

CPT-S 415

Big Data

Yinghui Wu

EME B45

CPT-S 415

Big Data

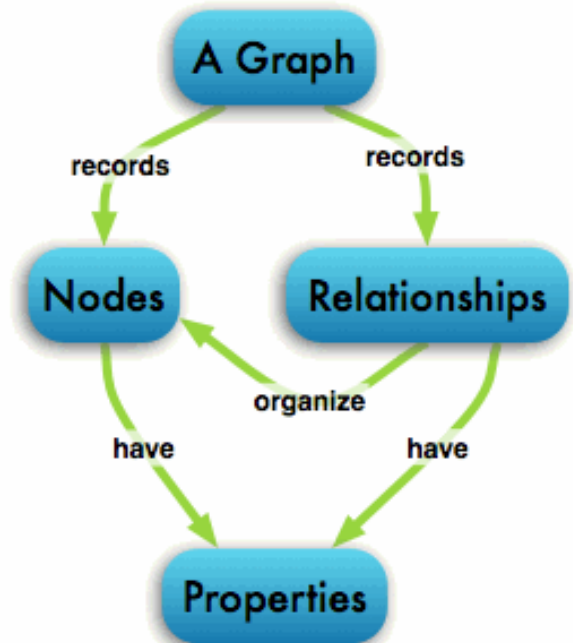
Beyond Relational Data

Graphs and RDF data

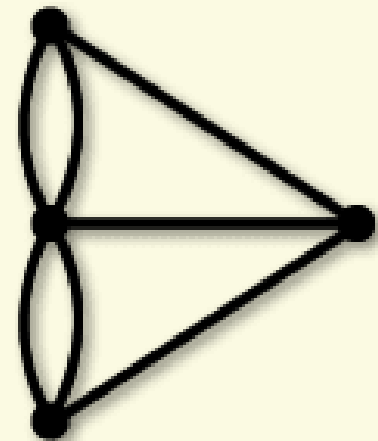
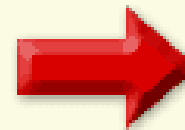
- ✓ Graph data basics
- ✓ Introduction to RDF
 - RDF data model and syntax
 - RDF schemas
 - RDF inferencing

What's a graph?

- ✓ $G = (V, E)$, where
 - V represents the set of vertices (nodes)
 - E represents the set of edges (links, relationships)
 - Both vertices and edges may contain additional information
- ✓ Different types of graphs:
 - Directed vs. undirected
 - Simple vs. multi-graphs
 - Weighted vs. unweighted
- ✓ Networks, linked data, Web, Grid...



Seven Bridges of Königsberg

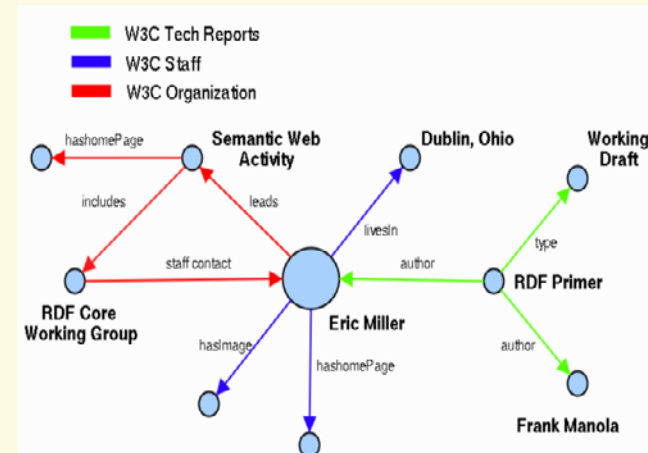
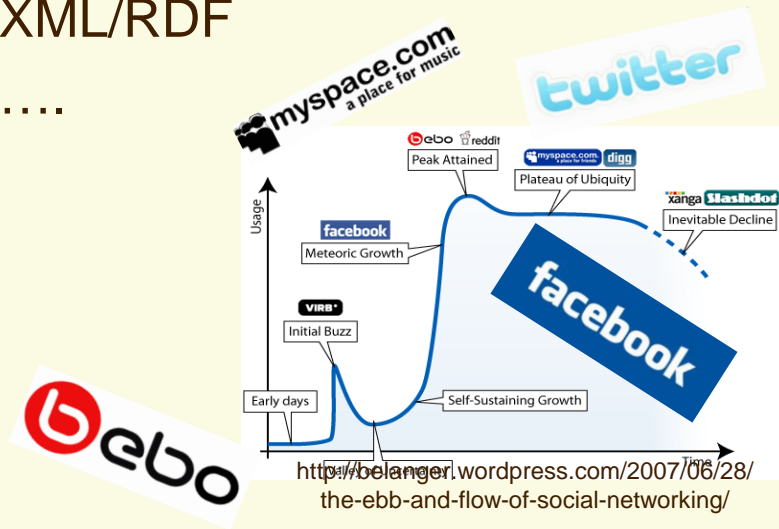


Leonhard Euler, 1736



Ubiquitous Network (Graph) Data

- Social Network
- Biological Network
- Road Network/Map
- WWW
- Semantic Web/Ontologies
- XML/RDF
-

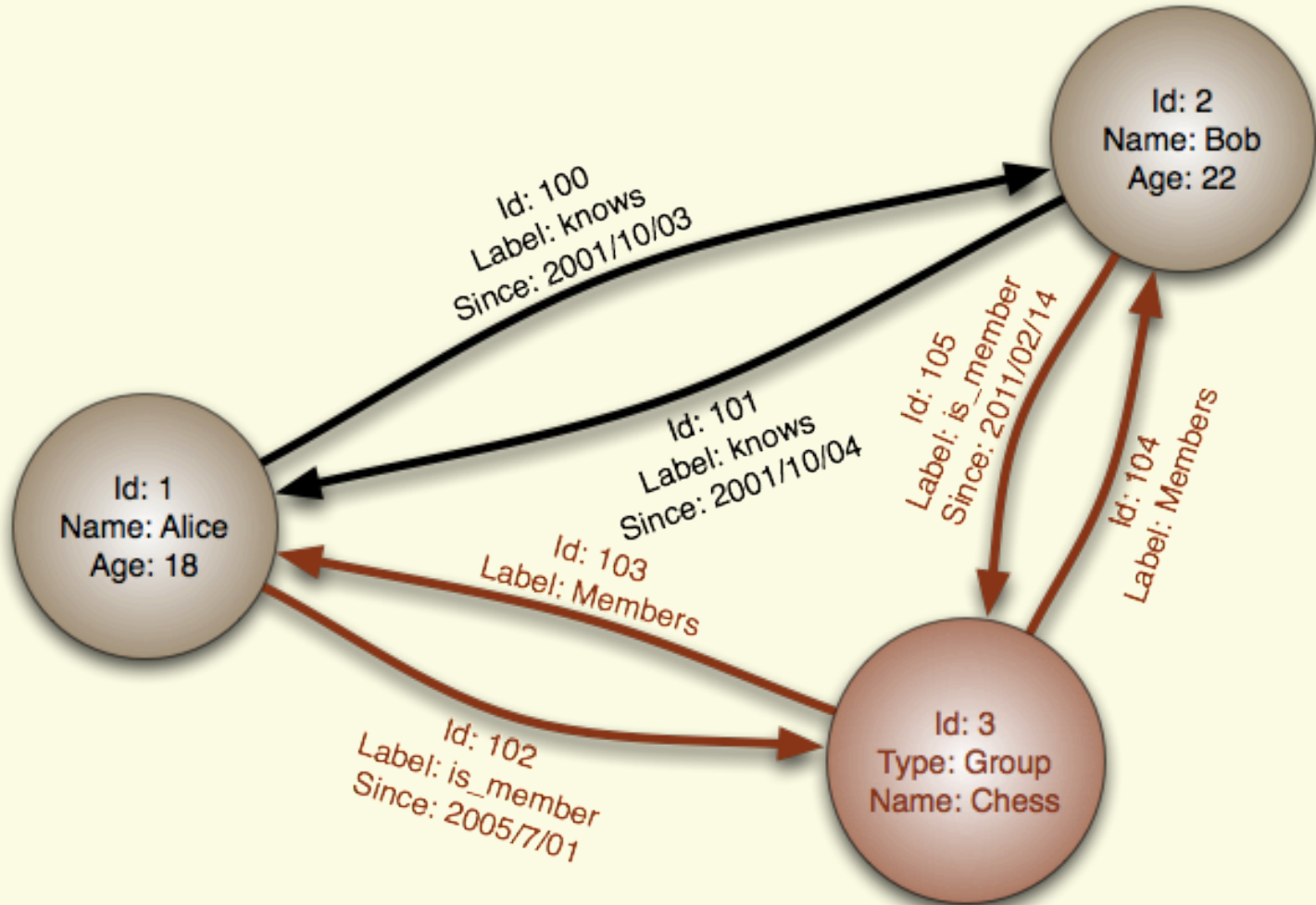


Semantic Search, Guha et. al., WWW'03



How to represent?

Property graph (Neo4j, Gremlin)



Can we use XML?

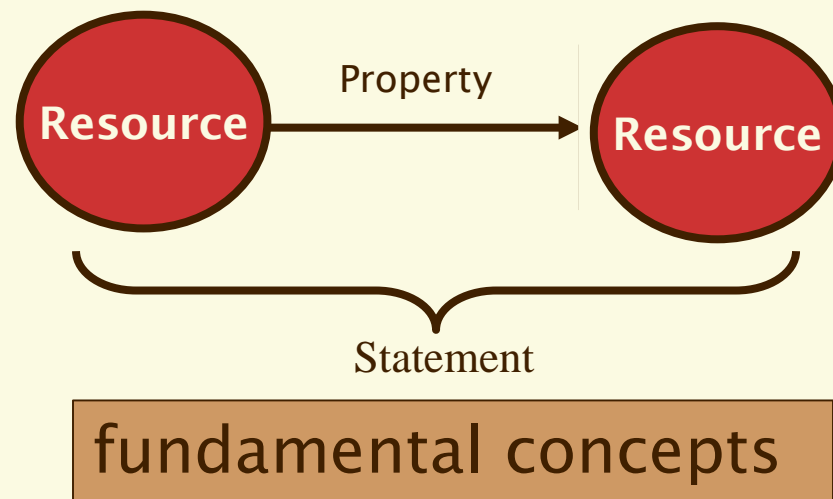
- ✓ XML is a universal metalanguage for defining markup
- ✓ It provides a uniform framework for interchange of data and metadata between applications
- ✓ However, XML does not provide any means of talking about the semantics (meaning) of data
- ✓ E.g., there is no intended meaning associated with the nesting of tags
 - It is up to each application to interpret the nesting.

What is RDF?

- ✓ Resource Description Framework: Developed by the World Wide Web Consortium (W3C) to provide a standard for defining an architecture for supporting the vast amount of web metadata.
- ✓ Human and machine readable
 - Machine-readable: it maintains the structure of the data.
- ✓ Short history:
 - Metadata: begins in 1995
 - *Platform for Internet Content Selection (PICS)*
 - Mechanism for communicating ratings of web pages from server to clients.
 - Interned resource description based on PICS architecture
 - PICS-NG working group -> RDF, 2004

Basic Ideas of RDF

- ✓ Basic building block: **object-attribute-value** triple
 - It is called a **statement**
 - Also: Subject-predicate-object
- ✓ RDF has been given a syntax in XML
 - inherits the benefits of XML
 - Other syntactic representations of RDF possible



Resources

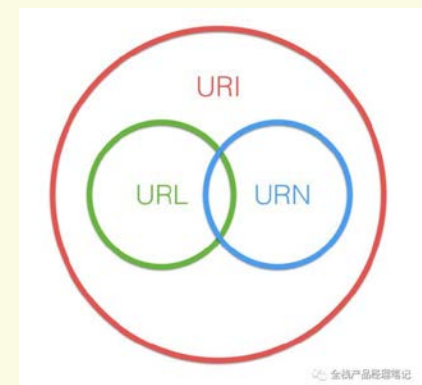
- ✓ We can think of a resource as an object
 - E.g. authors, books, publishers, places, people, hotels
- ✓ Every resource has a **URI**, a Universal Resource Identifier
- ✓ A URI can be
 - a URL (Web address) or
 - some other kind of unique identifier (e.g., URN; ISBN)

URIs are a foundation

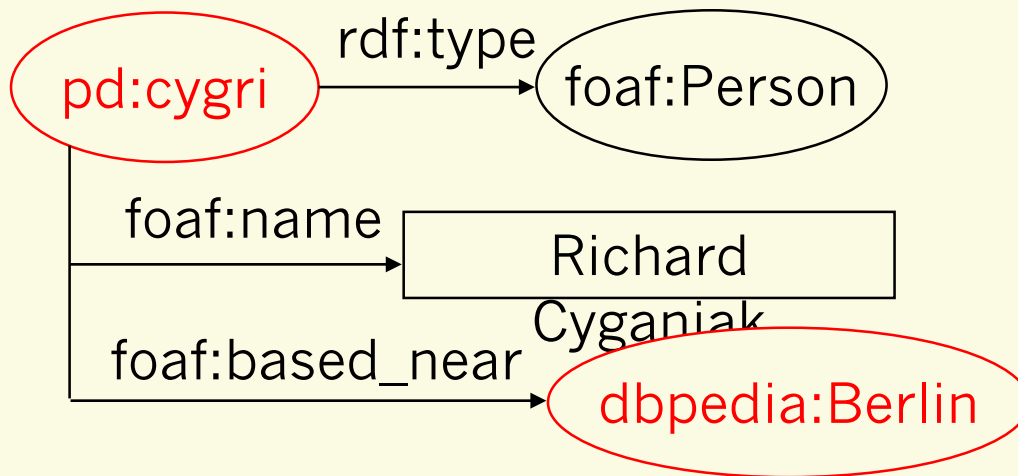
- ✓ URI = Uniform Resource Identifier
 - "The generic set of all names/addresses that are short strings that refer to resources"
 - URLs (Uniform Resource Locators) are a subset of URIs, used for resources that can be *accessed* on the web
- ✓ URIs look like URLs, often with fragment identifiers pointing to a document part:
 - `http://foo.com/bar/mumble.html#pitch`

Advantages of using URIs:

A global, worldwide, unique naming scheme
Reduces the homonym problem of distributed data representation



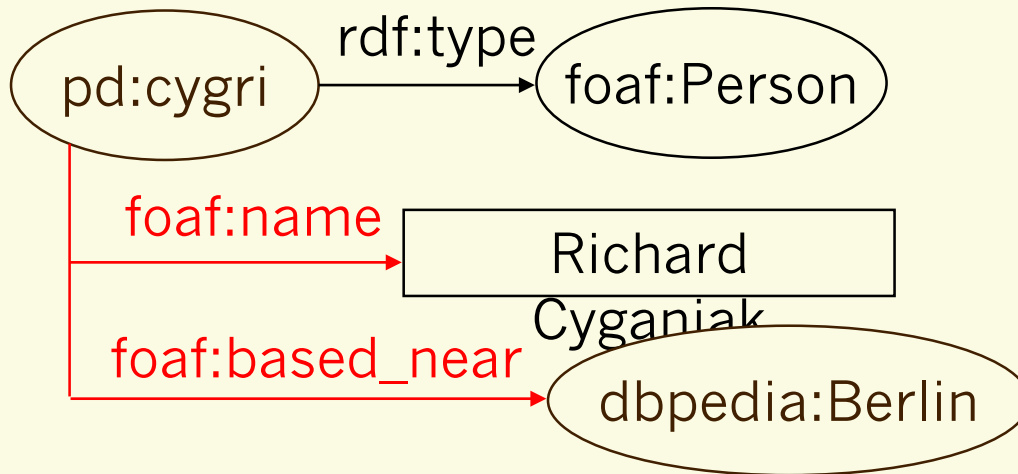
Resources identified with HTTP URIs



dbpedia:Berlin = <http://dbpedia.org/resource/Berlin>

pd:cygri = <http://richard.cyganiak.de/foaf.rdf#cygri>

Properties

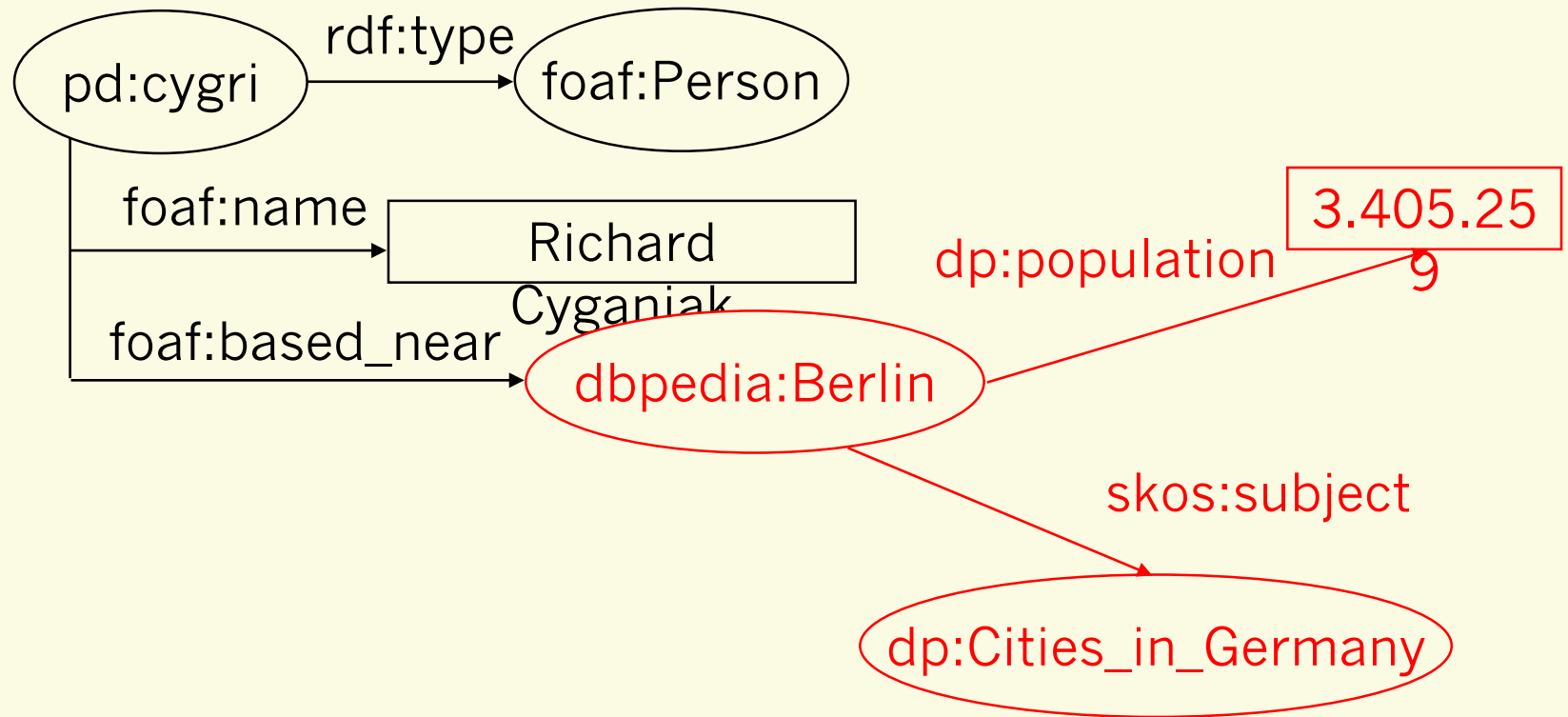


- ✓ Properties are a special kind of resources
- ✓ describe relations between resources
 - e.g. “written by”, “age”, “title”, etc.
- ✓ Properties are also identified by URIs
- ✓ Properties can be a subject, object or recursively defined

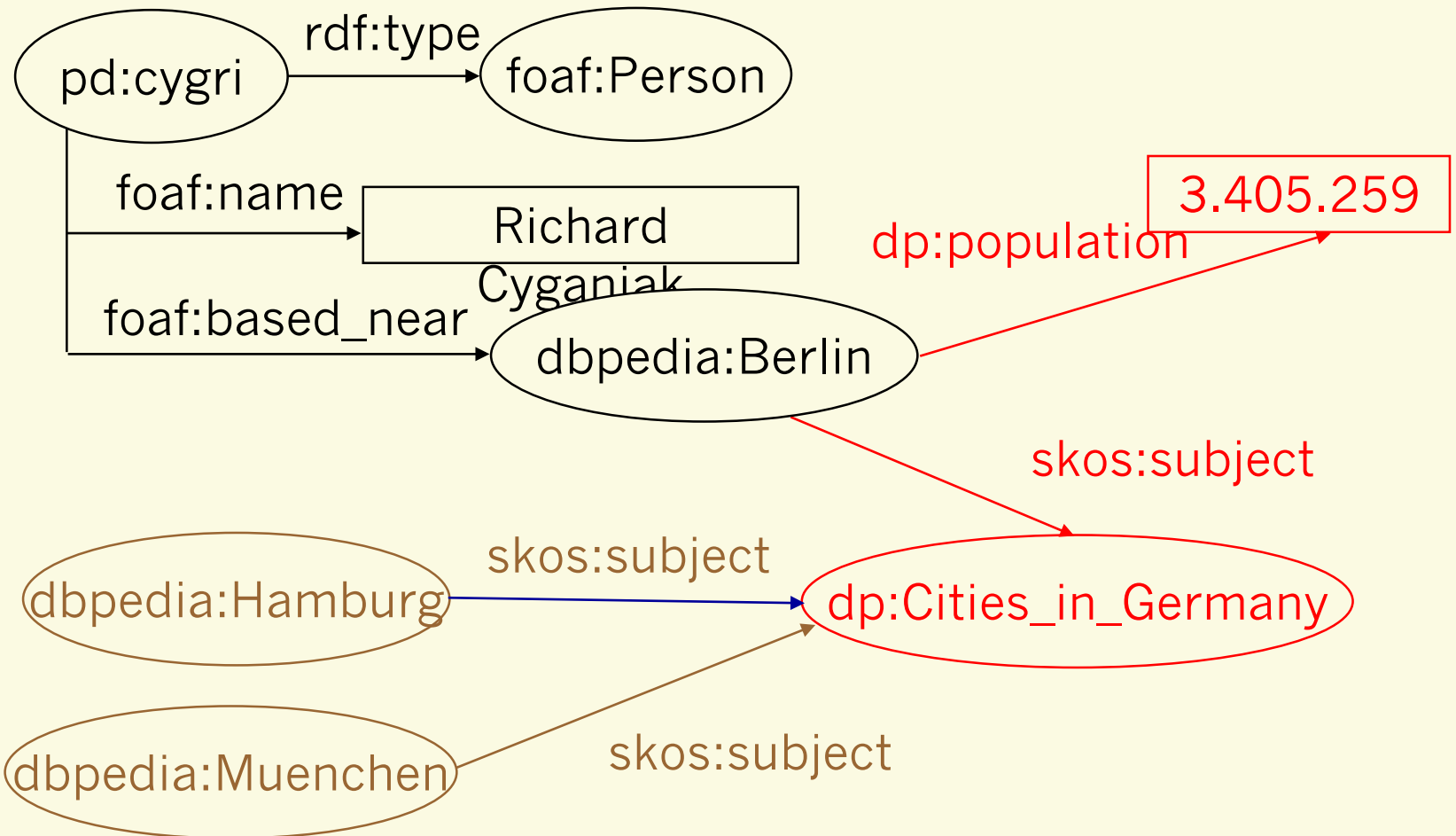
Statements

- ✓ Statements assert the properties of resources
 - ✓ A statement is an object-attribute-value triple
 - It consists of a resource, a property, and a value
 - ✓ Values can be resources or **literals**
 - Literals are atomic values (strings)
 - ✓ a Statement can be viewed as:
 - A triple
 - A piece of a graph
 - A piece of XML code
- ➡
- Hence an RDF document is
- A set of triples
 - A graph (semantic Web)
 - An XML document

Resolving URIs over the Web



Dereferencing URIs over the Web



Berlin

URI:

Property	Value	Sources
population	3398888	G2
type	http://dbpedia.org/City ↗	G2
comment	Berlin is the capital city and one of the sixteen Federal States of Germany. It is the country's largest city in area and population, and the second most populous city in the European Union.	G2
comment	Berlin ist die deutsche Bundeshauptstadt und als Stadtstaat ein eigenständiges Land der Bundesrepublik Deutschland. Berlin ist die bevölkerungsreichste und flächengrößte Stadt Deutschlands und nach Einwohnern die zweitgrößte Stadt der EU.	G2
label	Berlin	G2
sameAs	http://sws.geonames.org/2950159/ ↗	G2
subject	http://dbpedia.org/resource/category/Berlin ↗	G2
subject	http://dbpedia.org/resource/category/Capitals_in_Europe ↗	G2
subject	http://dbpedia.org/resource/category/Cities_in_Germany ↗	G2
subject	http://dbpedia.org/resource/category/German_state_capitals ↗	G2
subject	http://dbpedia.org/resource/category/Host_cities_of_the_Summer_Olympic_Games ↗	G2
subject	http://dbpedia.org/resource/category/States_of_Germany ↗	G2
sourceURL	Berlin ↗	G1
depiction		G2
page	http://en.wikipedia.org/wiki/Berlin ↗	G2
is birthplace of	Adolf von Baeyer ↗	G2
is birthplace of	http://dbpedia.org/resource/person/Albert_Speer_%28the_younger%29 ↗	G2

XML-Based Syntax of RDF

- ✓ An RDF document consists of an **rdf:RDF** element
 - The content of that element is a number of descriptions

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

xmlns:uni="http://www.mydomain.org/uni-ns">

<rdf:Description rdf:about="949318">

<uni:name>Yinghui Wu</uni:name>

<uni:title>Assistant Professor</uni:title>

<uni:office rdf:datatype="&xsd:string">EME B45</uni:office>

</rdf:Description>

<rdf:Description rdf:about="CPTS 415">

<uni:courseName>Big Data</uni:courseName>

<uni:isTaughtBy>Yinghui Wu</uni:isTaughtBy>

</rdf:Description>

</rdf:RDF>

Property Elements

- ✓ Content of **rdf:Description** elements

```
<rdf:Description rdf:about="CPTS 483-05">  
  <uni:courseName>Big Data</uni:courseName>  
  <uni:isTaughtBy>Yinghui Wu</uni:isTaughtBy>  
</rdf:Description>
```

- ✓ **uni:courseName** and **uni:isTaughtBy** define two property-value pairs for **CPTS 415**(two RDF statements)

The `rdf:resource` Attribute

We can denote that two entities are the same using the `rdf:resource` attribute

```
<rdf:Description rdf:about="CPTS 483-05">  
  <uni:courseName>Big Data  
  </uni:courseName>  
  <uni:isTaughtBy rdf:resource="949318"/>  
</rdf:Description>
```

```
<rdf:Description rdf:about="949318">  
  <uni:name>Yinghui Wu</uni:name>  
  <uni:title>Assistant Professor</uni:title>  
</rdf:Description>
```

RDF Containers

- ✓ Collect a number of resources or attributes about which we want to make statements as a whole
- ✓ Permit aggregation of several values for a property
- ✓ Different container semantics
 - Bag (rdf: Bag)
 - unordered grouping (e.g., students in this class)
 - Sequence (rdf: Seq)
 - ordered grouping (e.g., authors of a paper)
 - Alternatives (rdf: Alt)
 - alternate values (e.g., measurement in different units)

Example for a Bag and Alternative

```
<uni:lecturer rdf:ID="949352" uni:name="Yinghui Wu"
  uni:title= "Assistant Professor">
  <uni:coursesTaught>
    <rdf:Bag>
      <rdf:_1 rdf:resource="#CPTS 583-06"/>
      <rdf:_2 rdf:resource="#CPTS 415"/>
    </rdf:Bag>
  </uni:coursesTaught>
</uni:lecturer>

<uni:course rdf:ID="CPTS 415"
  uni:courseName="Big Data">
  <uni:lecturer>
    <rdf:Alt>
      <rdf:li rdf:resource="#949318"/>
      <rdf:li rdf:resource="# 949319"/>
    </rdf:Alt>
  </uni:lecturer>
</uni:course>
```


Reification

- ✓ Sometimes one wish to make statements about other statements
 - ✓ Idea: refer to a statement using an identifier
 - ✓ RDF allows such reference through a reification mechanism which turns a statement into a resource
- “John says those cherries are sweet”

Reification Example

```
<rdf:Description rdf:about="#949352">  
  <uni:name> Yinghui Wu</uni:name>  
</rdf:Description>
```

✓ reifies as

```
<rdf:Statement rdf:ID="StatementAbout949352">  
  <rdf:subject rdf:resource="#949352"/>  
  <rdf:predicate  
    rdf:resource="http://www.mydomain.org/  
      uni-ns#name"/>  
  <rdf:object>Yinghui Wu</rdf:object>  
</rdf:Statement>
```

Reification

- ✓ To access parts of a statement:
- ✓ Properties
 - **rdf:type** - subject is an *instance* of that category or class defined by the value
 - **rdf:subject**, **rdf:predicate**, **rdf:object** – relate elements of statement tuple to a resource of type statement.
- ✓ Types (or classes)
 - **rdf:Resource** – everything that can be identified (with a URI)
 - **rdf:Property** – specialization of a resource expressing a binary relation between two resources
 - **rdf:statement** – a triple with properties **rdf:subject**, **rdf:predicate**, **rdf:object**

RDF Schema

Basic Ideas of RDF Schema

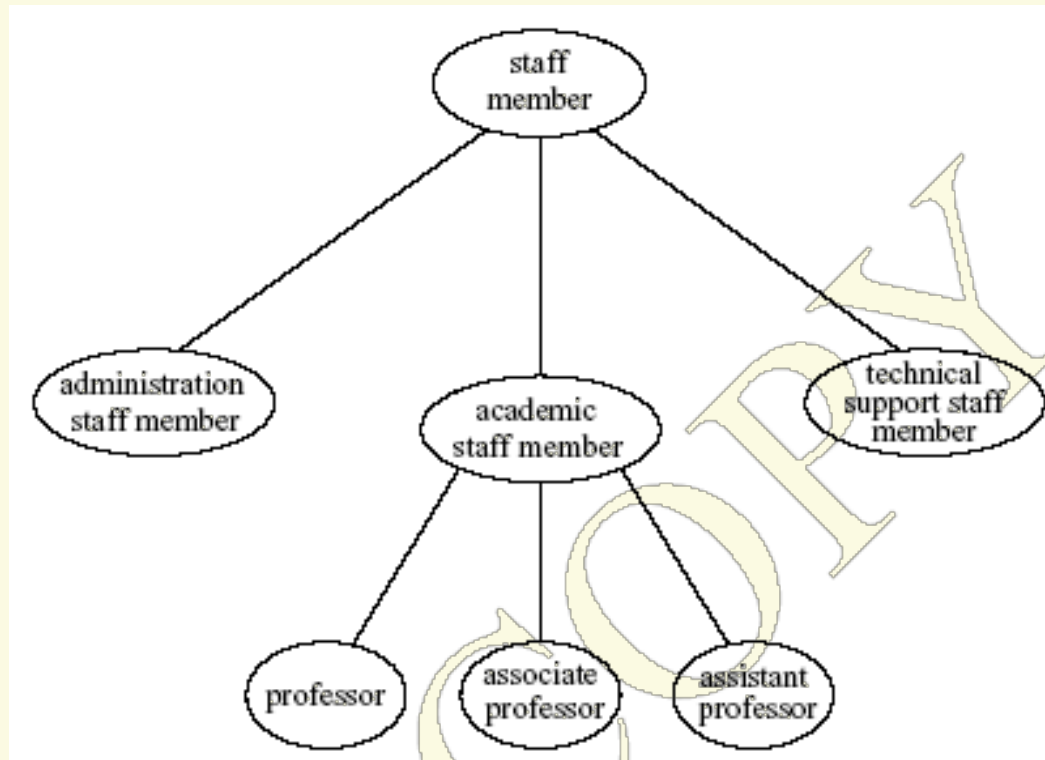
- ✓ RDF is a universal language that lets users describe resources in their own vocabularies
 - RDF does not assume, nor does it define semantics of any particular application domain
- ✓ The user can do so in **RDF Schema** using:
 - Classes and Properties
 - Class Hierarchies and Inheritance
 - Property Hierarchies
- ✓ Enables communities to share machine readable tokens and locally define human readable labels.

Classes and their Instances

- ✓ distinguish between
 - Concrete “things” (individual objects) in the domain: Big Data, WSU, Yinghui Wu etc.
 - Sets of individuals sharing properties called **classes**: lecturers, students, courses etc.
- ✓ Individual objects that belong to a class are referred to as **instances** of that class
- ✓ The relationship between instances and classes in RDF is through **rdf:type**
- ✓ **Specifying type disallow statement such as**
 - “Big data is taught by **Graph theory**”
 - “**Sloan 38** is taught by Yinghui Wu”

Class Hierarchy Example

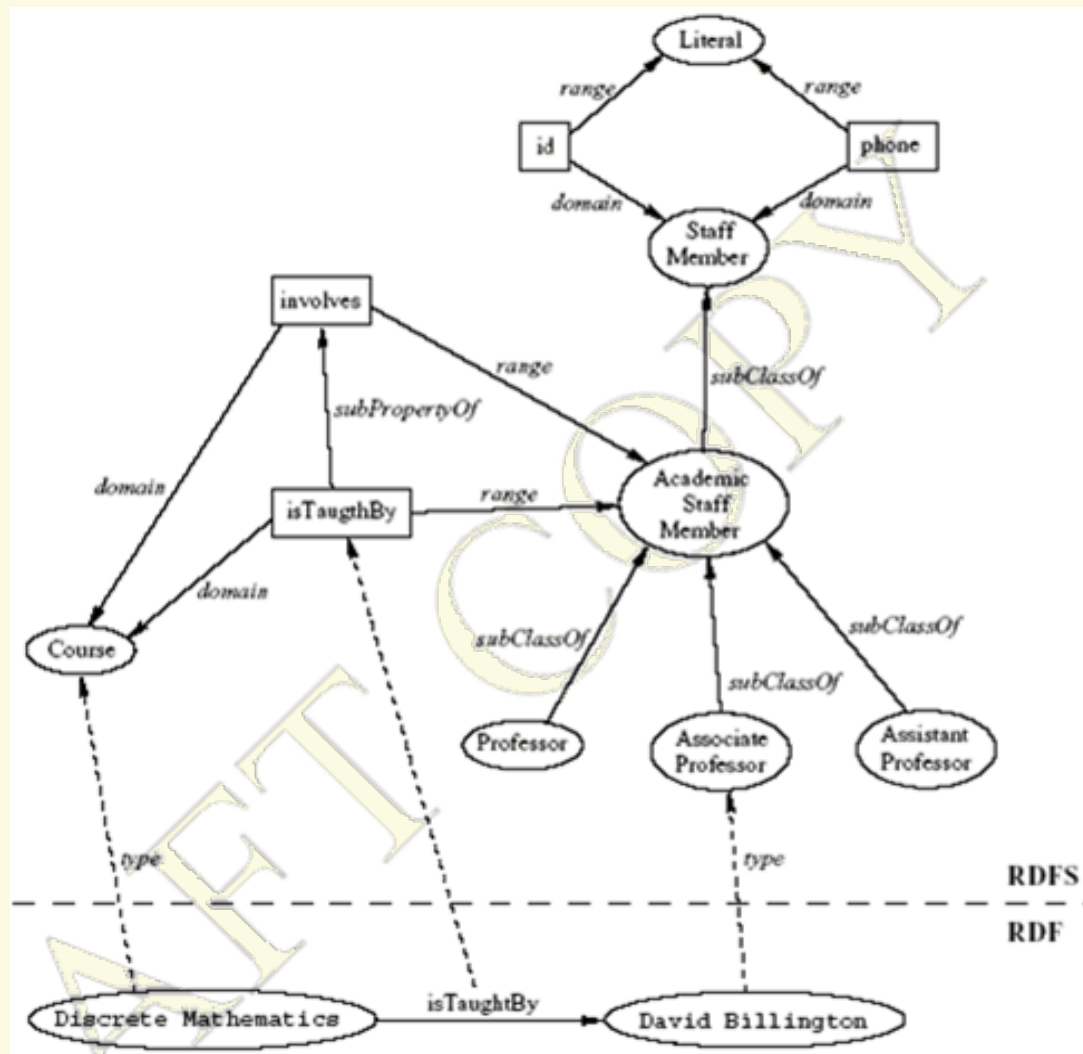
Class-related namespace
rdfs:Class, rdfs:subClassOf



Property Hierarchies

- ✓ Property-related namespace
 - `rdfs:subPropertyOf`, `rdfs:domain`, `rdfs:range`
- ✓ Hierarchical relationships for properties
 - E.g., “is taught by” is a subproperty of “involves”
 - If a course C is taught by an academic staff member A, then C also involves A
- ✓ The converse is not necessarily true
 - a tutor who marks student homework but does not teach C
- ✓ P is a **subproperty** of Q, if $Q(x,y)$ is true whenever $P(x,y)$ is true

RDF Layer vs RDF Schema Layer



RDF Schema in RDF

- ✓ The modeling primitives of RDF Schema are defined using resources and properties (an RDF!)
- ✓ To declare that “lecturer” is a subclass of “academic staff member”
 - Define resources **lecturer**, **academicStaffMember**, and **subClassOf**
 - define property **subClassOf**
 - Write triple (**lecturer**,**subClassOf**,**academicStaffMember**)
- ✓ XML-based syntax of RDF

Core Elements

✓ Core Classes:

- **rdfs:Resource**, the class of all resources
- **rdfs:Class**, the class of all classes
- **rdfs:Literal**, the class of all literals (strings)
- **rdf:Property**, the class of all properties.
- **rdf:Statement**, the class of all reified statements

✓ Core Properties

- **rdf:type**, which relates a resource to its class
- **rdfs:subClassOf**, relates a class to one of its superclasses
- **rdfs:subPropertyOf**, relates a property to one of its superproperties

Transitive



Reification and Containers

- ✓ **rdf:subject**, relates a reified statement to its subject
- ✓ **rdf:predicate**, relates a reified statement to its predicate
- ✓ **rdf:object**, relates a reified statement to its object
- ✓ **rdf:Bag**, the class of bags
- ✓ **rdf:Seq**, the class of sequences
- ✓ **rdf:Alt**, the class of alternatives
- ✓ **rdfs:Container**, which is a superclass of all container classes, including the three above

Utility Properties

- ✓ **rdfs:seeAlso** relates a resource to another resource that explains it
- ✓ **rdfs:isDefinedBy** is a subproperty of **rdfs:seeAlso** and relates a resource to the place where its definition, typically an RDF schema, is found
- ✓ **rdfs:comment**. Comments, typically longer text, can be associated with a resource
- ✓ **rdfs:label**. A human-friendly label (name) is associated with a resource

RDFS Vocabulary: Overview

RDFS introduces the following terms and gives each a meaning w.r.t. the rdf data model

✓ Terms for classes

- [rdfs:Class](#)
- [rdfs:subClassOf](#)

✓ Terms for properties

- [rdfs:domain](#)
- [rdfs:range](#)
- [rdfs:subPropertyOf](#)

✓ Special classes

- [rdfs:Resource](#)
- [rdfs:Literal](#)
- [rdfs:Datatype](#)

• Terms for collections

- [rdfs:member](#)
- [rdfs:Container](#)
- [rdfs:ContainerMembershipProperty](#)

• Special properties

- [rdfs:comment](#)
- [rdfs:seeAlso](#)
- [rdfs:isDefinedBy](#)
- [rdfs:label](#)

RDFS: problems (research in progress)

✓ RDFS **too weak** to describe resources in detail, e.g.

– No *localised range and domain* constraints

Can't say that the range of `hasChild` is `person` when applied to persons and `dog` when applied to dogs

– No *existence/cardinality* constraints

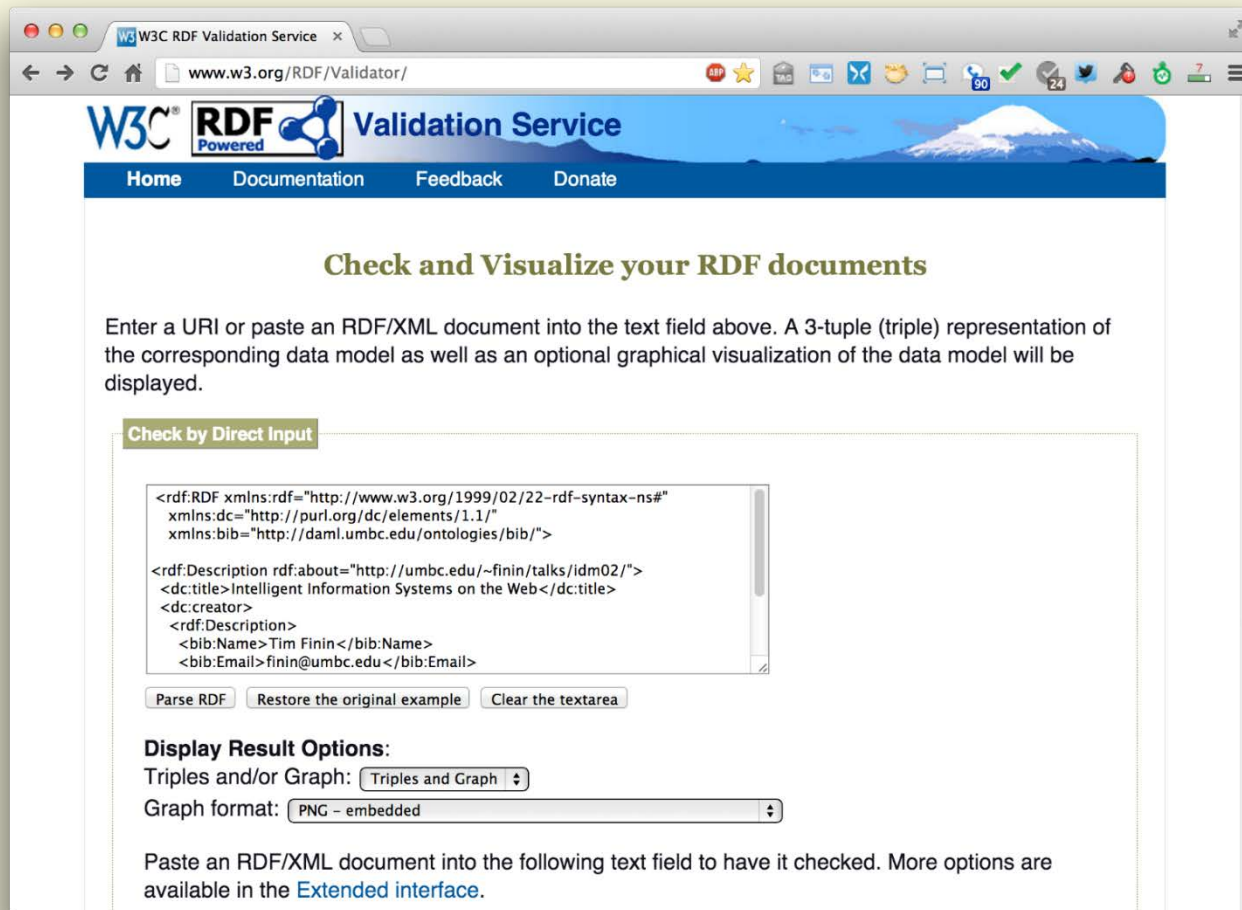
Can't say that all *instances* of `person` have a mother that is also a person, or that persons have exactly 2 parents

– No *transitive, inverse or symmetrical* properties

Can't say `isPartOf` is a transitive property, `hasPart` is the inverse of `isPartOf` or `touches` is symmetrical

✓ need RDF terms providing these and other features.

An RDF validation service



The screenshot shows the W3C RDF Validation Service web interface. The browser's address bar displays www.w3.org/RDF/Validator/. The page features a blue header with the W3C logo, the text "RDF Powered", and "Validation Service". Below the header is a navigation bar with links: Home, Documentation, Feedback, and Donate. The main content area has the heading "Check and Visualize your RDF documents". A paragraph explains that users can enter a URI or paste an RDF/XML document into a text field to get a 3-tuple representation and an optional graphical visualization. A section titled "Check by Direct Input" contains a text area with the following RDF/XML code:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:bib="http://daml.umbc.edu/ontologies/bib/">
  <rdf:Description rdf:about="http://umbc.edu/~finin/talks/idm02/">
    <dc:title>Intelligent Information Systems on the Web</dc:title>
    <dc:creator>
      <rdf:Description>
        <bib:Name>Tim Finin</bib:Name>
        <bib:Email>finin@umbc.edu</bib:Email>
      </rdf:Description>
    </dc:creator>
  </rdf:Description>
</rdf:RDF>
```

Below the text area are three buttons: "Parse RDF", "Restore the original example", and "Clear the textarea". Under the heading "Display Result Options:", there are two dropdown menus. The first is labeled "Triples and/or Graph:" and is set to "Triples and Graph". The second is labeled "Graph format:" and is set to "PNG - embedded". At the bottom, a paragraph states: "Paste an RDF/XML document into the following text field to have it checked. More options are available in the [Extended interface](#)."

<http://www.w3.org/RDF/Validator/uri>

RDF Semantics Inferencing

Semantics based on Inference Rules

- ✓ Semantics in terms of RDF triples instead of restating RDF in terms of first-order logic
- ✓ ... and sound and complete inference systems
- ✓ This inference system consists of **inference rules** of the form:

IF E contains certain triples

THEN add to E certain additional triples

- ✓ where **E** is an arbitrary set of RDF triples

Examples of Inference Rules

**IF E contains the triple (?x,?p,?y)
THEN E also contains (?p,rdf:type,rdf:property)**

**IF E contains the triples (?u,rdfs:subClassOf,?v) and
(?v,rdfs:subClassOf,?w)
THEN E also contains the triple
(?u,rdfs:subClassOf,?w)**

**IF E contains the triples (?x,rdf:type,?u) and
(?u,rdfs:subClassOf,?v)
THEN E also contains the triple (?x,rdf:type,?v)**

Transitivity!

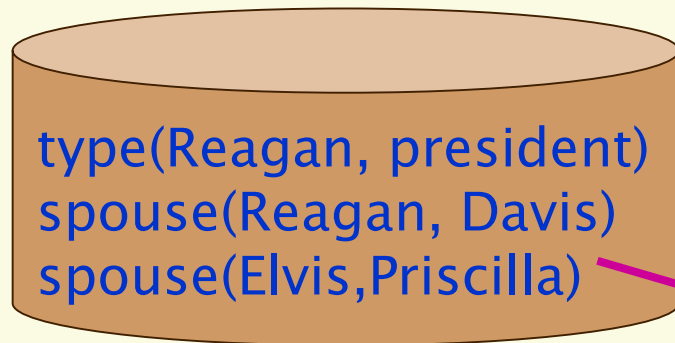
Examples of Inference Rules

- ✓ Any resource **?y** which appears as the value of a property **?p** can be inferred to be a member of the range of **?p**

IF E contains the triples (?x,?p,?y**) and
(**?p,rdfs:range,?u**)**

THEN E also contains the triple (?y,rdf:type,?u**)**

Application in Knowledge extending



"Elvis is married to Priscilla"



"is married to" ~ spouse

Add pattern deduction rules

$\text{occurs}(X, P, Y) \ \& \ \text{means}(X, X') \ \& \ \text{means}(Y, Y') \ \& \ R(X', Y') \Rightarrow P \sim R$

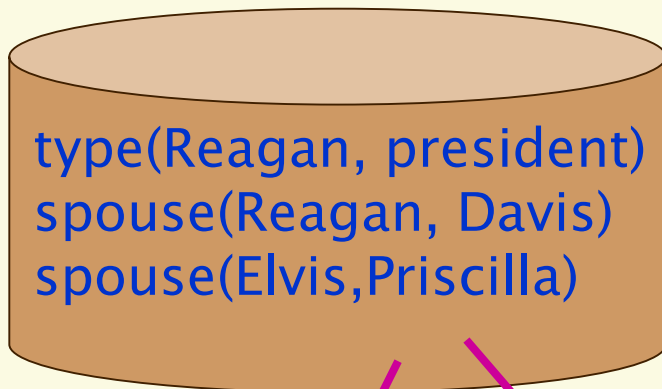
$\text{occurs}(X, P, Y) \ \& \ \text{means}(X, X') \ \& \ \text{means}(Y, Y') \ \& \ P \sim R \Rightarrow R(X', Y')$

Add semantic constraints (manually)

$\text{spouse}(X, Y) \ \& \ \text{spouse}(X, Z) \Rightarrow Y = Z$

(F. Suchanek et al.: WWW'09)

The rules deduce facts from patterns



"Hermione is married to Ron"

"is married to" ~ married



spouse(Hermione, Ronald Reagan)
spouse(Hermione, Ron Weasley)

Add pattern deduction rules

$\text{occurs}(X, P, Y) \ \& \ \text{means}(X, X') \ \& \ \text{means}(Y, Y') \ \& \ R(X', Y') \Rightarrow P \sim R$

$\text{occurs}(X, P, Y) \ \& \ \text{means}(X, X') \ \& \ \text{means}(Y, Y') \ \& \ P \sim R \Rightarrow R(X', Y')$

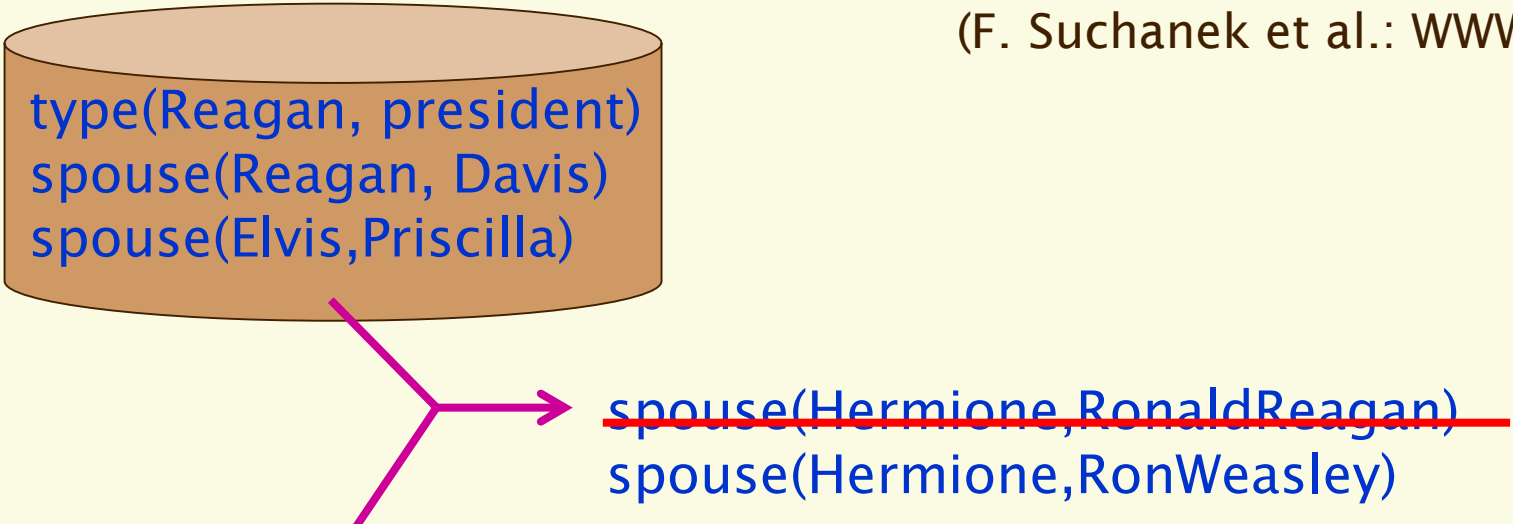
Add semantic constraints (manually)

$\text{spouse}(X, Y) \ \& \ \text{spouse}(X, Z) \Rightarrow Y = Z$

(F. Suchanek et al.: WWW'09)

The rules remove inconsistencies

(F. Suchanek et al.: WWW'09)



type(Reagan, president)
spouse(Reagan, Davis)
spouse(Elvis, Priscilla)

~~spouse(Hermione, RonaldReagan)~~
spouse(Hermione, RonWeasley)

Add pattern deduction rules

$\text{occurs}(X, P, Y) \ \& \ \text{means}(X, X') \ \& \ \text{means}(Y, Y') \ \& \ R(X', Y') \Rightarrow P \sim R$

$\text{occurs}(X, P, Y) \ \& \ \text{means}(X, X') \ \& \ \text{means}(Y, Y') \ \& \ P \sim R \Rightarrow R(X', Y')$

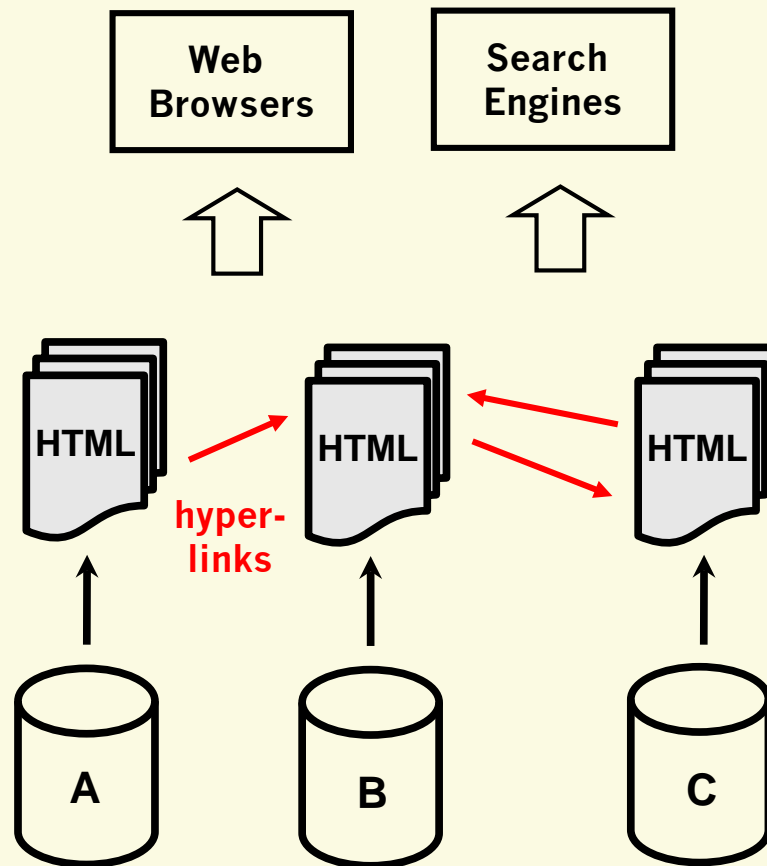
Add semantic constraints (manually)

$\text{spouse}(X, Y) \ \& \ \text{spouse}(X, Z) \Rightarrow Y = Z$



RDF and Linked data

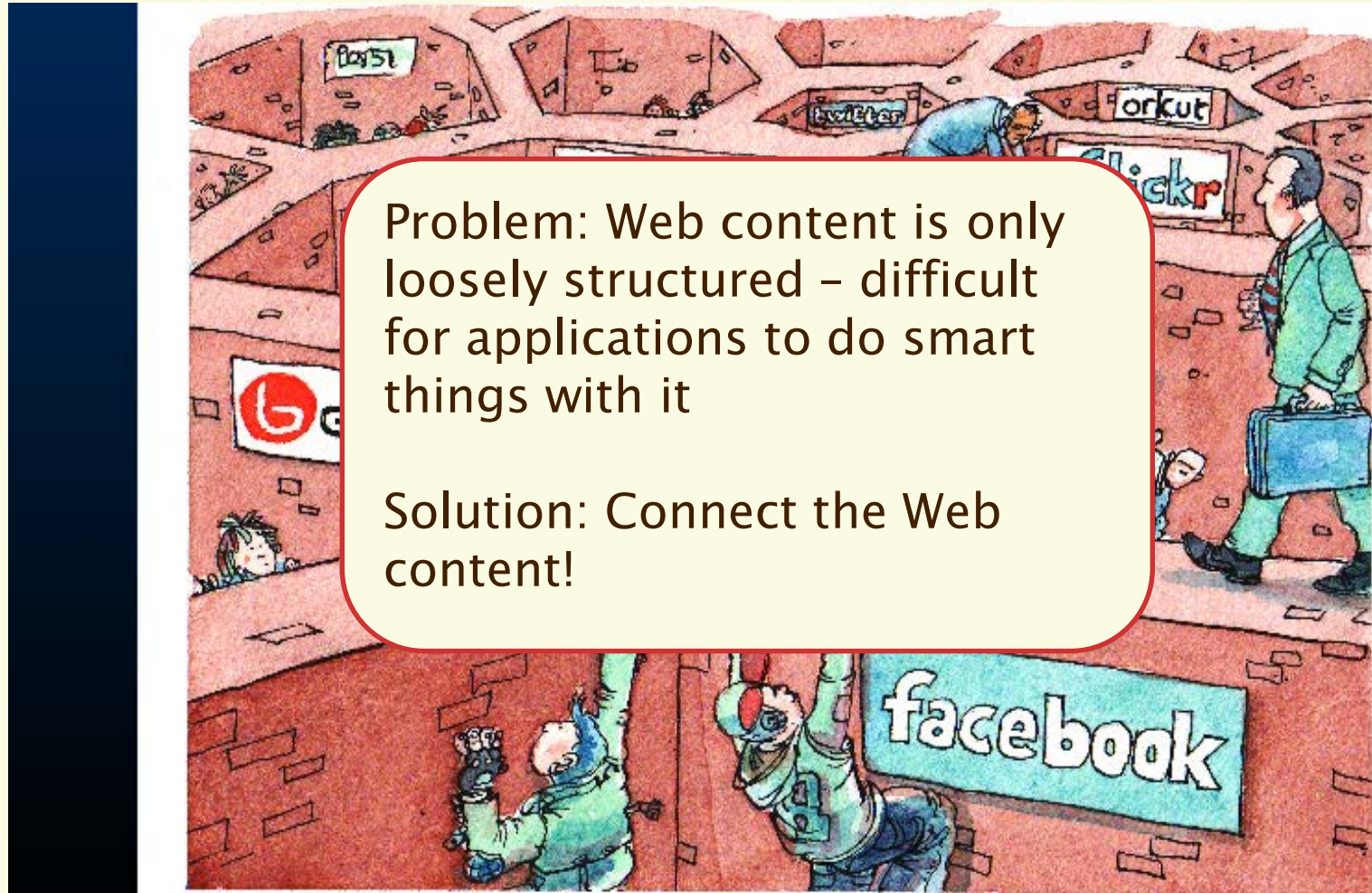
The Classic Web



Single Global Information Space

1. URLs as
 - globally unique IDs
 - retrieval mechanism
2. HTML as shared content format
3. Hyperlinks

Background: the rise of linked data

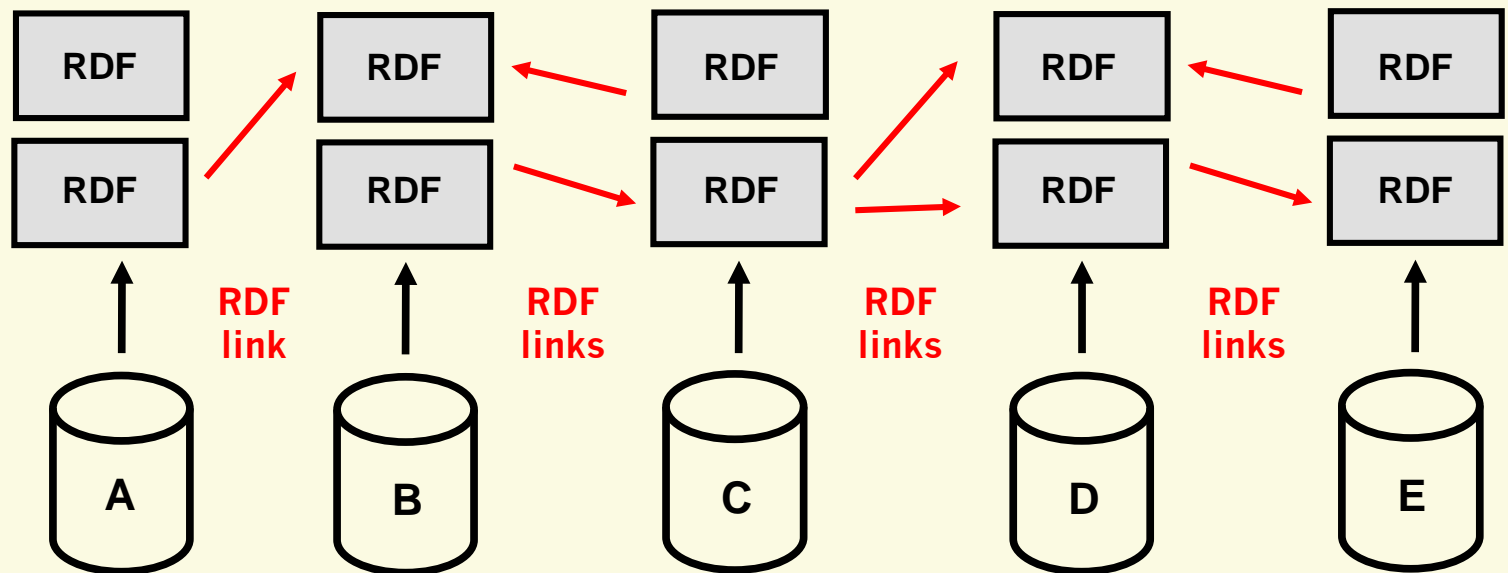


Problem: Web content is only loosely structured – difficult for applications to do smart things with it

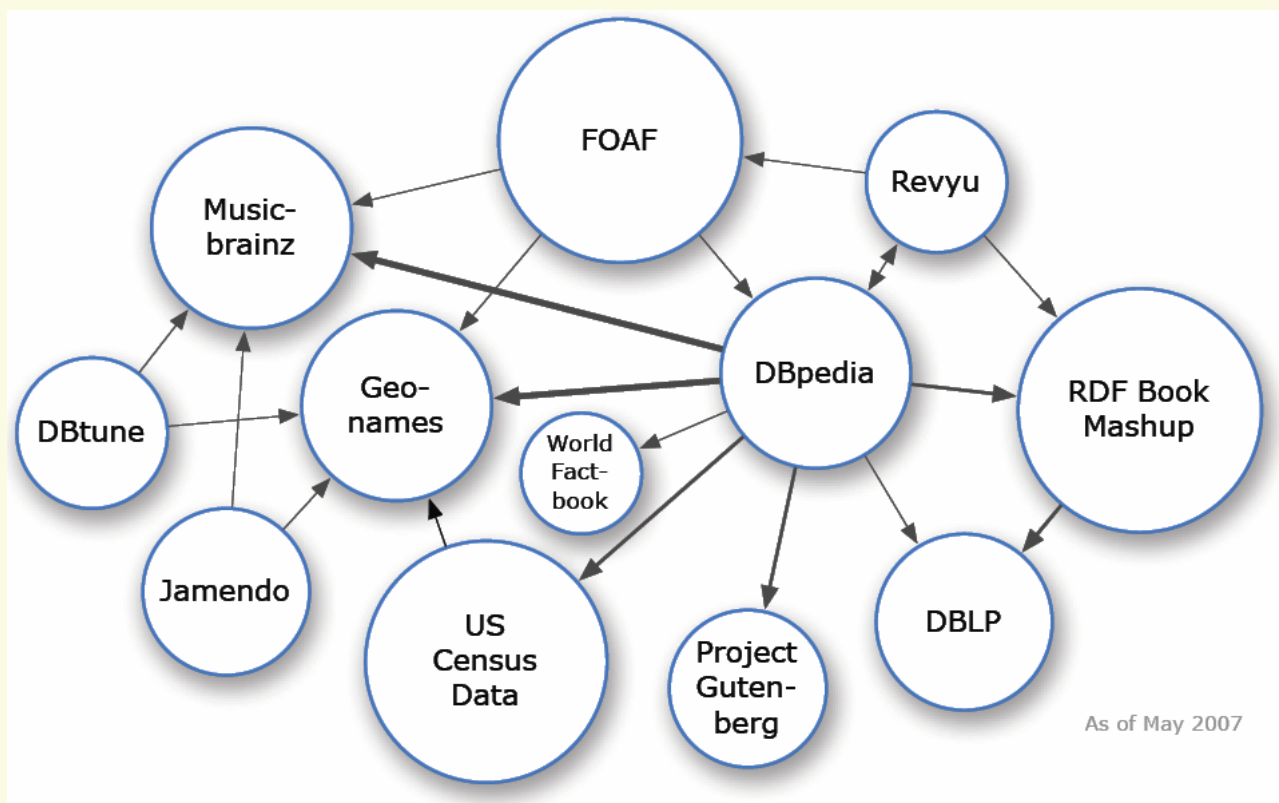
Solution: Connect the Web content!

The idea of Linked Data

- ✓ Use Semantic Web technologies to publish (semi)structured data on the Web,
- ✓ Set links between data from one data source to data within other data sources.

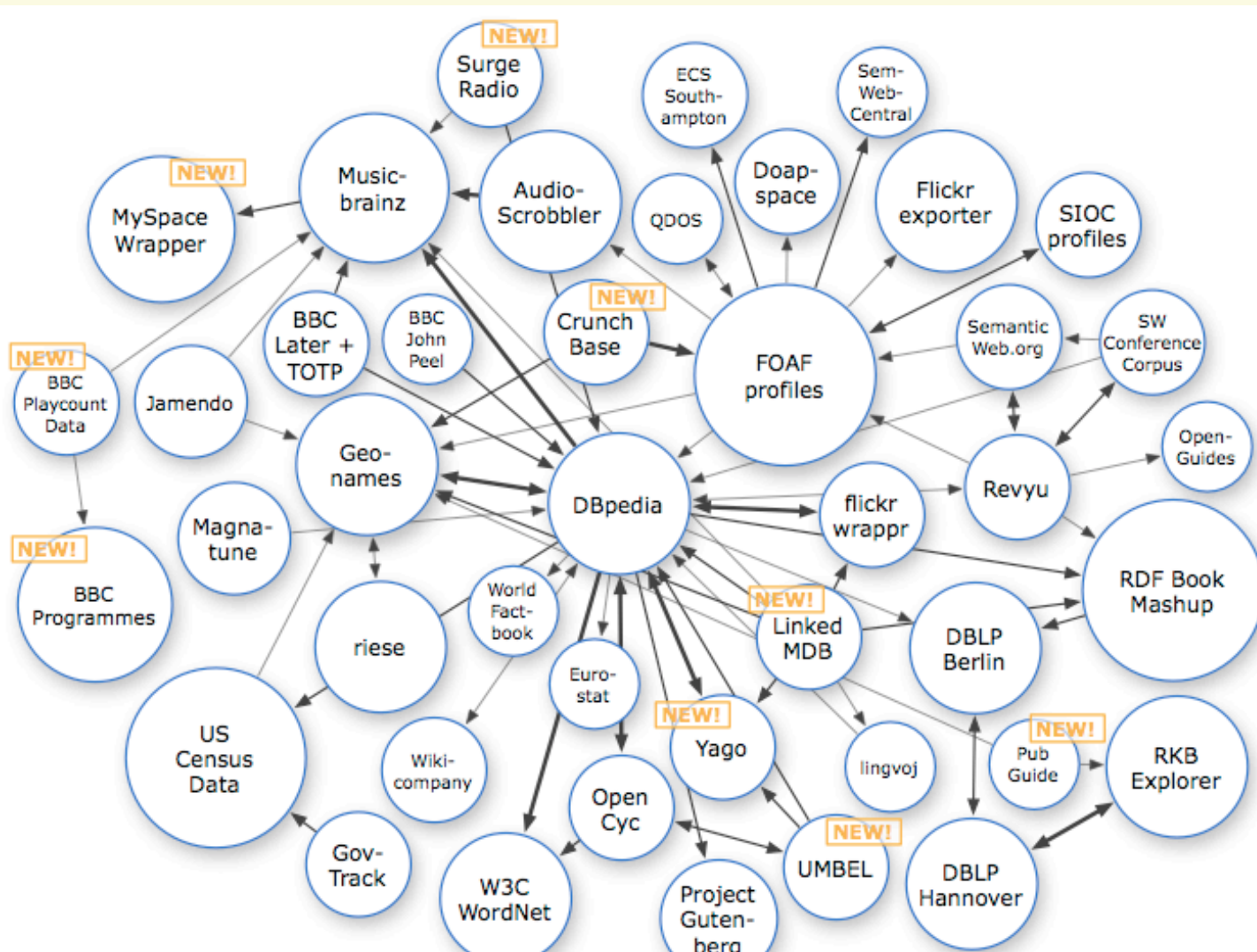


W3C Linking Open Data Project

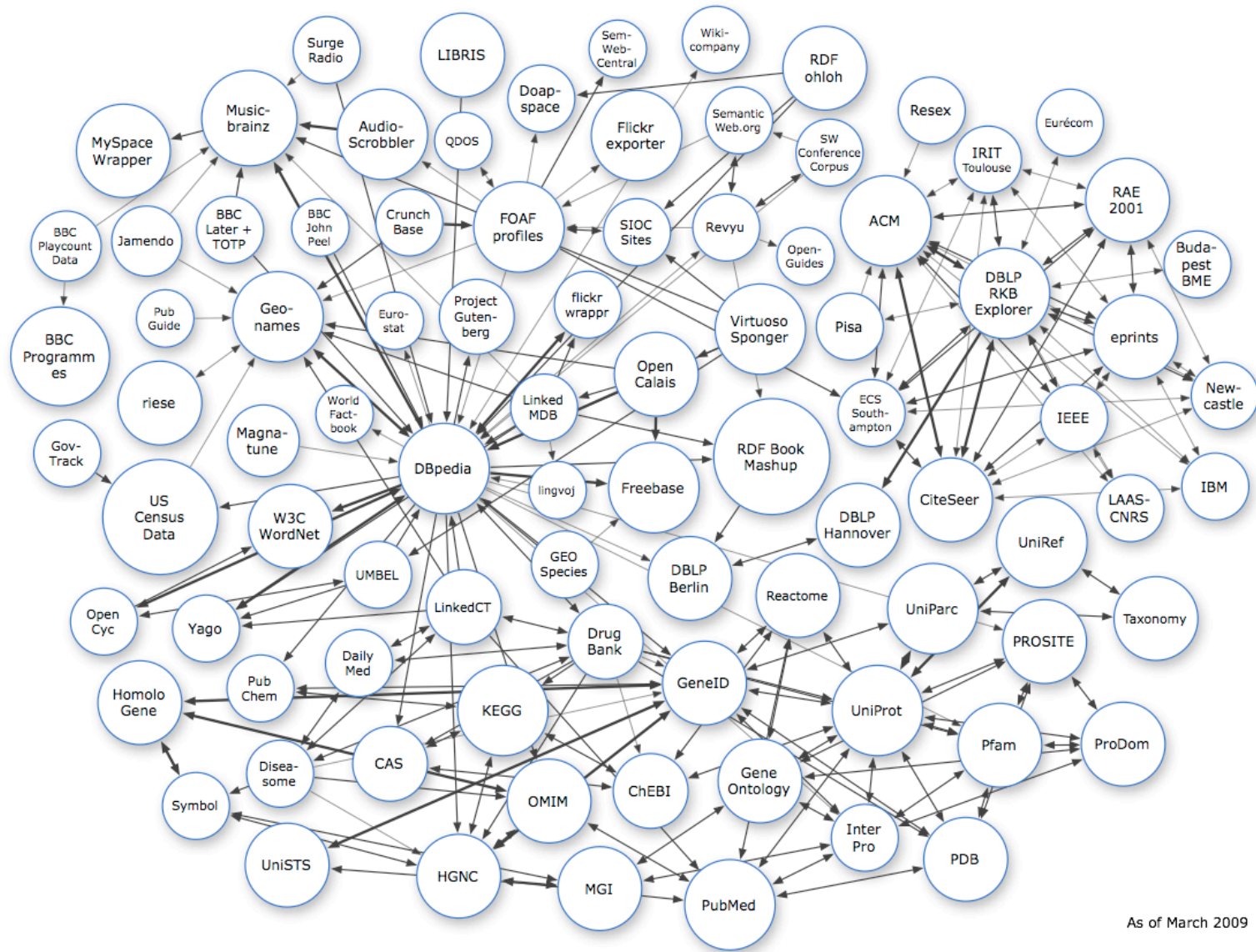


May 2007: 500 million RDF triples,
120,000 RDF links between data sources

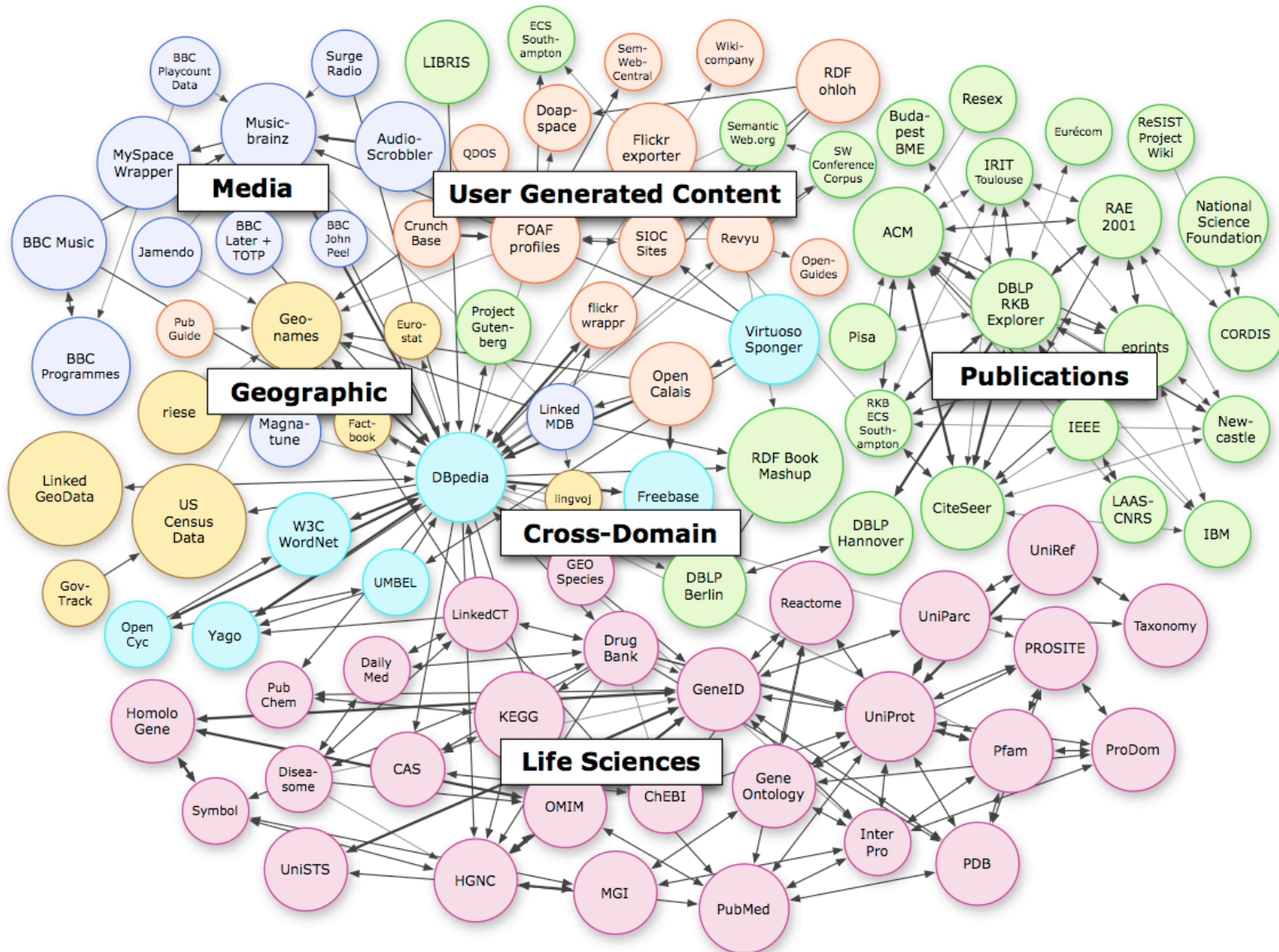
LOD Datasets on the Web: September 2008



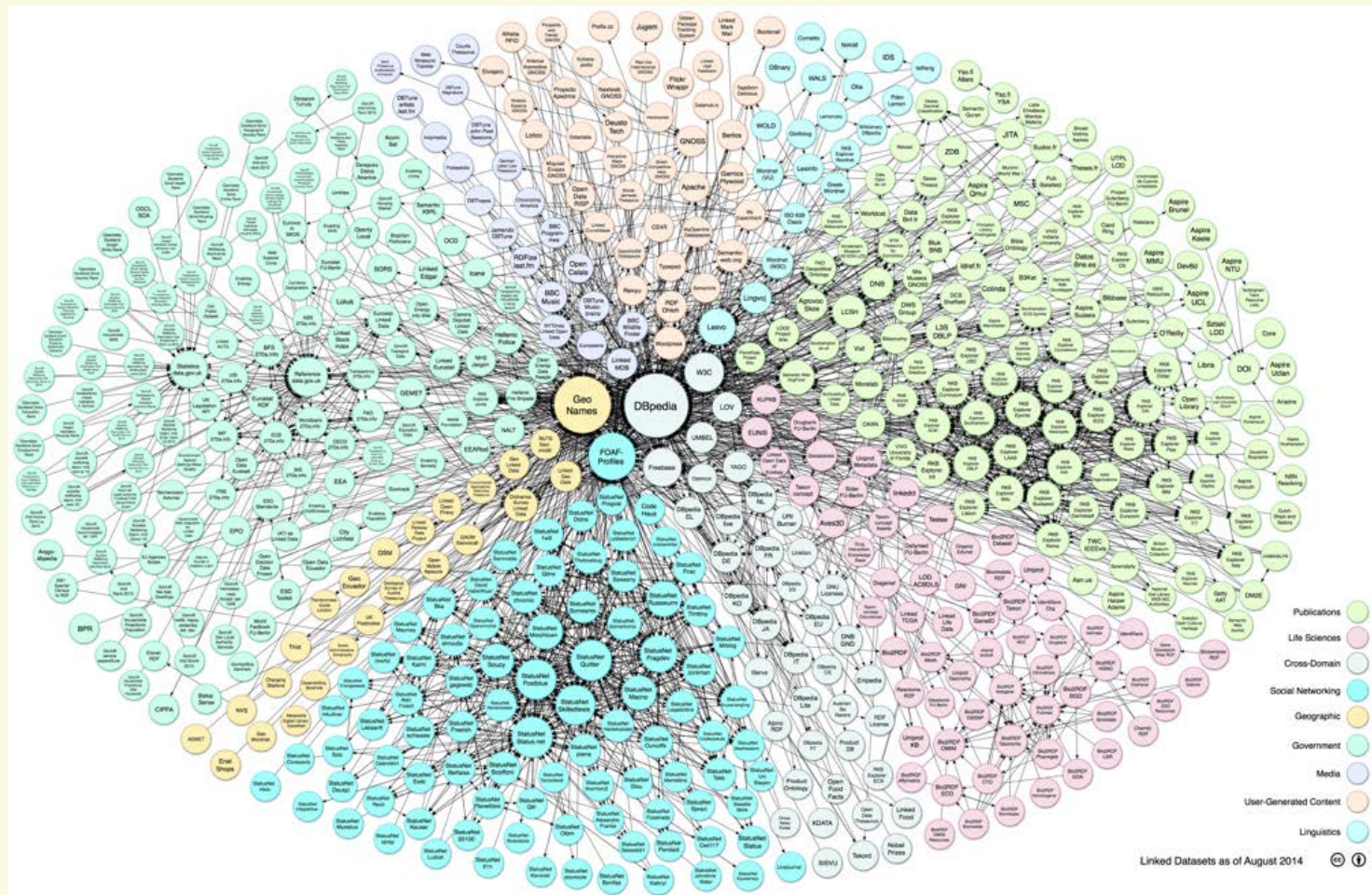
LOD Datasets on the Web: March 2009



LOD Datasets on the Web: July 2009

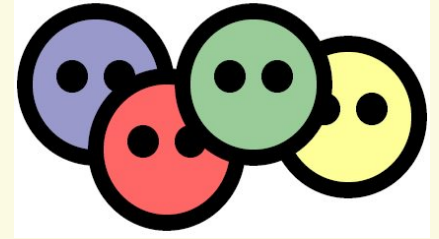


LOD cloud on the Web: 2014.8: 31 Billion triples



<http://lod-cloud.net/>

FOAF



✓ FOAF (Friend of a Friend) is a simple ontology to describe people and their social networks.

- the foaf project page: <http://www.foaf-project.org/>

✓ In 2008: over 1,000,000 valid RDF FOAF files.

- Most of these are from the <http://liveJournal.com/> blogging system which encodes basic user info in foaf
- See <http://apple.cs.umbc.edu/semdis/wob/foaf/>

```
<foaf:Person>
```

```
  <foaf:name>Tim Finin</foaf:name>
```

```
  <foaf:mbox_sha1sum>2410...37262c252e</foaf:mbox_sha1sum>
```

```
  <foaf:homepage rdf:resource="http://umbc.edu/~finin/" />
```

```
  <foaf:img rdf:resource="http://umbc.edu/~finin/images/passport.gif" />
```

```
</foaf:Person>
```

FOAF: why RDF? Extensibility!

- ✓ FOAF vocabulary provides 50+ basic terms for making simple claims about people
- ✓ FOAF files can use other RDF terms too: RSS, MusicBrainz, Dublin Core, Wordnet, Creative Commons, blood types, starsigns, ...
- ✓ RDF gives freedom of independent extension
 - OWL provides fancier data-merging facilities
- ✓ Freedom to say what you like, using any RDF markup you want, and have RDF crawlers merge your FOAF documents with other's and know when you're talking about the same entities.

Is RDF(S) better than XML?

Q: For a specific application, should I use XML or RDF?

A: It depends...

✓ XML's model is

- a tree, i.e., a strong hierarchy
- applications may rely on hierarchy position
- relatively simple syntax and structure
- not easy to *combine* trees

✓ RDF's model is

- a *loose* collections of relations
- applications may do database-like search
- not easy to recover hierarchy
- easy to combine relations in one big collection
- great for the integration of heterogeneous information

Summary

- ✓ RDF provides a foundation for representing and processing metadata
- ✓ RDF has a graph-based data model
- ✓ RDF has an XML-based syntax to support syntactic interoperability
 - XML and RDF complement each other because RDF supports semantic interoperability
- ✓ RDF has a decentralized philosophy and allows incremental building of knowledge, and its sharing and reuse

Summary (2)

- ✓ RDF is domain-independent
 - RDF Schema provides a mechanism for describing specific domains
- ✓ RDF Schema is a primitive ontology language
 - It offers certain modelling primitives with fixed meaning
- ✓ Key concepts of RDF Schema are class, subclass relations, property, subproperty relations, and domain and range restrictions
- ✓ There exist query languages for RDF and RDFS, including SPARQL