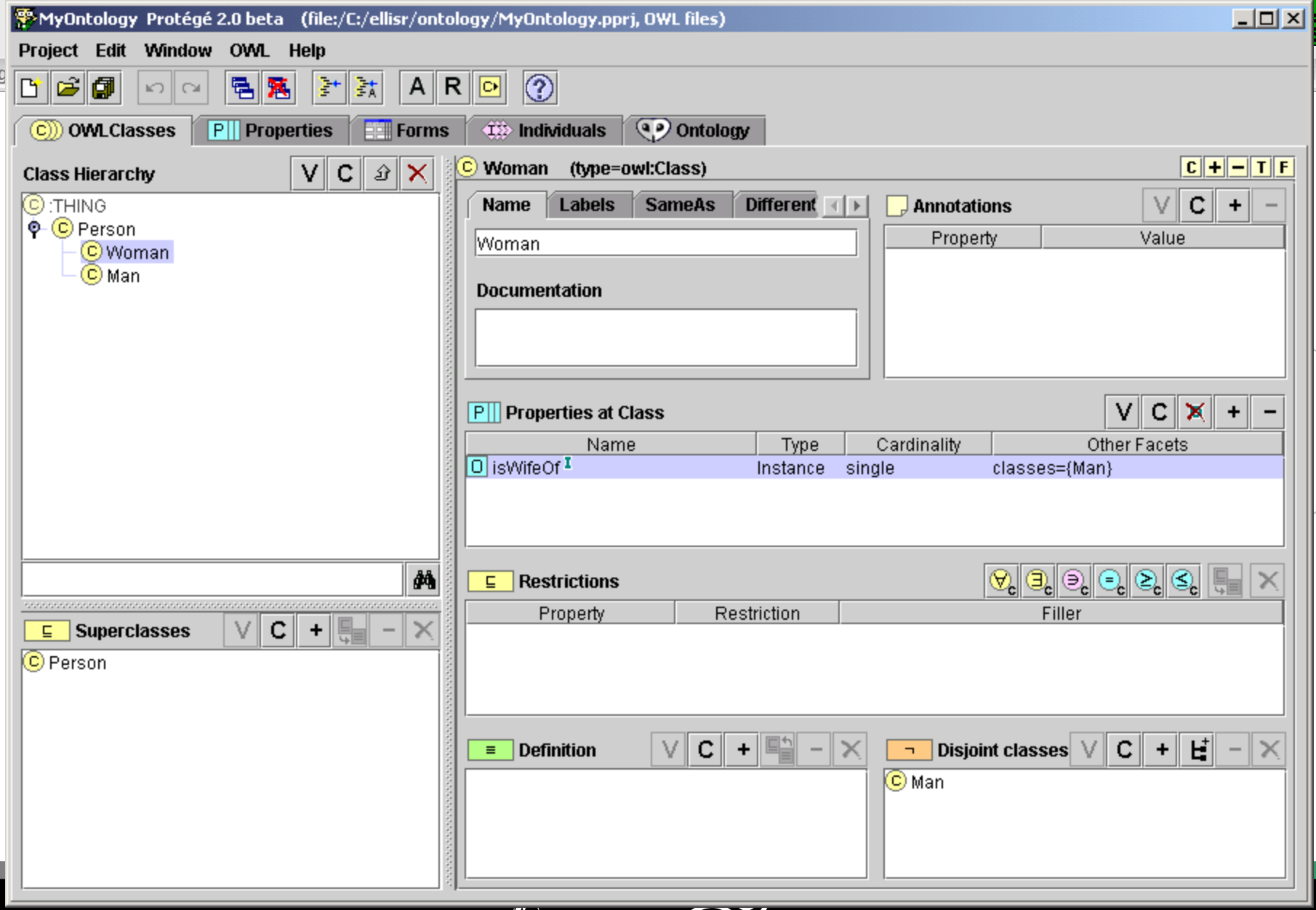
- owl:sameAs
  - Saying that a resource is equivalent to an other one
- owl:inverseFunctionalProperty
  - The value of a property univocally identifies a resource
- owl:transitiveProperty
  - If <A p B> and <B p C> then <A p C>
- Cardinality constraints
  - A person can have only one mother, at least one name, etc.
- ….

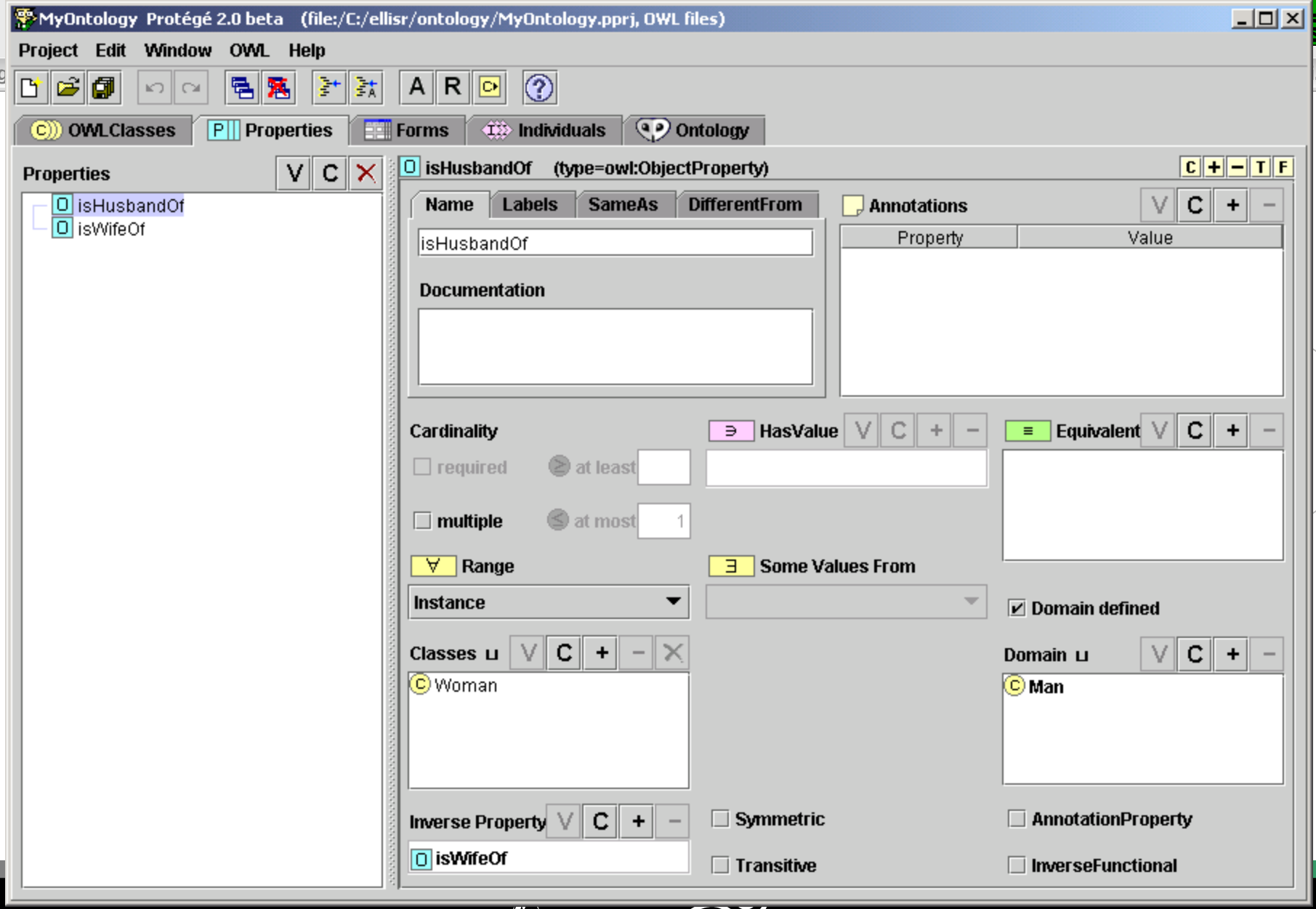
Design best practices: reuse existing ontologies

- A lot of ontologies have been already developed for different frequent domains
- Search on: SchemaWeb.info, Swoogle …
- Ontologies can be easily extended
  - You can reuse classes or properties
- Added Value: chance of interoperability

OWL Example

- Class
  - Person superclass
  - Man, Woman subclasses
- Properties
  - isWifeOf, isHusbandOf
- Property characteristics, restrictions
  - inverseOf
  - domain
  - range
  - Cardinality
- Class expressions
  - disjointWith





OWL Example In XML


```
<rdf:RDF
    xmlns="http://owl.protege.stanford.edu#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about="">
    <owl:imports
   rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Person"/>
  <owl:Class rdf:ID="Man">
    <rdfs:subClassOf rdf:resource="#Person"/>
    <owl:disjointWith>
      <owl:Class rdf:about="#Woman"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:ID="Woman">
    <owl:disjointWith rdf:resource="#Man"/>
    <rdfs:subClassOf rdf:resource="#Person"/>
  </owl:Class>
```

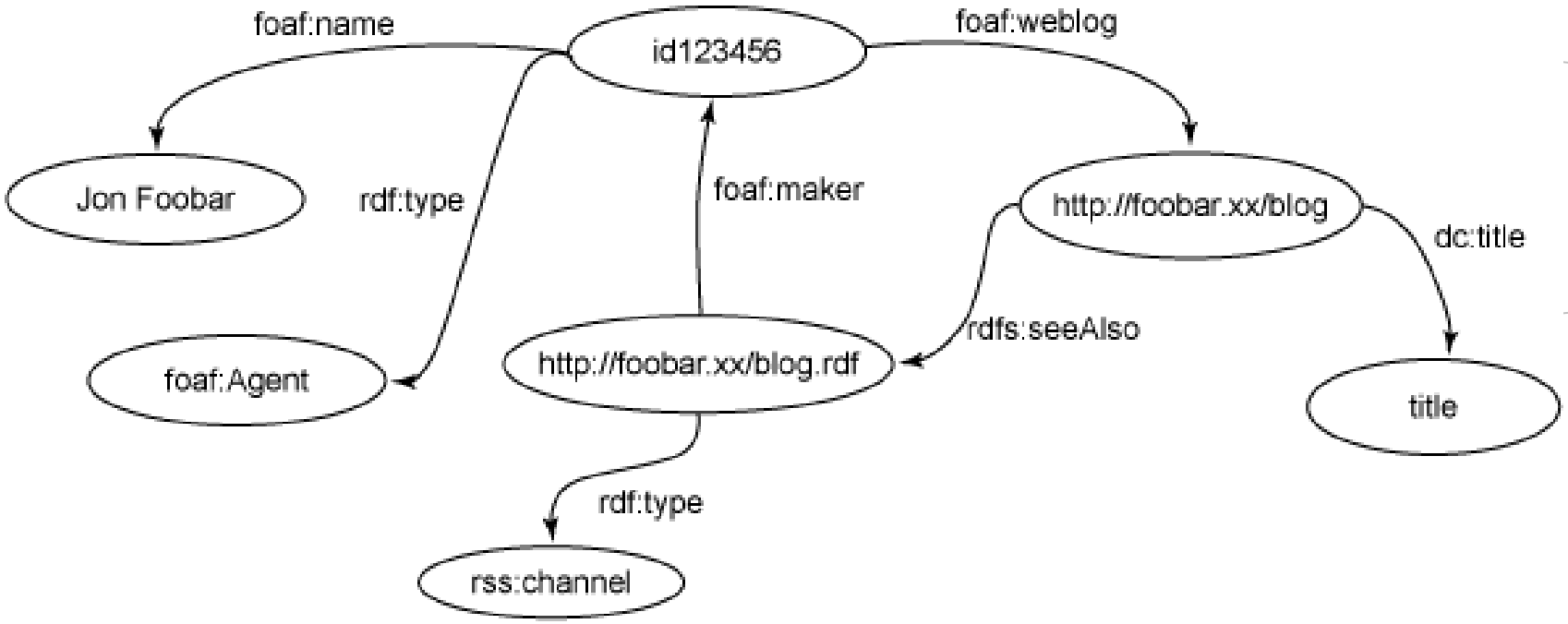
OWL Example In XML (cont.)


```
<owl:ObjectProperty rdf:ID="isHusbandOf"
     rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
    <rdfs:domain rdf:resource="#Man"/>
    <rdfs:range rdf:resource="#Woman"/>
    <owl:inverseOf rdf:resource="#isWifeOf"/>
    <owl:minCardinality>0</owl:minCardinality>
    <owl:maxCardinality>1</owl:maxCardinality>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isWifeOf"
     rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
    <rdfs:domain rdf:resource="#Woman"/>
    <rdfs:range rdf:resource="#Man"/>
    <owl:inverseOf rdf:resource="#isHusbandOf"/>
    <owl:minCardinality>0</owl:minCardinality>
    <owl:maxCardinality>1</owl:maxCardinality>
  </owl:ObjectProperty>
</rdf:RDF>
```

Querying RDF

- W3C has designed and standardized a language for querying RDF: SPARQL

Example (RDFPlanet.org blogger model)



“Find the URL of the blog by the person named Jon Foobar”

- Query:

```
PREFIX foaf: http://xmlns.com/foaf/0.1/

 SELECT ?url FROM <bloggers.rdf>

WHERE {
          ?contributor foaf:name "Jon Foobar".
          ?contributor foaf:weblog ?url.
}
```

- Results:

```
$ sparql --query jon-url.rq
-------------------------------------------
| url                                     |
===========================================
| <http://foobar.xx/blog>                 |
```

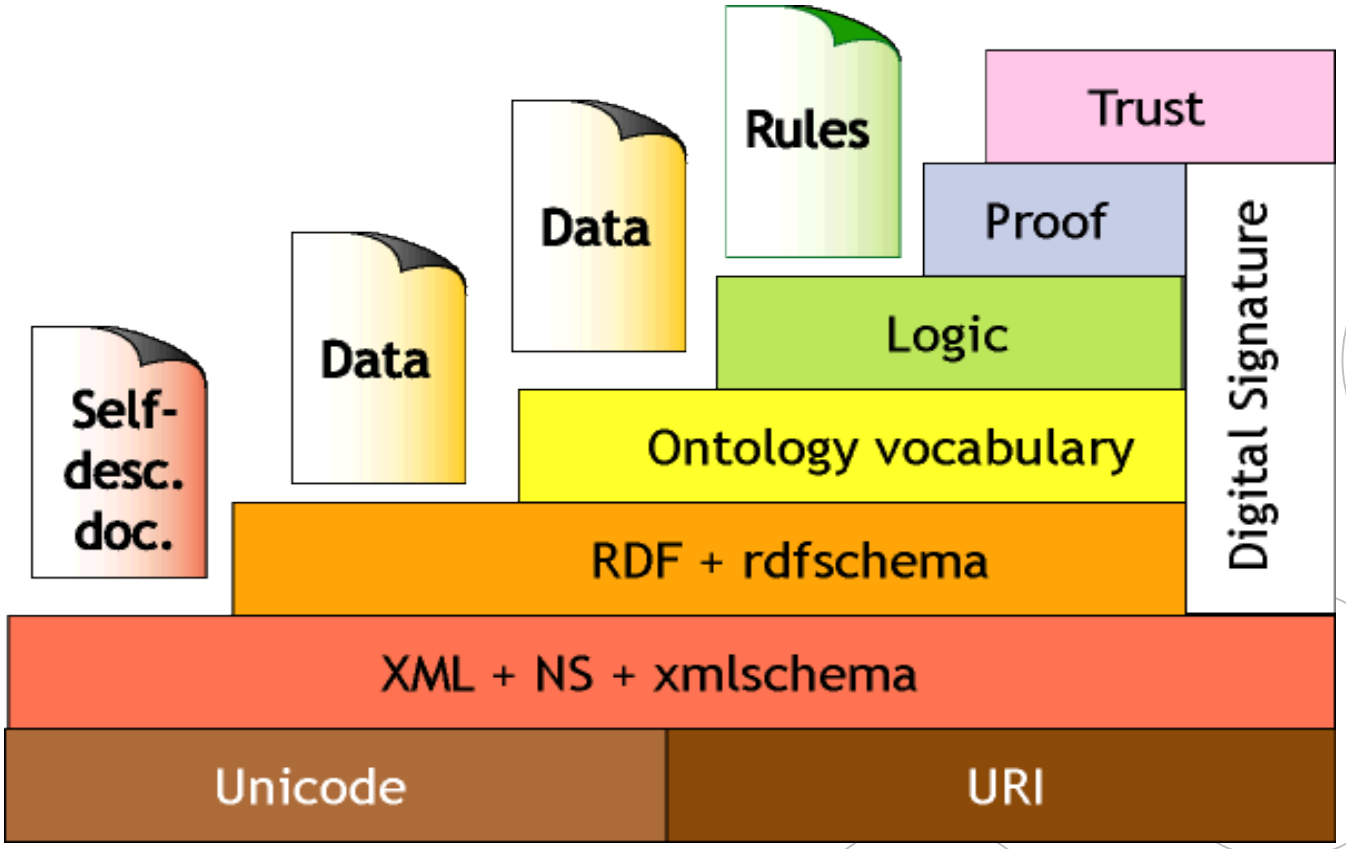
In Java


```
// Create an empty in-memory model and populate it from the graph
Model model = ModelFactory.createMemModelMaker().createModel();

// Open the bloggers RDF graph from the filesystem
InputStream in = new FileInputStream(new File("bloggers.rdf"));

model.read(in,null); // null base URI, since model URIs are absolute
in.close();

// Create a new query
String queryString =
      "PREFIX foaf: <http://xmlns.com/foaf/0.1/> " +
      "SELECT ?url " +
      "WHERE {" +
      "      ?contributor foaf:name \"Jon Foobar\" . " +
      "      ?contributor foaf:weblog ?url . " +
      "      }";

Query query = QueryFactory.create(queryString);

// Execute the query and obtain results
QueryExecution qe = QueryExecutionFactory.create(query, model);
ResultSet results = qe.execSelect();

// Output query results
ResultSetFormatter.out(System.out, results, query);
```

More advanced queries

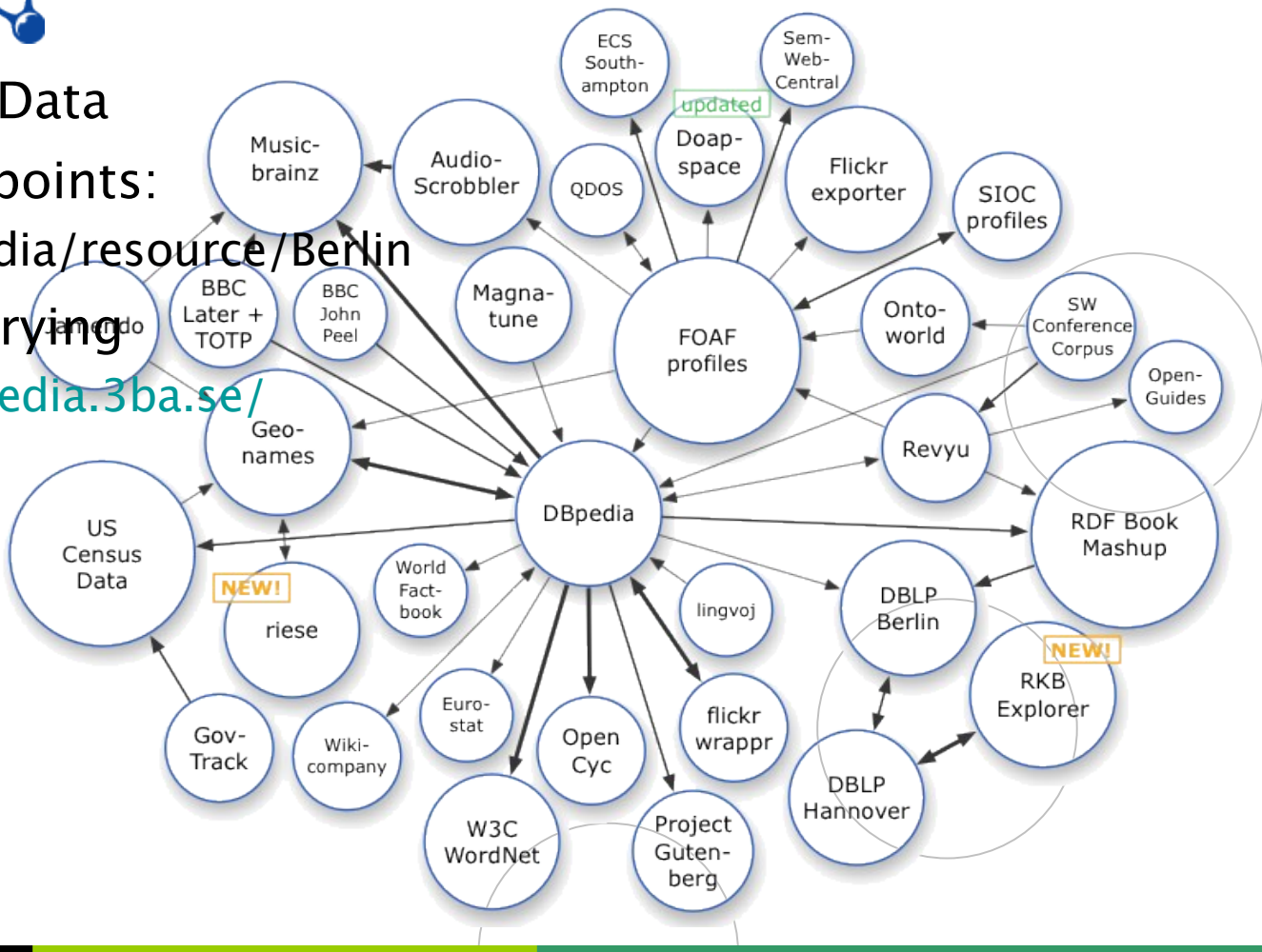

```
PREFIX rss:   <http://purl.org/rss/1.0/>
PREFIX xsd:   <http://www.w3.org/2001/XMLSchema#>
PREFIX  dc:   <http://purl.org/dc/elements/1.1/>
SELECT ?item_title ?pub_date ?depiction

WHERE {
      ?item rss:title ?item_title .
      ?item dc:date ?pub_date .
        OPTIONAL { ?item foaf:depiction ?depiction . }.

      FILTER xsd:dateTime(?pub_date) >= "2005-04-
        01T00:00:00Z"^^xsd:dateTime &&
             xsd:dateTime(?pub_date) < "2005-05-
   01T00:00:00Z"^^xsd:dateTime
      }
```

The Semantic Web tower



How much RDF out there?

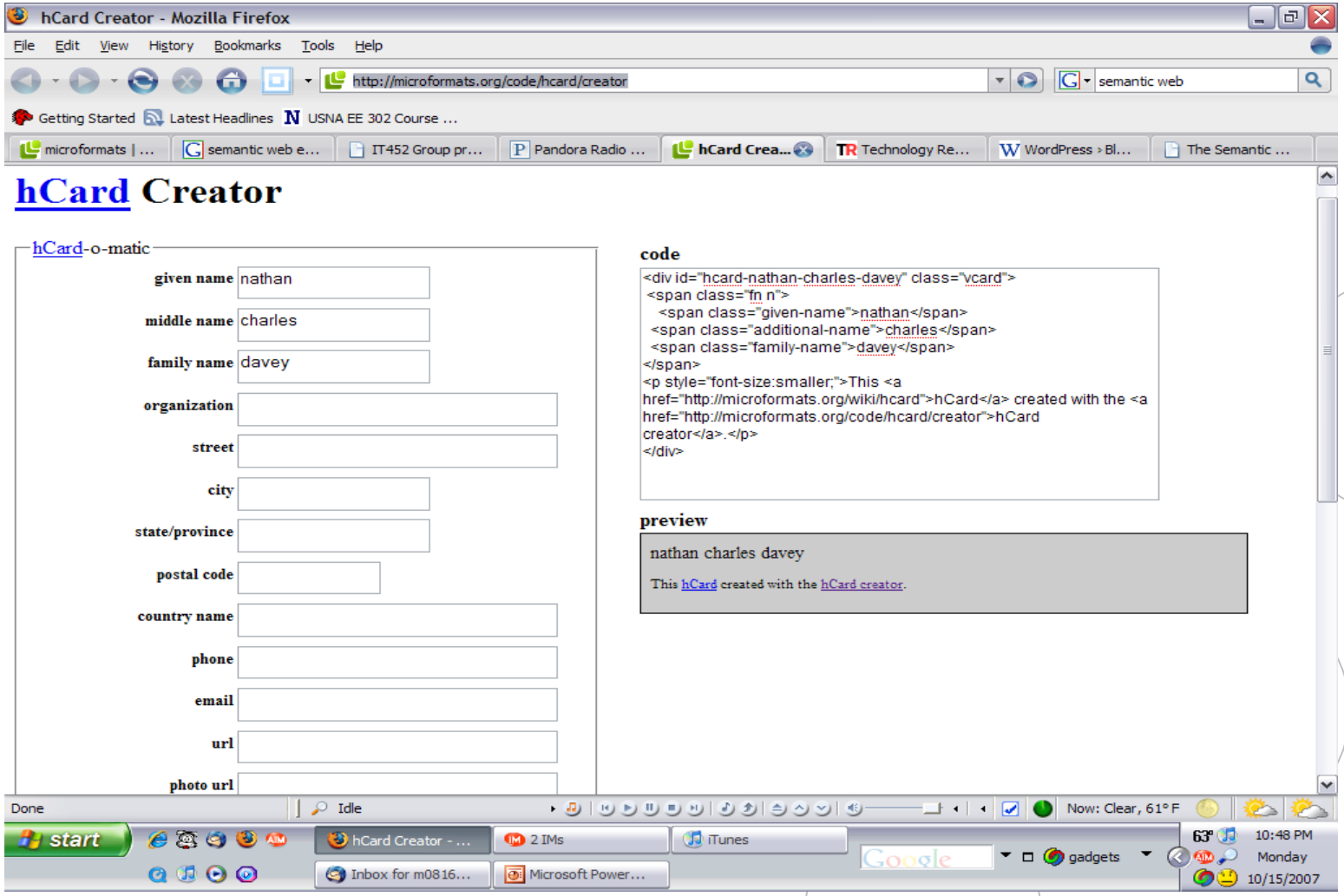
- RDF World:
  - Linking Open Data
  - URIs as entry points:
    - http://dbpedia/resource/Berlin
  - Advanced querying
    - http://wikipedia.3ba.se/



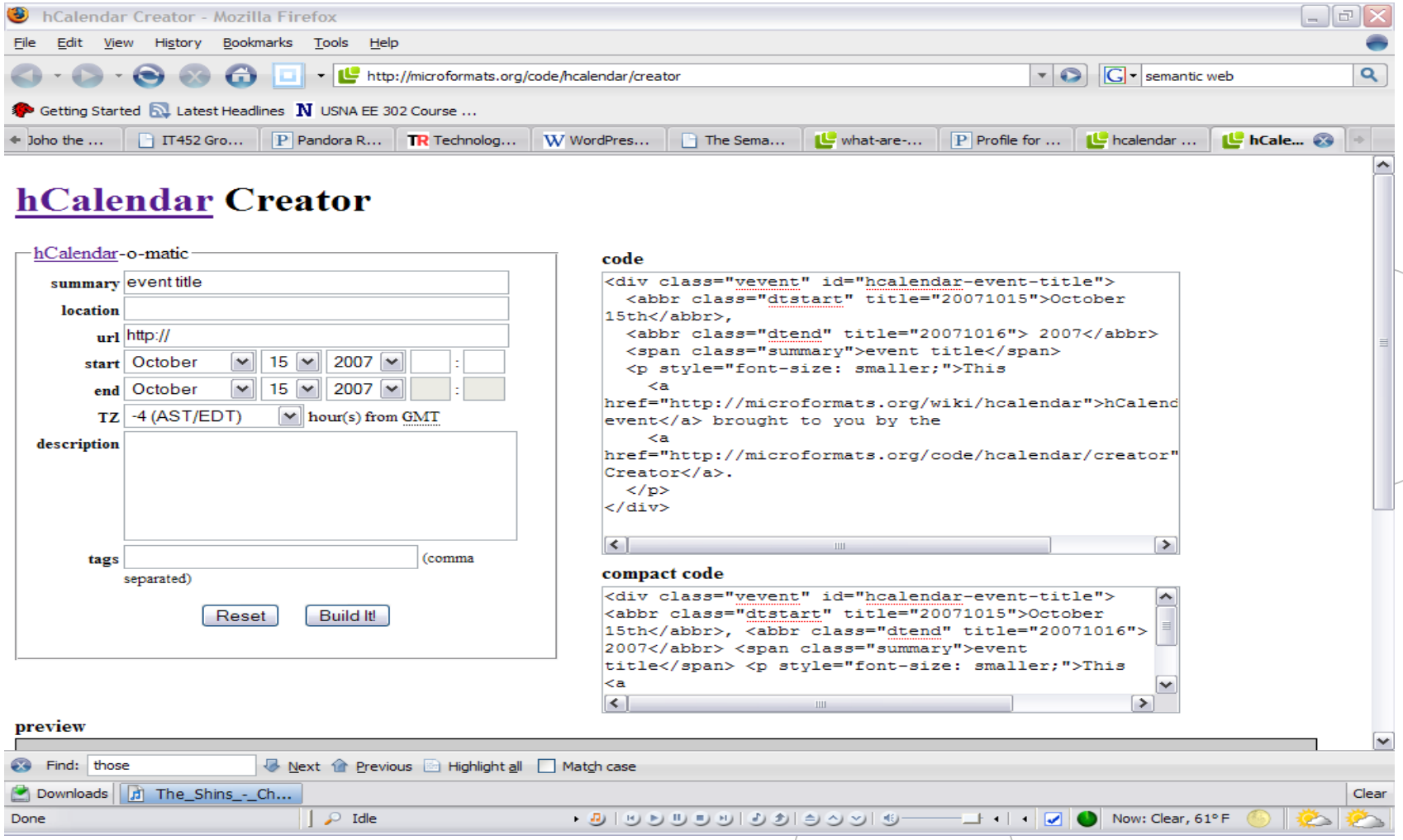
Microformats

- The Web 2.0 developer approach to a Web of Data
- HTML is enhanced with Microformats
- Microformats are “invisible tags” that make it easy to share and reuse data in your webpages
- For Instance, VCard Example.
  - For example, to populate an address book, browse social relationships, share reviews, tag content.

Card Example:



Calendar Example



hCalendar-o-matic

summary: IT452 Presentation
location: USNA
url: http://cs.usna.edu/~lmcdowell
start: October 22 2007 08 : 00
end: October 23 2007 09 : 00
TZ: -5 (EST/CDT) hour(s) from GMT
description: Presentation on microformats
tags: IT, Computers, Web Design (comma separated)

Reset | Build It!

code

```
<div class="vevent"
id="hcalendar-IT452-Presentation">
  <a class="url"
href="http://cs.usna.edu/~lmcdowell">
    <abbr class="dtstart"
title="20071022T0800-0500">October 22th
08am</abbr>,
    <abbr class="dtend" title="20071023T900-0500">
9am 2007</abbr> &mdash;
    <span class="summary">IT452
Presentation</span>&mdash; at
    <span class="location">USNA</span>
  </a>
  <div class="description">Presentation on
microformats</div>
```

preview

October 22th 08am, 9am 2007 — IT452 Presentation— at USNA
Presentation on microformats
Tags: IT Computers Web Design

This hCalendar event brought to you by the hCalendar Creator.

Web Examples

- LastFM events
- Linkedin profiles
- Kelkoo product listings

Rules of Microformats Club:

- 1st Rule: Don't create microformats
- 2nd Rule: Re-use already made microformats
  - If you can't find one, you haven't done enough research

Existing Microformats

- Specifications
  - hCalendar – hCalendar creator (*http://microformats.org/code/hcalendar/creator*)
  - hCard – hCard creator (*http://microformats.org/code/hcard/creator*)
  - rel-license
  - rel-nofollow
  - rel-tag
  - VoteLinks
  - XFN – XFN creator (*http://gmpg.org/xfn/creator*)
  - XMDP
  - XOXO

- **Drafts**
  - adr – for marking up address information
  - geo – for marking up WGS84 (*http://en.wikipedia.org/wiki/WGS84*) geographic coordinates (latitude; longitude)
  - hAtom – syndicating episodic content (e.g. weblog postings)
  - hResume – for publishing resumes and CVs
  - hReview – hReview creator (*http://microformats.org/code/hreview/creator*)
  - rel-directory – to indicate that the destination of a hyperlink is a directory listing containing an entry for the current page
  - rel-ecolabel – for indicating ecolabelled products/services/companies
  - rel-enclosure – for indicating attachments (e.g. files) to download and cache
  - rel-home – indicate a hyperlink to the homepage of the site
  - rel-payment – indicate a payment mechanism
  - robots exclusion
  - xFolk

What to use the Web of Data for.

- *Consuming* the Data Web:
  - To increase the internal page views:
    - Import content (news, media, events, product listings)
    - Finding out unusual connections between your pages
  - To increase advertisement revenues
    - Finding related keywords for more interesting ads.
    - Enhanced social networking
  - To help close sales
    - Proposing positive reviews, additional media.
    - Showing the user there exist a community of people who have that product
  - Community use cases using open RDF
    - Supported right away!
- *Participating* to the Web of Data
  - Participating break monopoly, reaches the long tail.

Tools you can use right away

For Developers:

- Sindice: locate structured data sources in the open web
- Semantic Web Pipes: remix data on the fly
- Semantic Sitemaps: effectively publish large amount of semantic data

Sindice: *Data Web* Services

- Sindice Monitors, Indexes, Cleans, Filters, Process.. and Remixes the Web of Data
- Provides clean high level APIs to reuse Web Data
  - Domain specific
  - General purpose
- Helps publish effectively on the Web of Data
- Community of people who want to reuse web data

Which API?

- Find Semantic Sources which:
  - Talk about something which has specific properties
  - Talk about the relationship between something and something else, mention a term Etc..
- High Level APIs
  - SIOC api (user messages on the web)
  - Next releases: Social Graph API++ , products, events etc.
- Summarize
  - Microformats and RDF are treated and served as one
- Ranking, Spam filtering, built in

Sindice Core Technologies

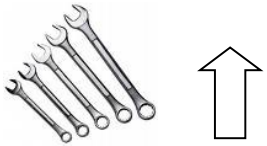
- Tetascale Semantic Index Layer
  - *Trillion* of semantic statements, simple but useful queries
- Hadoop based large Scale Semantic Processing
  - Reasoning on each semantic source
  - Deep semantic Microformat integration
  - Ranking calculation
- Gigascale Semantic advanced index cluster
  - A cloud of *Billion* scale semantic stores
  - Advanced queries for (refinements, high level API).
- Metadata focused crawling cluster
  - With generic crawler integration

Sindice Technologies (overview)

Semantic Sitemaps

Feeds, Notifications

API + Developer Frontend



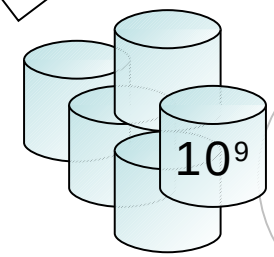
Crawler

Processing Reasoning

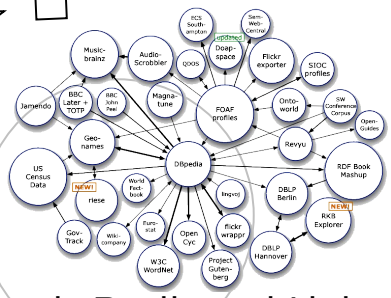
Storage Data Analisys

Domain API Logic

10^{12} Index



10^{9}



Cloud of conventional Semantic stores

Replicated Linked open data

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh

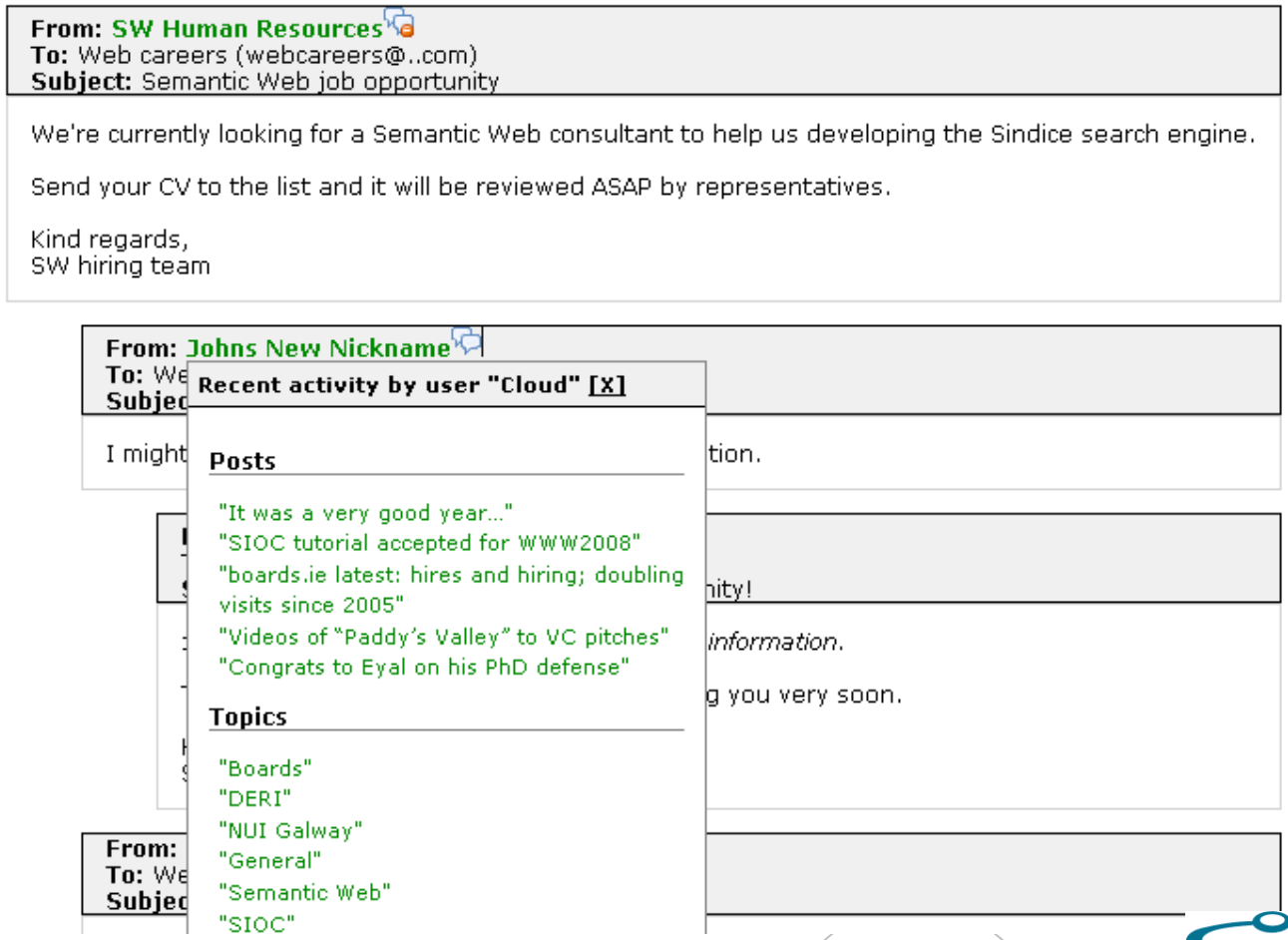
science foundation ireland
fondúireacht eolaíochta éireann

Enabling **networked** knowledge.

Higher Level APIs

- Sioc API (posts, comments, users, topics)
- In the pipeline:
  - Software development (world wide issue and bug tracking)
  - Social Graph API (++, thanks to reasoning)
  - Cross facets: Events in Locations with Friends

Example app: Sindice SIOC Widgets



Live demo available

All an HTTP call away.. but

- Data is easy to access but very difficult to integrate
  - data might be fragmented or incomplete
    - Multiple sources need to be joined
  - Identifiers (URIs) and semantic terms are not used consistently
  - Data needs “patching”
  - In general many other data transformation and normalization procedures
  - Conditional lookup and importing of extra data based on previous data

Software Pipelines + Web

- = Web Pipes (Yahoo 2007) !
  - Pipes have a URL and are computed live when requested
  - Pipes themselves are data sources and can be used within other pipes
  - Deals mostly with unstructured data, but uses RSS as a sort of “backbone” array.
  - Unfortunately cannot be extended easily
- So we hacked Semantic Web Pipes specifically to experiment:
  - Data level mashups and operators
  - Performance issues and optimizations
  - Simple UI models to turn a Semantic Pipe into an “application”

A use case: “TBL on the Semantic Web”

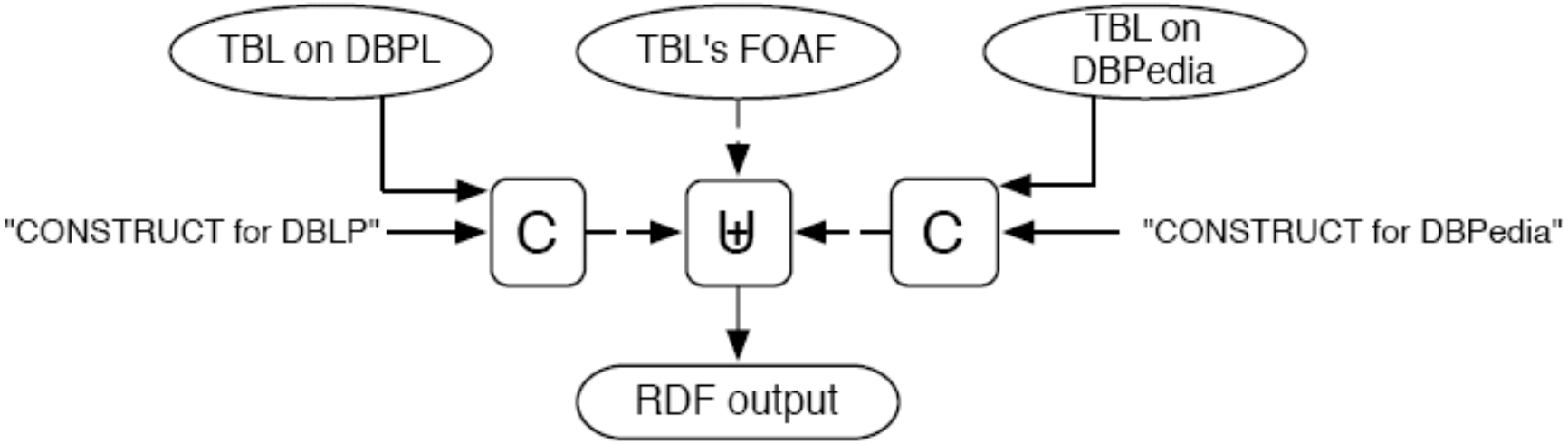
- Goal: Live aggragate of diversa data:
  - Personal semantic statements, e.g. friends
  - Public facts, e.g. awards
  - Peer reviewed publications
- Lets look at available datasources
- We found sources, but also integration issues

Transforming RDF with SPARQL

- The SPARQL query language “CONSTRUCT” functionalities comes to rescue, e.g. :

```
CONSTRUCT
  {<http://www.w3.org/People/Berners-Lee/card#i> ?p ?o.
     ?s2 ?p2 <http://www.w3.org/People/Berners-Lee/card#i>}

 WHERE
 {
  {<http://dblp.l3s.de/d2r/resource/authors/Tim_Berners-Lee> ?p ?o}
  UNION
  {?s2 ?p2 <http://dblp.l3s.de/d2r/resource/authors/Tim_Berners-Lee>}
}
```

An integration pipe for TBL on the SW



In Semantic Web XML language

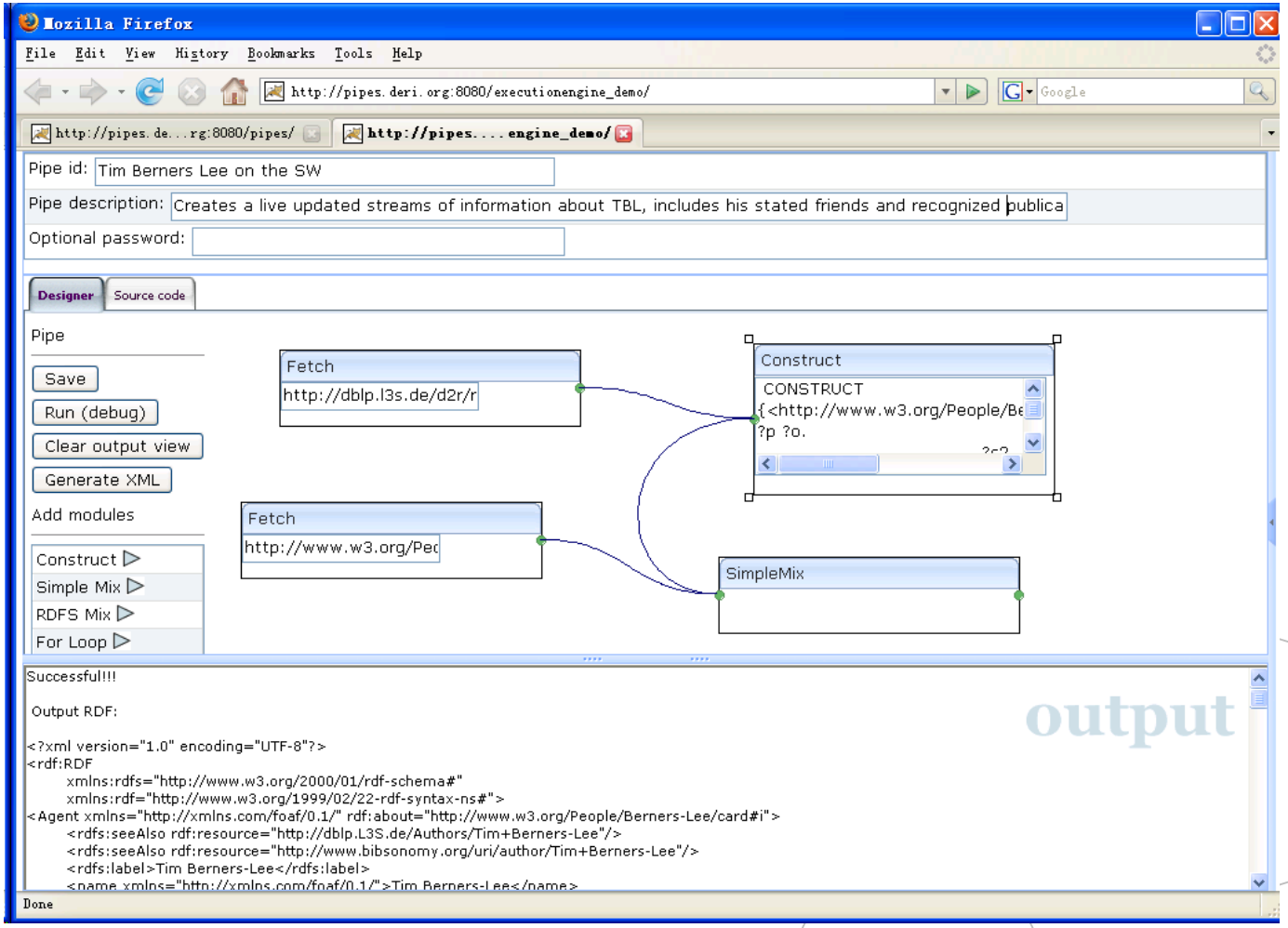

```
<construct>
    <source>
      <fetch>
        <location>http://dblp.l3s.de/d2r/resource/authors/Tim_Berners-
    Lee></location>
      </fetch>
    </source>

<query> <![CDATA[ CONSTRUCT {<http://www.w3.org/People/Berners-
    Lee/card#i> ?p ?o. ?s2 ?p2 <http://www.w3.org/People/Berners-Lee/card#i>}
    where {{<http://dblp.l3s.de/d2r/resource/authors/Tim_Berners-Lee> ?p ?o}
    UNION {?s2 ?p2 <http://dblp.l3s.de/d2r/resource/authors/Tim_Berners-Lee>}} ]]
    >
</query>

</construct>
```

- **Full specifications at http://pipes.deri.org/documentation.html)**

For people or machines?

- Why not for both!
  - Content negotiation decides if the output is RDF/XML
  - Or should be wrapped in a human legible format
- Demo time

Screenshot



More Operators

- Query driven FOR loops (with subpipes)
  - Very important for “linked data” semantic web scnarios where more information needs to be imported
- Perform reasoning (RDFS)
  - Augment, smush data
- Produce XML streams from (SPARQL construct)

..”Exotic” Operators

- Patching “Who the FOAF knows Alice”
  - A block to create and apply patces (“negates statements”)
- Custom operators:
  - A pipe itself automatically becomes an operator example:
    - <construct>
    - <source>
    - **<mixderifoaf>**
    - **<id1>12</id1>**
    - **<id2>70</id2>**
    - **</mixderifoaf>**
    - </source>
    - <query>
    - …. </query>
- So basically allows new blocks to be constructed from arbitrary REST services

Parametric Pipes

- A Pipe can get a parameter from the URL and use it within its source code
- Implements RESTful APIs in using SW pipes!
- E.g. http://pipes.deri.org:8080/pipes/Pipes/?id=mixderifoaf
- Parametric pipes can be used as custom pipe operators allowing parameters ☺

Parma Mashups

- A CAS number is a widely accepted ID for chemical compounds.
- We mash from 2 sources:
  - PDSP Ki database. A part of the HCLS demo. Contains information about binding of pharmacological substances to receptors (e.g., receptors in the brain)
  - DBPedia. Contains additional information about these substances.
- The combination of both databases can give a researcher a better impression about the action of a substance.
  - Where does it bind to (PDSP Ki database),
  - how is it metabolized (DBPedia),
  - a descriptive text (DBpedia)....
- http://pipes.deri.org:8080/pipes/Pipes/?id=pharma_mashup

(thanks to Mattiash Samwald for this)

Semantic Web Sitemaps

- Announce large quantities of data on your SiteMap

Extending Sitemaps to expose data

- Sitemaps:
  - Originally by Google, immediately adopted by all (Yahoo, MSN) etc
  - Expose the “deep web”, by providing a list of pages “to be crawled”
  - Written in XML, Linked directly in the robot.txt

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
< urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
 < url>
    < loc>http://www.example.com/</loc>
   < lastmod>2005-01-01</lastmod>
   < changefreq>monthly</changefreq>
   < priority>0.8</priority>
 </url>
</urlset>
```

The Semantic Sitemap Extention

Example first:

```
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
        http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
        xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">

  <sc:dataset>
    <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
    <sc:dataDumpLocation>http://example.org/cataloguedump.rdf
    </sc:dataDumpLocation>
    <sc:linkedDataPrefix>http://example.org/products/</sc:linkedDataPrefix>
    <changefreq>monthly</changefreq>
  </sc:dataset>

</urlset>
```

The Semantic Sitemap Extention

Example first:

```
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
        http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
        xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">

  <sc:dataset>
    <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
    <sc:dataDumpLocation>http://example.org/cataloguedump.rdf
    </sc:dataDumpLocation>
    <sc:linkedDataPrefix>http://example.org/products/</sc:linkedDataPrefix>
    <changefreq>monthly</changefreq>
  </sc:dataset>

</urlset>
```

The Semantic Sitemap Extention

Example first:

```
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
        http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
        xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">

  <sc:dataset>
    <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
    <sc:dataDumpLocation>http://example.org/cataloguedump.rdf
    </sc:dataDumpLocation>
    <sc:linkedDataPrefix>http://example.org/products/</sc:linkedDataPrefix>
    <changefreq>monthly</changefreq>
  </sc:dataset>

</urlset>
```

The Semantic Sitemap Extention

Example first:

```
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
        http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
        xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">

  <sc:dataset>
    <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
    <sc:dataDumpLocation>http://example.org/cataloguedump.rdf
    </sc:dataDumpLocation>
    <sc:linkedDataPrefix>http://example.org/products/</sc:linkedDataPrefix>
    <changefreq>monthly</changefreq>
  </sc:dataset>

</urlset>
```

The Semantic Sitemap Extention

Example first:

```
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
        http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
        xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">

  <sc:dataset>
    <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
    <sc:dataDumpLocation>http://example.org/cataloguedump.rdf
    </sc:dataDumpLocation>
    <sc:linkedDataPrefix>http://example.org/products/</sc:linkedDataPrefix>
    <changefreq>monthly</changefreq>
  </sc:dataset>

</urlset>
```

The Semantic Sitemap Extention

Example first:

```
<urlset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
        http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd"
        xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">

  <sc:dataset>
    <sc:datasetLabel>Product Catalog for Example.org</sc:datasetLabel>
    <sc:dataDumpLocation>http://example.org/cataloguedump.rdf
    </sc:dataDumpLocation>
    <sc:linkedDataPrefix>http://example.org/products/</sc:linkedDataPrefix>
    <changefreq>monthly</changefreq>
  </sc:dataset>

</urlset>
```

How it is meant to be used

As a crawler:

- If you are given a URL for an RDF site check for the sitemap
- If a dump is available, download that instead

As a client:

- If you have a dump, and want an update
- Check the sitemap, to locate it in case it has changed position
- Or to locate a SPARQL endpoint

Conclusion

- Data Web is an exciting reality
- Great opportunity to intelligently interconnect applications
  - Data level meshup can work inside the enterprise or outside
  - Publishing of RDF allows interconnecting applications (blogs, bugtracking etc)
- Find more help on how to integrate
  - http://forum.sindice.com
  - http://sindice.com/developers/welcome
- Positions are available

Acknowledgments

Data Web Services

Domain Driven Design and Mashup Oriented Development based on Open Source Java Metaframework for Pragmatic, Reliable and Secure Web Development

http://www.ict-romulus.eu

Enabling a Web of Entities

http://fp7.okkam.org

Includes adapted slides from Nate Davey & Ira Thompson