Making the Web Searchable

Peter Mika

Researcher, Data Architect

Yahoo! Research

Yahoo! Research (research.yahoo.com)

Yahoo! Research Barcelona

- Established January, 2006
- Led by Ricardo Baeza-Yates
- Research areas
  - Web Mining
    - content, structure, usage
  - Distributed Web retrieval
  - Multimedia retrieval
  - NLP and Semantics

Yahoo! by numbers (April, 2007)

- There are approximately **500 million users** of Yahoo! branded services, meaning we reach 50 percent – or **1 out of every 2 users** – online, the largest audience on the Internet (Yahoo! Internal Data).
- Yahoo! is the most visited site online with nearly **4 billion visits** and **an average of 30 visits per user per month in the U.S.** and leads all competitors in audience reach, frequency and engagement (comScore Media Metrix, US, Feb. 2007).
- Yahoo! accounts for the largest share of time Americans spend on the Internet with 12 percent (comScore Media Metrix, US, Feb. 2007) and **approximately 8 percent of the world's online time** (comScore WorldMetrix, Feb. 2007).
- **Yahoo! is the #1 home page** with 85 million average daily visitors on Yahoo! homepages around the world, an increase of nearly 5 million visitors in a month (comScore WorldMetrix, Feb. 2007).
- Yahoo!'s social media properties (Flickr, delicious, Answers, 360, Video, MyBlogLog, Jumpcut and Bix) have **115 million unique visitors worldwide** (comScore WorldMetrix, Feb. 2007).
- Yahoo! Answers is the largest collection of human knowledge on the Web with more than 90 million unique users and **250 million answers** worldwide (Yahoo! Internal Data).
- There are more than **450 million photos** in Flickr in total and **1 million photos** are uploaded daily. 80 percent of the photos are public (Yahoo! Internal Data).
- **Yahoo! Mail is the #1 Web mail provider in the world** with 243 million users (comScore WorldMetrix, Feb. 2007) and nearly 80 million users in the U.S. (comScore Media Metrix, US, Feb. 2007)
- Interoperability between Yahoo! Messenger and Windows Live Messenger has formed the largest IM community approaching 350 million user accounts (Yahoo! Internal Data).
- **Yahoo! Messenger is the most popular in time spent** with an average of 50 minutes per user, per day (comScore WorldMetrix, Feb. 2007).
- Nearly 1 in 10 Internet users is a member of a **Yahoo! Groups** (Yahoo! Internal Data).
- Yahoo! is one of only 26 companies to be on both the Fortune 500 list and the Fortune's "Best Place to Work" List (2006).

Agenda

- Part 1: today
  - Publishing content on the Semantic Web
    - Intro to the Semantic Web
    - Basic technologies
    - Linked Data
    - Exercise
    - Metadata in HTML
    - RDFa
    - Exercise
- Part 2: tomorrow
  - Semantic Web development
  - Research in Semantic Search

Remarks

- This is going to be a LOOOONG presentation
  - Ask questions, make comments, point out mistakes
  - We will have some time for exercises
- Even such a long presentation can not cover everything
  - An overview of the most important features of languages, tools, research
  - After today, let me know of additional topics you would like to hear about
  - Look for additional help and resources
    - Mailing lists
      - semantic-web@w3c.org, public-rdfa@w3c.org, public-lod@w3c.org
    - PlanetRDF blog aggregator
    - http://www.w3.org/2001/sw/
    - http://semanticweb.org
    - http://linkeddata.org
    - http://vocamp.org
    - Papers from ESWC and ISWC conferences
    - Slideshare, Twitter, YouTube etc.

Motivation

- Why publish data on the Semantic Web?
  - Multiply the value of your data by increasing content agility
    - The potential for reuse and aggregation with other datasets
    - Make your data more easily findable
- Why develop applications using semantic technologies?
  - Content agility means you can more rapidly develop applications by reusing and recombining data. Content agility leads to increased agility and robustness of your application.

Intro to the Semantic Web

Semantics demystified: the interpretation function

$$I(\text{"puppy"}) =$$

Assertions put constraints on the world, i.e. the possible interpretations

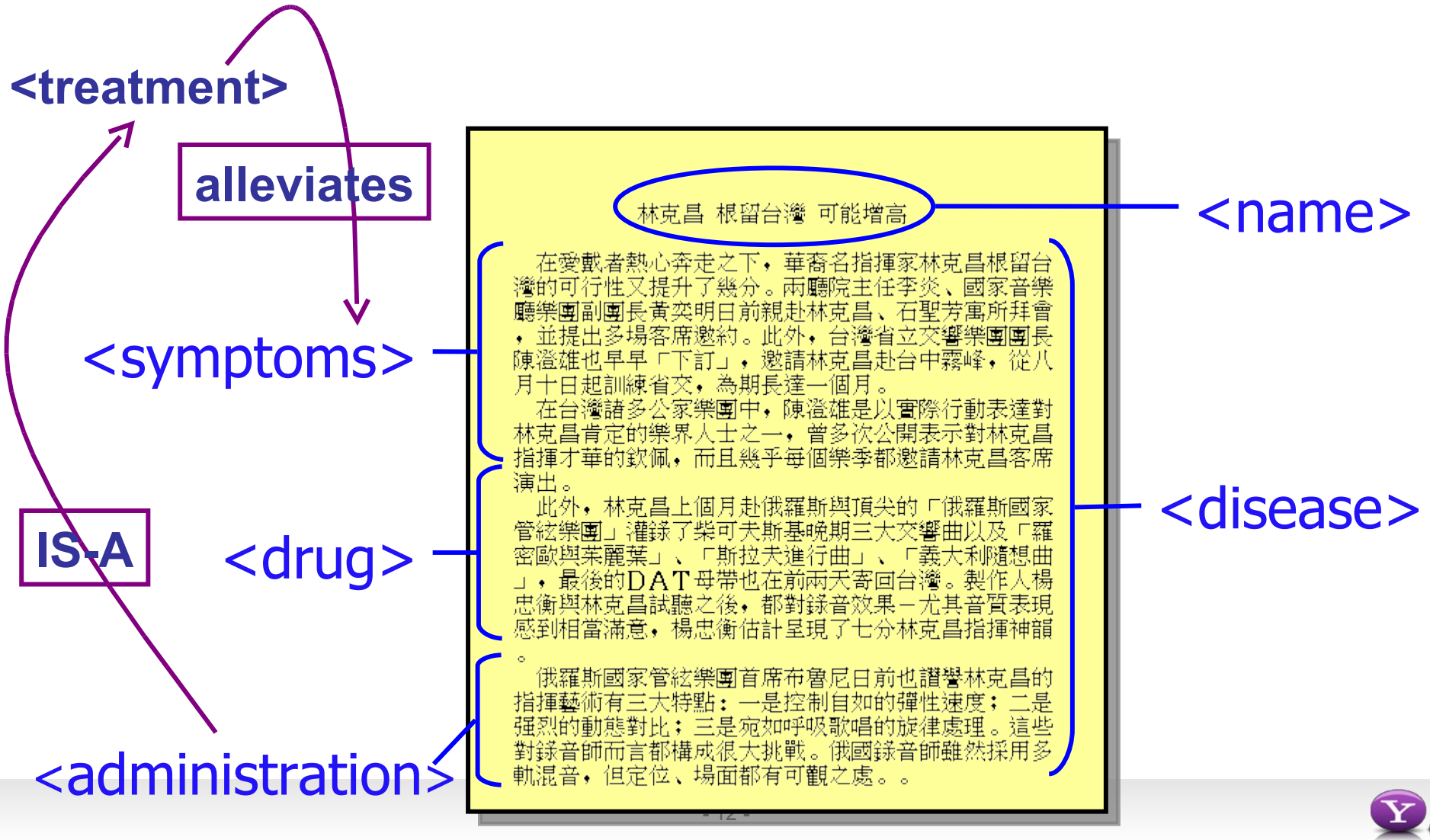
I("the puppy is in the snow")

Inference: statements that are necessarily true

Possible Worlds

“puppies have four legs”

What it's like to be a computer?



What it’s like to be a computer?

<αθζΧαψωκλ>

ηεθκδΰήχφυ

<εηκβçυψωΰτ>

ΓΔ-Λ

<ζβφυ>

<αχυπκμηχδβξμψ>

林克昌 根留台灣 可能增高

<εηιç>

在愛戴者熱心奔走之下，華裔名指揮家林克昌根留台灣的可行性又提升了幾分。兩廳院主任李炎、國家音樂廳樂團副團長黃奕明日前親赴林克昌、石聖芳寓所拜會，並提出多場客席邀約。此外，台灣省立交響樂團團長陳澄雄也早早「下訂」，邀請林克昌赴台中霧峰，從八月十日起訓練省交，為期長達一個月。

在台灣諸多公家樂團中，陳澄雄是以實際行動表達對林克昌肯定的樂界人士之一，曾多次公開表示對林克昌指揮才華的欽佩，而且幾乎每個樂季都邀請林克昌客席演出。

此外，林克昌上個月赴俄羅斯與頂尖的「俄羅斯國家管絃樂團」灌錄了柴可夫斯基晚期三大交響曲以及「羅密歐與茱麗葉」、「斯拉夫進行曲」、「義大利隨想曲」，最後的DAT母帶也在前兩天寄回台灣。製作人楊忠衡與林克昌試聽之後，都對錄音效果－尤其音質表現感到相當滿意，楊忠衡估計呈現了七分林克昌指揮神韻。

俄羅斯國家管絃樂團首席布魯尼日前也讚譽林克昌的指揮藝術有三大特點：一是控制自如的彈性速度；二是強烈的動態對比；三是宛如呼吸歌唱的旋律處理。這些對錄音師而言都構成很大挑戰。俄國錄音師雖然採用多軌混音，但定位、場面都有可觀之處。。

<βηκρξοπ>

Machines and semantics

- Computers manipulate symbols, people interpret the results
- Two pronged approach to capture semantics
  - Using formal semantics to encode the meaning of symbols
    - Based on formal descriptions, machines can manipulate symbols as expected from a human ('intelligence')
  - Social agreements
    - Formal semantics alone is typically not strong enough to capture meaning in full
- Layered approach to formal semantics
  - Semantic Web languages define basic constructs
    - Example: if *a* **subClassOf** *b*, and *x* is **type** *a* then *x* is **type** *b*
  - Domain-specific knowledge is encoded using ontologies (vocabularies) described in those languages
    - Example: Person subClassOf Mammal

Semantic Web

- The **Semantic Web** is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.
- It is the idea of having data on the Web defined and linked in a way that it can be used for more effective discovery, automation, integration, and reuse across various applications.
- The Web can reach its full potential if it becomes a place where data can be shared and processed by automated tools as well as by people. (w3c.org)

The development of the Semantic Web

- 1996-2004
  - Not much Web in the Semantic Web
    - Community originates from Knowledge Representation and Reasoning in the context of knowledge bases
    - Heavy focus on expressive languages
    - Lots of work on reasoning, mapping between ontologies
    - Broad claims on intelligence, but little achieved in practice
- 2004-today
  - Taking the reality of the Web into account
    - Focus on lightweight languages and querying
    - Scalable storage and querying of metadata with minimal reasoning
    - Addressing large communities of developers

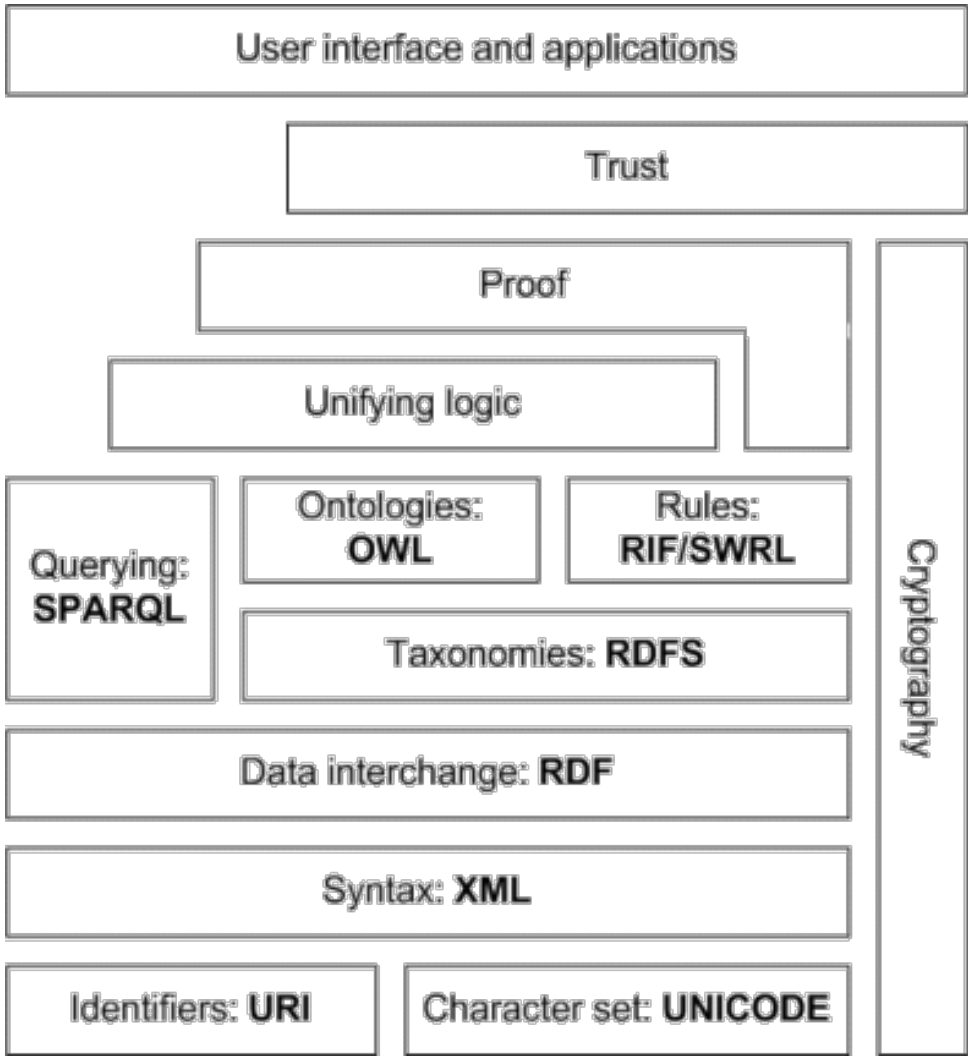
State of the Semantic Web

- Success finally arrives in two large incarnations
  - Linked Data
    - Public datasets, typically exposed and hosted independently of original publisher
    - Different methods of publishing
      - Converting to RDF/XML directly
      - Wrappers around APIs
      - RDBMS-to-RDF proxies
  - Metadata inside HTML
    - Publisher has to modify the human-readable HTML
    - Different syntaxes with various levels of expressivity
      - Microformats
      - RDFa
      - Microdata (HTML5)
- Not exclusive, but in practice publishers choose only one

Basic technology

The famous Semantic Web 'layer cake'

- Useful, but misleading in many ways
  - Don't have to implement or use all of the stack in all SemWeb applications
  - XML is just one of the many syntaxes for RDF; RDF technologies are generally incompatible with the XML world
  - Upper layers undefined, and may never be defined



RDF

- The Resource Description Framework (RDF) provides the basic data model for the Semantic Web
- RDF has two basic types of entities: resources and literals
  - Roughly objects and built-in types in Object Oriented Programming
  - Resources are identified by a URI or otherwise called a blank node
    - URIs are a generalization of URLs
    - Notation: **<http://www.example.org/Person>** or **ex:Person**

Uniform Resource Identifiers (URIs)

- "A URI is a compact sequence of characters that identifies an abstract or physical resource." RFC 3986
  - A resource is whatever a URI identifies… (!)
  - Generalizations of URLs to include names for 'things' that are not on the Web
  - Includes URIs and URNs
- IRIs are a generalization of URIs
  - Mapping from IRIs to URIs allows IRIs to be used wherever URI is required

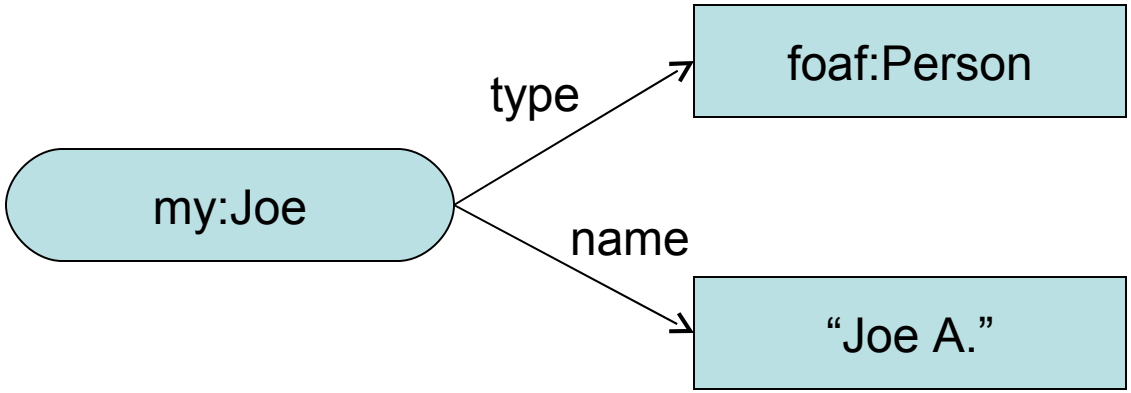
Uniform Resource Identifiers (URIs)

- How to choose URIs?
  - Choose URIs in the namespace you control, e.g. a domain name that you own
  - Choose different URIs for informational resources and conceptual resources
    - Informational resource: an HTML document, image, any other file on the Web
      - Retrievable in its entirety from the Web
    - Conceptual (non-informational) resource: a person, an event, a place, etc.
      - A description of it may be retrievable from the Web

RDF models

- A triple aka a statement is a tuple of (subject, predicate, object)
  - Example: (Joe, loves, Mary)
  - Each triple gives the value of a property for a given resource or relates two objects to one another
  - The subject is always a URI or a blank node
  - A predicate is always a resource with a URI
  - An object is either a URI, a blank node or a literal
    - Literals are strings with an optional language and datatype (string, integer etc.)
      - Datatypes are identified by URIs, e.g. XML Schema datatypes
      - Two literals are the same if their components are the same
      - Notation: **"Joe B."** or **Joe@en^^http://…#string**
  - A triple is also called a statement
- An RDF model is a set of triples
  - Ordering of statements in an RDF document is irrelevant (unlike XML)

Graphical and textual notations



- A number of text-based interchange formats for RDF
  - RDF/XML, Turtle, N3, N-Triples
  - Example: http://www.cs.vu.nl/~pmika/foaf.rdf
  - Try the RDF Validator to validate RDF documents
  - Use online services such as Triplr to quickly convert

Ontologies

- Ontologies are collections of classes and properties used to describe objects in a particular domain
  - Ontologies themselves are described in RDF or OWL (the Web Ontology Language), an extension of RDF
  - Example: the Friend-Of-A-Friend (FOAF) ontology for personal profiles
- Classes can be described by sub- and superclasses, required properties
  - Class membership in RDF is expressed using the rdf:type property
  - An instance can have multiple classes (types)
  - A class can have multiple superclasses
- Properties can be described by their domain, range, cardinalities, etc.

Advanced topic: Resources vs Literals

- Resources are objects, Literals are strings
- Resources are instances of classes, Literals have datatypes
- Whether something is a resource or literal sometimes depends on the detail of modeling

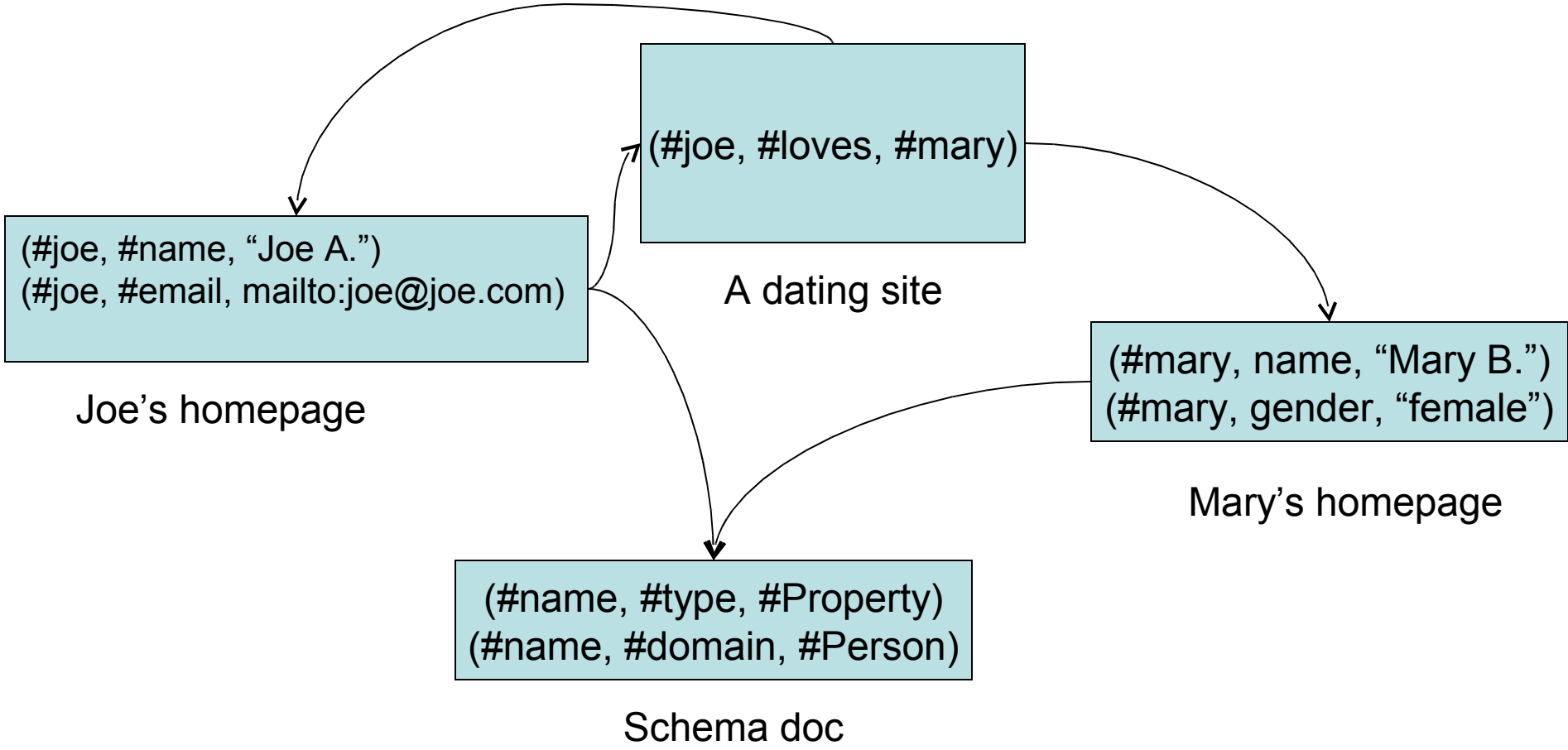
```
<meta property="myvocab:knows">Paris Hilton</meta>
<item rel="foaf:knows">
    <meta property="foaf:name">Paris Hilton</meta>
</item>
```

- You cannot make statements about literals (literals are always the object in a triple)
- Resources can carry a globally unique identifier, literals have no identity
- Web resources such as documents and images are resources
  - <item rel="rdfs:seeAlso" resource="http://www.some.related.page.com/"/>
  - <item rel="foaf:img" resource="http://photosite.example.org/photo.jpg"/>
- When in doubt: it's a resource

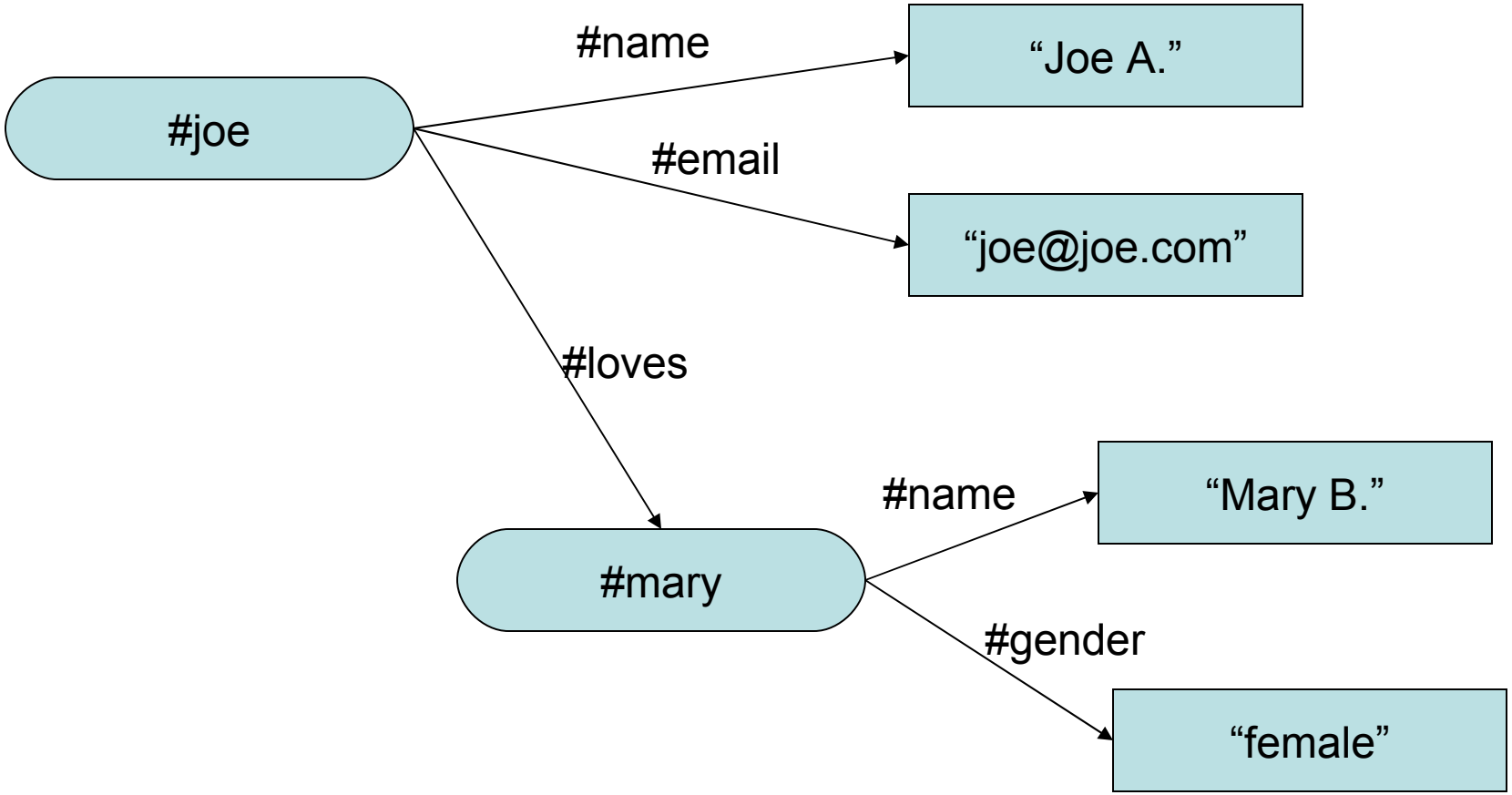
RDF is designed for distributed systems

- URIs provide web-wide global identification across documents
  - A resource may be described by multiple documents
  - We know it's the same resource because the same URI is used or through reasoning (advanced topic…)
  - URIs are intented to be reused
  - Unique, but not single identifiers: two URIs may denote the same thing
- URIs are dereferencable (can be retrieved)
  - A well-behaved URI returns a description of the resource
  - Provides authority: the definition of foaf:Person lives at that URI
- Ontologies can be looked up as well
  - Typically at the root of the URIs, also known as the namespace
  - Example: http://xmlns.com/foaf/0.1/Person redirects to the specification

URIs implicitly link data together



When put together, they form a single ‘global’ graph



Linked Data

Motivational video ;)

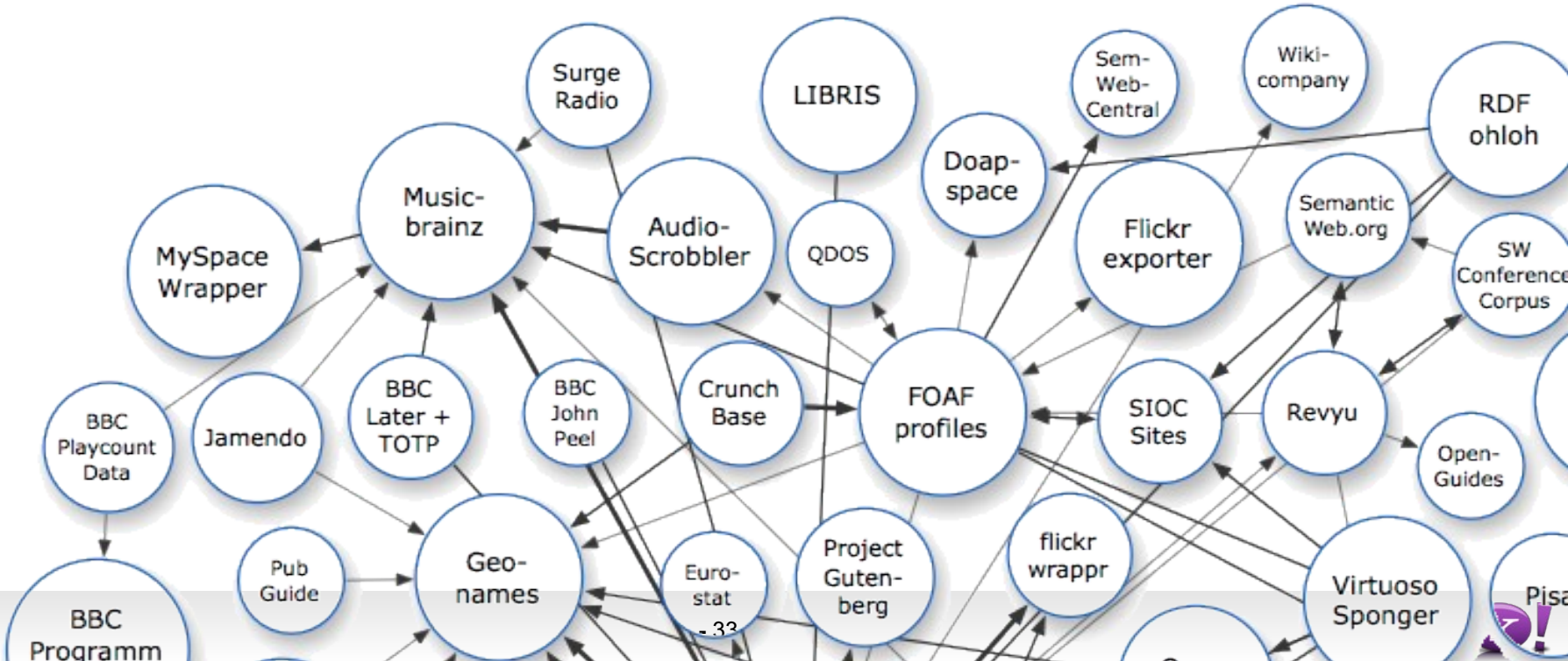
- http://www.youtube.com/watch?v=OM6XIICm_qo

What is Linked Data?

- "Linked Data is the Semantic Web done right." (Tim Berners-Lee)
  - A bottom-up, stripped down version of the Semantic Web
- Opening up the data silos in a way that maximizes interoperability and serendipitous re-use of data
  - A shared syntax for data on the Web (RDF)
  - A uniform way to discover data through links
  - Query individual data sets (SPARQL)
- What is missing?
  - Not much concern for ontologies, reasoning

Linked Data is booming…

- The “Linked Data cloud” illustrates connectivity among Linked Data datasets at the macro-level
- Primarily public datasets, often published by an independent party
  - Wikipedia infobox data published as Dbpedia by U Leipzig and FU Berlin
  - data.gov datasets published on the Data-gov wiki by RPI



How does Linked Data work?

- Summed up in four points:
  1. Use URIs as names for things
  2. Use HTTP URIs so that people can look up those names
  3. When someone looks up a URI, provide useful information, using the standards (i.e. RDF)
  4. Include links to other URIs so that they can discover more things
- Note: these are the same principles of the HTML web, applied to data

Serving content for both humans and machines

GET http://dbpedia.org/resource/Madrid

Accept: text/html

Accept: application/rdf+xml

http://dbpedia.org/page/Madrid

About: Madrid

An Entity in Data Space: dbpedia.org

Madrid is the capital and largest city of Spain. It is the third-most
and Berlin, and its metropolitan area is the third-most populous c

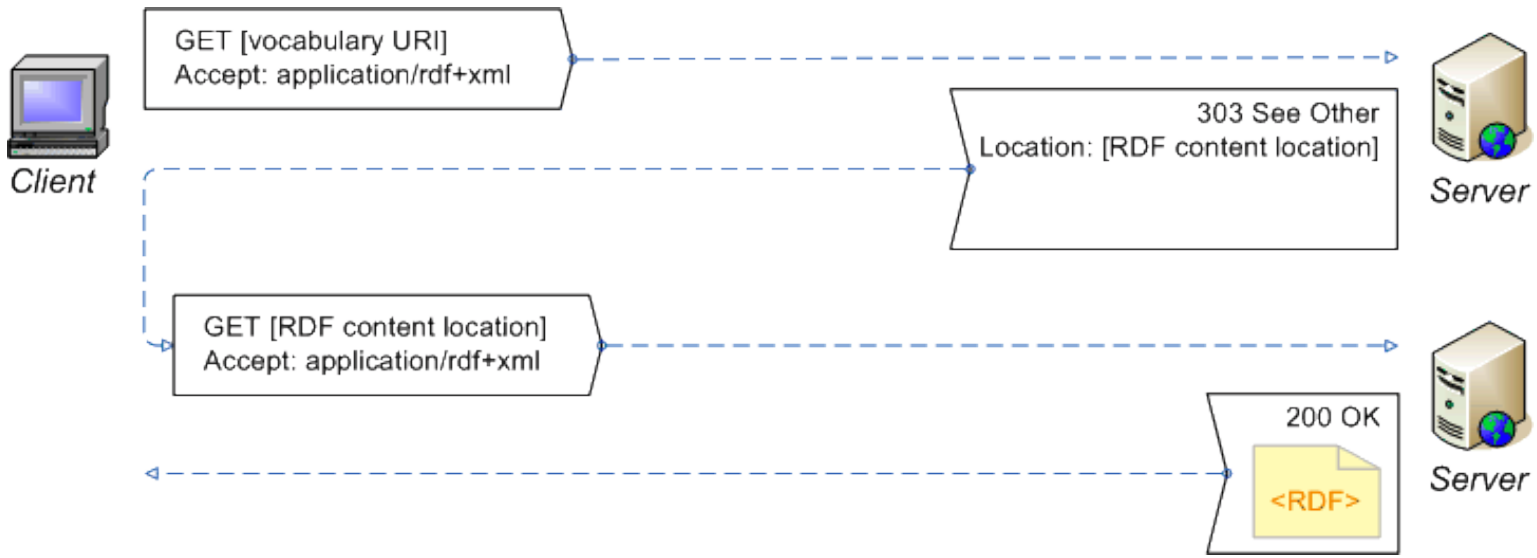
Property	Value
dbpedia-owl:Place/areaLand	▪ 607
dbpedia-owl:Place/elevation	▪ 667
dbpedia-owl:PopulatedPlace/areaCode	▪ 34 (Spain) + 9
dbpedia-owl:PopulatedPlace/areaMetro	▪ 10506 ▪ 10508
dbpedia-owl:PopulatedPlace/establishedTitle	▪ Founded
dbpedia-owl:PopulatedPlace/leaderName	▪ dbpedia:Alber
dbpedia-owl:PopulatedPlace/leaderTitle	▪ Mayor

http://dbpedia.org/data/Madrid

```
– <rdf:RDF>
  – <rdf:Description rdf:about="http://dbpedia.org/resource
      <dbpprop:stadium rdf:resource="http://dbpedia.org/
    </rdf:Description>
  – <rdf:Description rdf:about="http://dbpedia.org/resource
      <dbpprop:stadium rdf:resource="http://dbpedia.org/
    </rdf:Description>
  – <rdf:Description rdf:about="http://dbpedia.org/resource
      <dbpprop:location rdf:resource="http://dbpedia.org/
    </rdf:Description>
  – <rdf:Description rdf:about="http://dbpedia.org/resource
      <rdfs:label xml:lang="de">Madrid</rdfs:label>
    </rdf:Description>
  – <rdf:Description rdf:about="http://dbpedia.org/resource
      <dbpprop:deathPlace rdf:resource="http://dbpedia.o
    </rdf:Description>
  – <rdf:Description rdf:about="http://dbpedia.org/resource
      <dbpprop:deathPlace rdf:resource="http://dbpedia.o
    </rdf:Description>
```

Slight complication: redirection

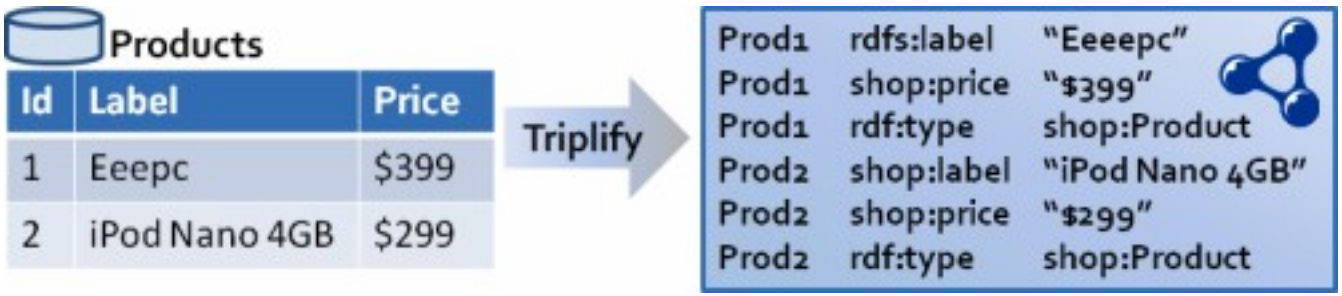
- The server should not return immediately with a 200 (success) error code
- A 303 (see other) response is required in order to formally distinguish conceptual resources from normal web resources (information resources)



- For more details, see Bizer et al.
  How to Publish Linked Data

General approach of RDBMs to Linked Data mapping

- Mapping relational tables



  - The first column must contain identifiers which can be used to generate instance URIs (the primary key of your database table).
  - Column names will be used to generate property URIs; by renaming the columns of your database table (e.g. SELECT id,name AS 'foaf:name' FROM users)
  - The individual cells of the query result contain data values or references to other instances (foreign keys) and will eventually constitute the objects of resulting triples.

Tools to publish Linked Data

- Linked Data wrappers around relational databases
  - Triplify
    - PHP API
    - Preconfigured mappings for a number of popular CMSs and Web application frameworks
  - D2RQ
    - Declarative mapping language between the relational schema and an ontology
    - A plug-in for Sesame and Jena for rewriting SPARQL queries into SQL queries and executing them on the database
    - An HTTP Server providing the Linked Data functionality plus a SPARQL endpoint
- Linking tools to create links to external datasets
  - SILK
    - Declarative language to specify mappings, i.e. when two instances should be linked
    - Interactive tool to create mappings and validate them
    - Protocol for keeping mappings up-to-date
- Some triple stores (e.g. Virtuoso) have built-in Linked Data interfaces

Linked Data browsers and validators

- Browsers
  - Tabulator
    - Online browser and Firefox extension
  - Zitgist
    - Online browser
  - Marbles
    - Online browser
  - Disco
    - Online browser
  - Openlink Data Explorer
    - Online browser and Firefox extension
- Validators
  - Vapour Linked Data validator
- Other tools

Querying data sets using SPARQL

- SPARQL is both a query language for RDF and a protocol for executing queries on remote servers
  - A server implementing SPARQL access is called a SPARQL endpoint
  - Currently no data manipulation (read-only)
  - Querying using SPARQL is limited to individual servers
- Example: people who were born in Berlin before 1900

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?name ?birth ?death ?person WHERE {
    ?person dbpedia2:birthPlace <http://dbpedia.org/resource/Berlin> .
    ?person dbo:birthDate ?birth .
    ?person foaf:name ?name .
    ?person dbo:deathDate ?death
    FILTER (?birth < "1900-01-01"^^xsd:date) .
}
ORDER BY ?name
```

Advanced topics in Linked Data

- Federated querying of Linked Data datasets
  - e.g. DARQ
- Linked Data licensing
  - e.g. CC0, Talis's Open Data Commons
- Description languages for datasets and SPARQL endpoints
  - VoiD
- Hosted Linked Data services
  - e.g. Talis's Connected Commons

Exercises

- RDF basics
  - Take a look at the RDF/XML Primer
  - Create a simple RDF description of yourself using the FOAF vocabulary
  - Try the RDF validator
  - Store the document on a Web server
  - Convert it to the different alternative syntaxes of RDF using Triplr
- Try SPARQL
  - DBpedia SPARQL interface
- Try some Linked Data browsers
  - e.g. take the example Semantic Web ride with Disco

Metadata in HTML

Brief history of the Annotated Web

- 1995: HTML meta tags
- 1996: Simple HTML Ontology Extensions (SHOE)
- 1998: RDF/XML
  - RDF/XML in HTML
  - RDF linked from HTML
- 2003: Web 2.0
  - Tagging
  - Microformats
  - Metadata in Wikipedia
  - Machine tags in Flickr
- 2005: eRDF
- 2008: RDFa

HTML meta tags

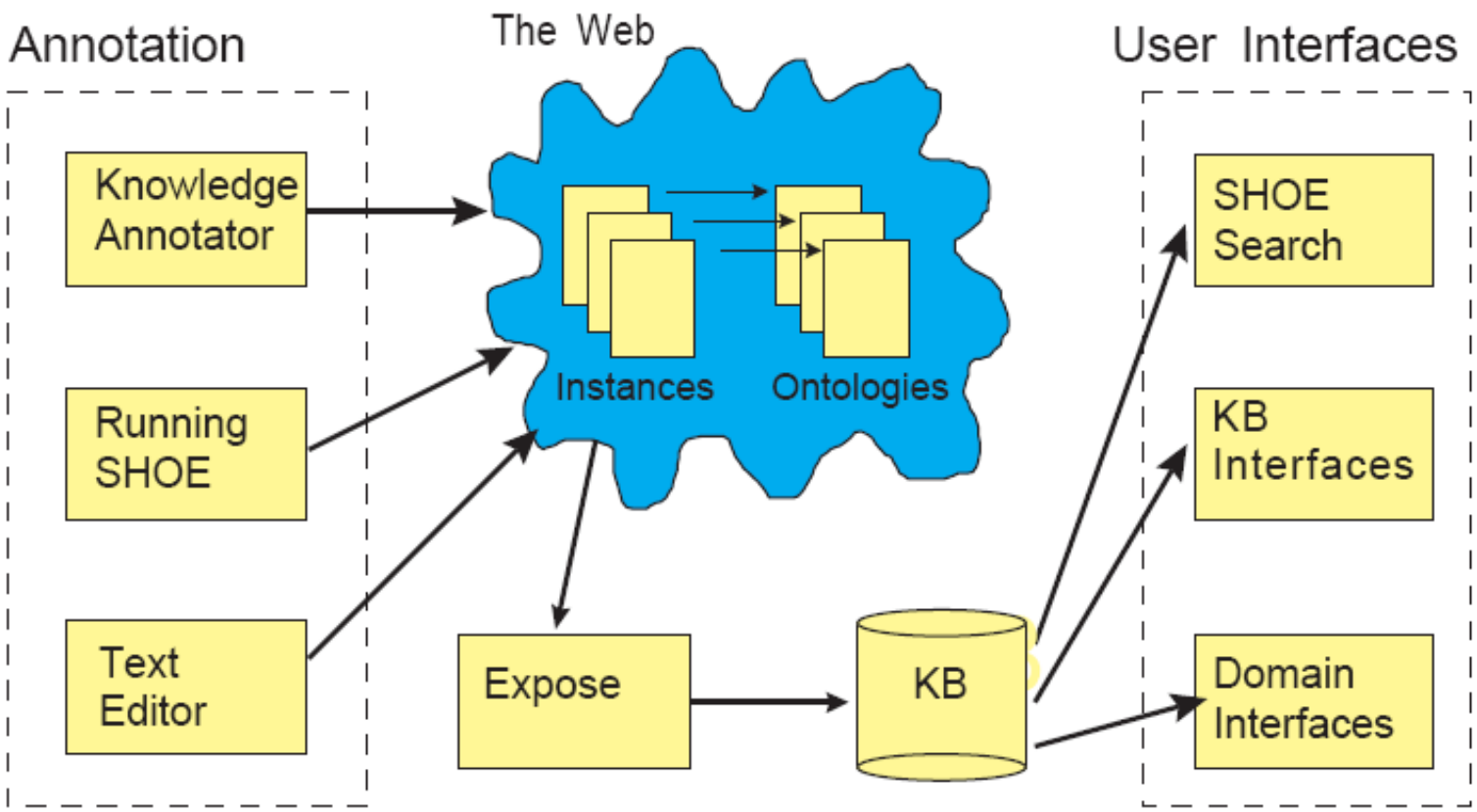
```
<HTML>
<HEAD profile="http://dublincore.org/documents/dcq-html/">
<META name="DC.author" content="Peter Mika">
<LINK rel="DC.rights copyright"
  href="http://www.example.org/rights.html" />
<LINK rel="meta" type="application/rdf+xml" title="FOAF"
     href= "http://www.cs.vu.nl/~pmika/foaf.rdf">
</HEAD>
…
</HTML>
```

SHOE example
(Hefflin & Hendler, 1996)

```
<ONTOLOGY "our-ontology" VERSION="1.0">
<ONTOLOGY-EXTENDS "organization-ontology" VERSION="2.1" PREFIX="org"
    URL="http://www.ont.org/orgont.html">
<ONTDEF CATEGORY="Person" ISA="org.Thing">
<ONTDEF RELATION="lastName" ARGS="Person STRING">
<ONTDEF RELATION="firstName" ARGS="Person STRING">
<ONTDEF RELATION="marriedTo" ARGS="Person Person">
<ONTDEF RELATION="employee" ARGS="org.Organization Person">
</ONTOLOGY>
```

```
<HEAD>
<META HTTP-EQUIV="Instance-Key" CONTENT="http://www.cs.umd.edu/~george">
<USE-ONTOLOGY "our-ontology" VERSION="1.0" PREFIX="our" URL="http://ont.org/our-ont.html">
</HEAD>
<BODY>
<CATEGORY "our.Person">
<RELATION "our.marriedTo" TO="http://www.cs.umd.edu/~helena">
<RELATION "our.employee" FROM="http://www.cs.umd.edu">
My name is
<ATTRIBUTE "our.firstName"> George </ATTRIBUTE>
<ATTRIBUTE "our.lastName"> Cook </ATTRIBUTE> and I live at...
```

SHOE system


Annotation
The Web
User Interfaces
Knowledge Annotator
Running SHOE
Text Editor
Instances
Ontologies
Expose
KB
SHOE Search
KB Interfaces
Domain Interfaces

SHOE Text-based query interface

Untitled

File Help

Ontology: cs-dept-ontology, v.1.0

Query

Select a category:

AssociateProfessor
Chair
Dean
FullProfessor
VisitingProfessor
Publication
Article
ConferencePaper
JournalArticle
TechnicalReport

Select

Subject Category: Article

publicationResearch	Simple HTML Ontology Extension:
publicationAuthor	Heflin
publicationDate	
name	
softwareDocumentation OF	
orgPublication OF	

First Item: 1

Last Item: 3

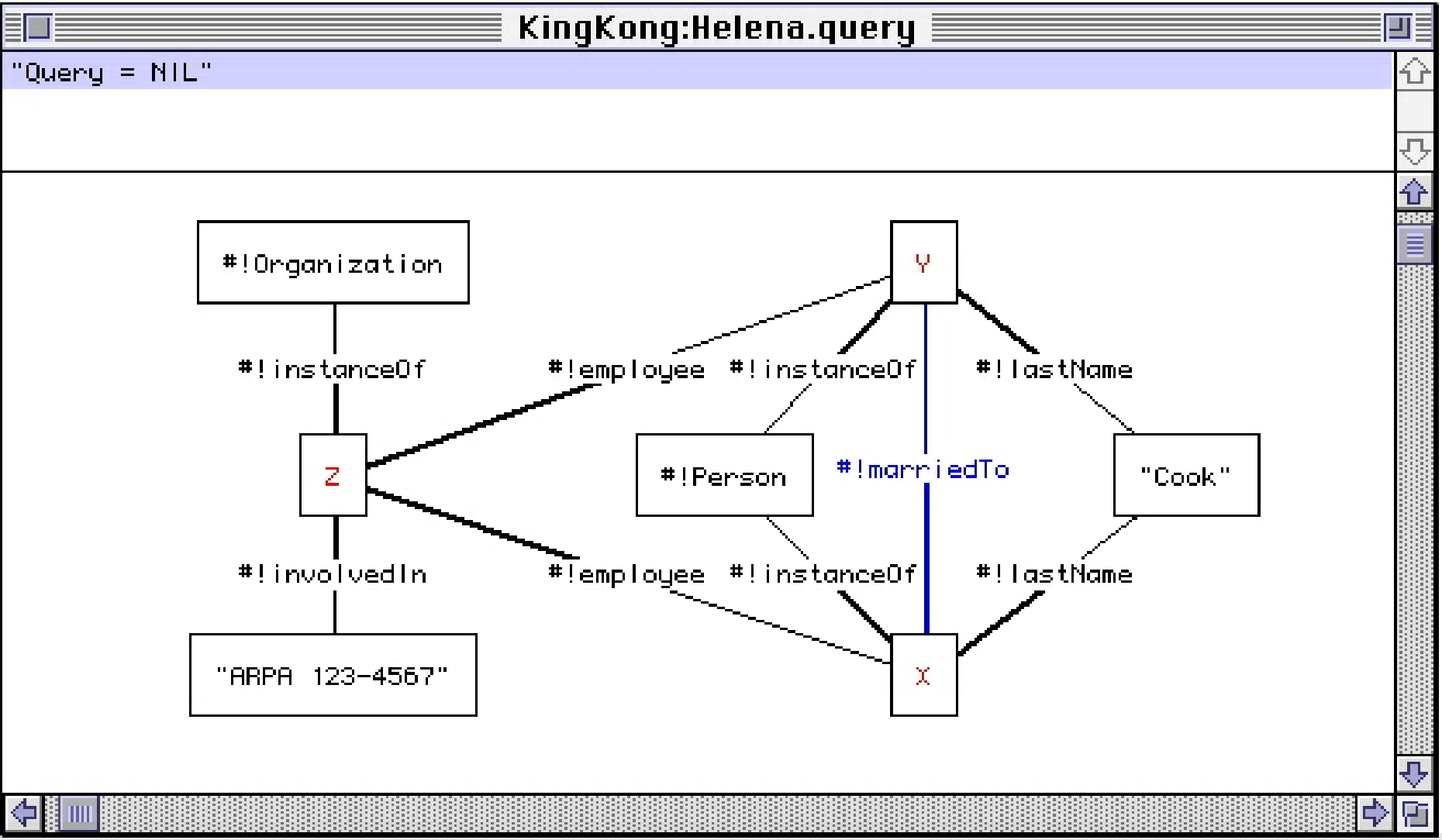
Get Last Group

Get Next Group

Article		publicationResearch(name)		publicationAuthor(name)	name
http://ww	htt	Simple HTML Ontology Extensio	htt	Jeff Heflin	Coping with Changing Ontologies in a
http://ww	htt	Simple HTML Ontology Extensio	htt	Jeff Heflin	Applying Ontology to the Web: A Case
http://ww	htt	Simple HTML Ontology Extensio	htt	Jeff Heflin	Reading Between the Lines: Using SH

Web Search Get Page Add To Query 3 answers.

SHOE Graphical Query Interface



Example: Creative Commons

Embedding CC license in HTML (now deprecated):

```
<HTML>
<HEAD>… </HEAD>
<BODY>
…
<!--
 <rdf:RDF xmlns="http://creativecommons.org/ns#"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
 <Work rdf:about="http://www.yergler.net/averages/">
 <dc:title>The Law of Averages</dc:title>
 <dc:description>...because eventually i&apos;ll be right...</dc:description>
 <license rdf:resource="http://creativecommons.org/licenses/by-nc/1.0/" />
 </Work>
 <License rdf:about="http://creativecommons.org/licenses/by-nc/1.0/">
<requires rdf:resource="http://web.resource.org/cc/Notice" />
 <permits rdf:resource="http://web.resource.org/cc/Reproduction" />
 <permits rdf:resource="http://web.resource.org/cc/Distribution" />
 <prohibits rdf:resource="http://web.resource.org/cc/CommercialUse" />
 </License>
 </rdf:RDF>
-->
```

Example: Creative Commons

- Current: rel attribute (HTML4)

```
This work is licensed under a <a rel="license"
href="http://creativecommons.org/licenses/by/3
.0/us/">Creative Commons Attribution 3.0
United States License</a>.
```

- Use of the “rel” attribute for semantic annotation is the birth of the microformat…

Microformats (µf)

- Community centered around microformats.org
  - Specifications and discussions are hosted there
- Agreements on the way to encode certain kinds metadata in HTML
  - Reuse of semantic-bearing HTML elements
  - Based on existing standards
  - Minimality
- Microformats exist for a limited set of objects
  - hCard (persons and organizations)
  - hCalendar (events)
  - hResume
  - hProduct
  - hRecipe
- Varying degrees of support and stability
  - hCard and rel-tag are widely supported

Microformats: limitations

- No shared syntax
  - Each microformat has a separate syntax tailored to the vocabulary
- No formal schemas
  - Limited reuse, extensibility of schemas
  - Unclear which combinations are allowed
- No datatypes
- No namespaces, unique identifiers (URIs)
  - no interlinking
  - mapping between instances is required
- Relationship to page context is often unclear

Example: microformats

```
<div class="vcard">
  <a class="email fn" href="mailto:jfriday@host.com">Joe Friday</a>
  <div class="tel">+1-919-555-7878</div>
  <div class="title">Area Administrator, Assistant</div>
</div>
```

```
<cite class="vcard">
<a class="fn url" rel="friend colleague met"
href="http://meyerweb.com/">Eric Meyer</a>
</cite> wrote a post (<cite>
<a href="http://meyerweb.com/eric/thoughts/2005/12/16/tax-relief/">
Tax Relief</a></cite>) about an unintentionally humorous letter
he received from the <span class="vcard">
<a class="fn org url" href="http://irs.gov/">
Internal Revenue Service</a> </span>.
```

Microformats vs. RDFa

- Choose microformats when you find a microformat that fits your needs and supported by Yahoo!
  - Microformats are first option because they are simple
  - We support all major microformats, see the documentation
  - It's a common misconception that RDFa requires XHTML: it doesn't
- If you find none that *perfectly* fits your needs then you need RDFa
  - Microformats have a fixed schema: you can not add your own attributes
- Example: a social networking site with user profiles
  - VCard is a good candidate, but for example it doesn't have a way to express the user's social connections
  - You either live without this, or go with RDFa
- The rest of this presentation is about RDFa, which is thus more powerful, but also more complex
  - We will focus on the concepts that are hard to grasp

Keep an eye on HTML5

- Currently under standardization at the W3C
  - Last Call this fall, keep an eye on it
- Introduces Microdata
  - Similar to microformats
    - Some predefined vocabularies with central registration
  - Some of the flexibility of RDFa
  - Introduce new terms using reverse domain names or full URIs
- Semantic HTML elements such as <time>, <video>, <article>…

Microdata example

```
<div item="http://www.yahoo.com/resource/person">

 <p>My name is <span itemprop="name">Neil</span>.</p>

 <p>My band is called

 <span itemprop="band">Four Parts Water</span>.

 I was born on

 <time itemprop="birthday" datetime="2009-05-10">May 10th 2009</time>.

 <img itemprop="image" src="me.png" alt="me">

 </p>

</div
```

Introduction to RDFa

Slides courtesy of Mark Birbeck

What does RDFa look like?

- There are some metadata features in HTML already...
- ...so we give them an RDF interpretation...
- ...then we generalise them...
- ...and then we add a few more.

HTML's metadata features (1)

```
<html>
    <head>
      <title>RDFa: Now everyone can have an API</title>
      <meta name="author" content="Mark Birbeck" />
      <meta name="created" content="2009-05-09" />
      <link rel="license"
   href="http://creativecommons.org/licenses/by-sa/3.0/" />
    </head>
    .
    .
    .
   </html>
```

HTML's metadata features (2)

```
<a href="http://creativecommons.org/licenses/by-sa/3.0/"
    >CC Attribution-ShareAlike</a>
```

```
<a rel="license"
 href="http://creativecommons.org/licenses/by-sa/3.0/"
    >CC Attribution-ShareAlike</a>
```

RDFa extends @rel/@href to images

```
<img src="image01.png"
 rel="license"
 href="http://creativecommons.org/licenses/by-sa/3.0/"
/>

<img src="image02.png"
 rel="license"
 href="http://creativecommons.org/licenses/by-sa/3.0/"
/>
```

RDFa extends meta/@content to body

```
<html>
    <head>
      <title>RDFa: Now everyone can have an API</title>
      <meta name="author" content="Mark Birbeck" />
      <meta name="created" content="2009-05-09" />
    </head>
    <body>
      <h1>RDFa: Now everyone can have an API</h1>
      Author: <em>Mark Birbeck</em>
      Created: <em>May 9th, 2009</em>
    </body>

</html>
```

RDFa extends meta/@content to body

```
<html>
    <head>
      <title>RDFa: Now everyone can have an API</title>
    </head>
    <body>
      <h1>RDFa: Now everyone can have an API</h1>
      Author: <em property="author" content="Mark Birbeck"
       >Mark Birbeck</em>
      Created: <em property="created" content="2009-05-09"
       >May 9th, 2009</em>
    </body>

</html>
```

RDFa extends meta/@content to body

```
<html>
    <head>
      <title>RDFa: Now everyone can have an API</title>
    </head>
    <body>
      <h1>RDFa: Now everyone can have an API</h1>
      Author: <em property="author"
       >Mark Birbeck</em>
      Created: <em property="created" content="2009-05-09"
       >May 9th, 2009</em>
    </body>

</html>
```

Vocabularies use CURIEs

```
<html xmlns:dc="http://purl.org/dc/terms/">
    <head>
      <title>RDFa: Now everyone can have an API</title>
    </head>
    <body>
      <h1>RDFa: Now everyone can have an API</h1>
      Author: <em property="dc:creator"
       >Mark Birbeck</em>
      Created: <em property="dc:created" content="2009-05-09"
       >May 9th, 2009</em>
    </body>
</html>
```

CURIEs, or Compact URIs

- Named after Marie Curie, who was the first person to receive two Nobel prizes, one for physics and one for chemistry.

- CURIEs allow a full URI to be expressed in a simple prefix:suffix form.

Properties can also apply to images

```
<img src="image01.png”
 rel="license"
 href=“http://creativecommons.org/licenses/by-sa/3.0/”
/>

<img src="image02.png”
 rel="license"
 href="http://creativecommons.org/licenses/by-sa/3.0/”
/>
```

Properties can also apply to images

```
<img src="image01.png"
 rel="license"
 href="http://creativecommons.org/licenses/by-sa/3.0/"
 property="dc:creator" content="Mark Birbeck”
/>

<img src="image02.png"
 rel="license"
 href="http://creativecommons.org/licenses/by-sa/3.0/"
    property="dc:creator" content="Mark Birbeck"
/>
```

Relationships and properties on anything

```
<a
 href="http://www.slideshare.net/mark.birbeck/the-5-minute-
    guide-to-rdfain-only-6-minutes-40-seconds"
     >The 5 minute guide to RDFa...in only 6 minutes and 40
    seconds</a>
```

Relationships and properties on anything

```
<a rel="license"
 href="http://www.slideshare.net/mark.birbeck/the-5-minute-
    guide-to-rdfain-only-6-minutes-40-seconds"
     >The 5 minute guide to RDFa...in only 6 minutes and 40
    seconds</a>
```

Doesn't say what we want.

Relationships and properties on anything

```
<a
 href="http://www.slideshare.net/mark.birbeck/the-5-minute-
   guide-to-rdfain-only-6-minutes-40-seconds"
    >The 5 minute guide to RDFa...in only 6 minutes and 40
   seconds</a>

   is licensed under

   <a



 href="http://creativecommons.org/licenses/by-sa/2.5/"

    >CC BY SA</a>.
```

Relationships and properties on anything

```
<a
 href="http://www.slideshare.net/mark.birbeck/the-5-minute-
   guide-to-rdfain-only-6-minutes-40-seconds"
    >The 5 minute guide to RDFa...in only 6 minutes and 40
   seconds</a>

   is licensed under

   <a about="http://www.slideshare.net/mark.birbeck/the-5-
   minute-guide-to-rdfain-only-6-minutes-40-seconds"
 rel="license"
 href="http://creativecommons.org/licenses/by-sa/2.5/"

    >CC BY SA</a>.
```

Relationships and properties on anything

```
<a
 href="http://www.slideshare.net/mark.birbeck/the-5-minute-
   guide-to-rdfain-only-6-minutes-40-seconds"
    >The 5 minute guide to RDFa...in only 6 minutes and 40
   seconds</a>
   is licensed under

   <a about="http://www.slideshare.net/mark.birbeck/the-5-
   minute-guide-to-rdfain-only-6-minutes-40-seconds"
 rel="license"
 href="http://creativecommons.org/licenses/by-sa/2.5/"
 property="dc:creator" content="Mark Birbeck">
   CC BY SA
</a>.
```

@about sets context

```
<div about="http://www.slideshare.net/mark.birbeck/the-5-
   minute-guide-to-rdfain-only-6-minutes-40-seconds">

  <h1>The 5 minute guide to RDFa...</h1>
     Author: <em property="dc:creator"
      >Mark Birbeck</em>
     Created: <em property="dc:created" content="2009-05-09"
      >May 9th, 2009</em>

</div>
```

@about sets context

```
<html xmlns:dc="http://purl.org/dc/terms/">
 <head>
      <title>RDFa: Now everyone can have an API</title>
 </head>
    <body>


  <h1>RDFa: Now everyone can have an API</h1>
  Author: <em property="dc:creator"
       >Mark Birbeck</em>
      Created: <em property="dc:created" content="2009-05-09"
       >May 9th, 2009</em>
    </body>
</html>
```

Basics of RDFa

- generalise HTML's existing semantic features;
- add support for CURIEs for property and relationship names;
- add @about.

Advanced RDFa

- use of @datatype to set the data type of @content;
- use of @typeof to set rdf:type;
- support for bnodes;
- support for XML literals;
- ability to chain statements together.

- Note that since RDFa supports all of the features you'll find in RDF, then it means that you can even mark-up OWL documents in HTML.

RDFa pitfalls

- Validation problems can stop us from extracting data
  - Use the W3C validator
  - Use the right DOCTYPE declaration if using XHTML
  - Set the encoding of your page properly (using HTTP headers or XML declaration)
- Prefixes need to be defined using the xmlns attribute
- Unless you are making statements about the document, set the subject using the about attribute
- Do not include HTML elements in literal values
  - Incorrect: Peter Mika
- Use absolute URIs as the value of the resource attribute
  - Or make sure you specify HTML base

More pitfalls: precedence rules

- Be careful when using rel and typeof in combination because of the precedence rules
- BAD example:

```
<div about="#id">
    <span property="foaf:name">Peter Mika</span>
    <span rel="foaf:img" typeof="foaf:Image">
      <span property="dc:format">jpg</span>
      …
    </span
</div>
```

- To correct, you need to put the typeof inside the node with rel="foaf:img"

More pitfalls: the typeof attribute

- Typeof does two things at once: it creates a new subject resource and assigns the type to it
- BAD example:

```
<div about="#id">
    <span property="foaf:name">Peter Mika</span>
    <span rel="foaf:img"
    resource="http://www.example.org/photo.jpg">
      <span typeof="foaf:Image">
        <span property="dc:format">jpg</span>
      </span
    </span
</div>
```

- To correct, you have to repeat the resource attiribute on the span node with the typeof

HTML markup pitfalls

- Marking up <h1>:

  <h1 property="dc:title">My homepage</h1>

  NOT: <h1> My homepage</h1>

- Marking up an image:

```
<span rel="foaf:img">
  <img alt="Alex" src="http://example.org/photos/alex.jpg"/>
</span>
```

  This doesn't work:

- In the header you need

  <meta property="…" content="…">

  NOT

  <meta name="…" content="…">

More pitfalls: breaking up descriptions

- You can not break up a description like this:

```
<span rel="foaf:knows">
   <span property="foaf:name">Peter Mika</span>
</span>
….
<span rel="foaf:knows">
   <a rel="foaf:email" href="mailto:pmika@yahoo-inc.com />
</span>
```

- This is not the same as:

```
<span rel="foaf:knows">
   <span property="foaf:name">Peter Mika</span>
   <a rel="foaf:email" href="mailto:pmika@yahoo-inc.com />
</span>
```

- In the first case there are two related resources, with one attribute each, in the second case there is a single related resource with two attributes.

Tips

- Hiding information from being displayed
  - Links without content will not be rendered
  - Use
- Use datatypes to provide the expected type of a literal.
  - This helps validation because any tool can check whether the literal is indeed of that type.

Choosing a vocabulary

- Look at SearchMonkey objects
  - Video, Games, Presentations, Events, News, Businesses, Products, Discussion
- Search the Web or ask for advice on mailing lists
  - semantic-web@w3.org
  - public-rdfa@w3.org
- Wikis
  - semanticweb.org
  - vocamp.org
- Beware of people who claim to have the vocabulary of everything
  - Preferably you want something small and targeted
- Never a 100% fit → you will need to introduce vocabulary terms (classes and properties)
  - Do not introduce new classes/properties in existing namespaces
  - Example: the namespace http://xmlns.com/foaf/0.1/ is used by the FOAF project. Try not to introduce a new term without contacting the owner, i.e. the membership of the FOAF mailing list:

The process of annotating with RDFa

1. Invest in familiarizing with the RDFa syntax by reading the RDFa Primer
   - It is also highly recommended that you read the RDF Primer. RDF is the data model used by RDFa.
2. Choose a vocabulary from the SearchMonkey documentation that fits your needs
   - A vocabulary describes a set of types and attributes within a given domain
   - If you don't find a good candidate, extend an existing one or create a new one
3. Annotate your page.
   - Before you start, you might want to validate your page for (X)HTML conformance using the W3C's (X)HTML Validator to reduce the chance of errors. Choose Document Type XHTML + RDFa.
   - No specific tool support. If you have an HTML or XML editor that supports DTDs, you will have syntax checking and highlighting.
   - Use the RDFa Distiller to validate which data can be extracted from your page.
   - If you fancy, use the RDF Validator to graphically visualize the RDF graph that is outputted.
4. Put the annotated page online. The data will extracted the next time your page is crawled
   - No need to explicitly submit anything
   - No notification when your site is crawled

- See http://rdfa.info/rdfa-implementations for new tools and APIs

Automated approaches

- Trade-off between precision, recall and effort
  - Low cost, broad coverage, but error-prone
  - Expensive, targeted but precise
- Variety of approaches
  - NLP
    - Named entity extraction with or without disambiguation
    - From text to triples
      - Linguistic patterns
      - Deep parsing
  - Information Extraction
    - Form filling combined with list extraction (Halevy et al.)
    - Wrapper induction
- (Public) examples:
  - NLP: Zemanta, OpenCalais,
  - Wrapper induction: Dapper, MashMaker

Example: Zemanta

- A personal writing assistant for bloggers
  - Plugin for popular blogging platforms and web mail clients
- Analyzes text as you type and suggests hyperlinks, tags, categories, images and related articles
  - Inserts tags using the Common Tag semantic tagging format
- API available with the same functionality

Your content enhanced!

Branded "unfilmable", Watchmen - the cult graphic novel about a group of retired, flawed superheroes - has finally made it to the big screen. From the second the opening credits roll, it is clear Watchmen is not your typical superhero movie.

Image by TCM Hitchhiker via Flickr

An ageing vigilante, The Comedian, is attacked in his high-rise apartment before being hurled 10 storeys to his death... in graphic slow motion. What follows is a two-and-three-quarter hour epic that centres on an outlawed group of deeply flawed former heroes as a Cold War Doomsday clock inches ever closer to midnight and nuclear apocalypse.

First published in 12 parts by DC Comics in 1986, Watchmen was

Exercise

- Explore data on the Web
  - Microformats
    - Search for pages on Yahoo using searchmonkey:com.yahoo.page.uf.hcard
    - Try Operator Firefox Plug-in
    - Try Optimus
  - RDFa
    - Create yourself or search for pages on Yahoo using searchmonkey:com.yahoo.page.rdf.rdfa
    - Try the RDFa bookmarklet to highlight RDFa
    - Try the RDFa Distiller to extract RDF from HTML
    - Try the RDF Validator to visualize your RDF data
- Mark up your webpage using RDFa
  - RDFa Distiller
- Zemanta, OpenCalais, Common Tag

Advanced topic: creating a vocabulary

1. Get advice on methodology
   - vocamp.org and semanticweb.org
2. Choose a namespace and a prefix
   - Give sensible names, e.g. name it after your site, but don't call it searchmonkey
   - Namespace ends either with a slash or a hash
3. Create an RDF or OWL document describing your classes and properties
   - Use an ontology editor such as Protégé 4.0
   - Follow naming conventions
4. Publish your vocabulary
   - Make sure the URIs of your properties and classes are resolvable
     - E.g. myvocab:digicam should resolve to a document containing the definition of myvocab:digicam
5. Convince others to adopt your vocabulary
   - If you are in fishing, convince other fishing businesses