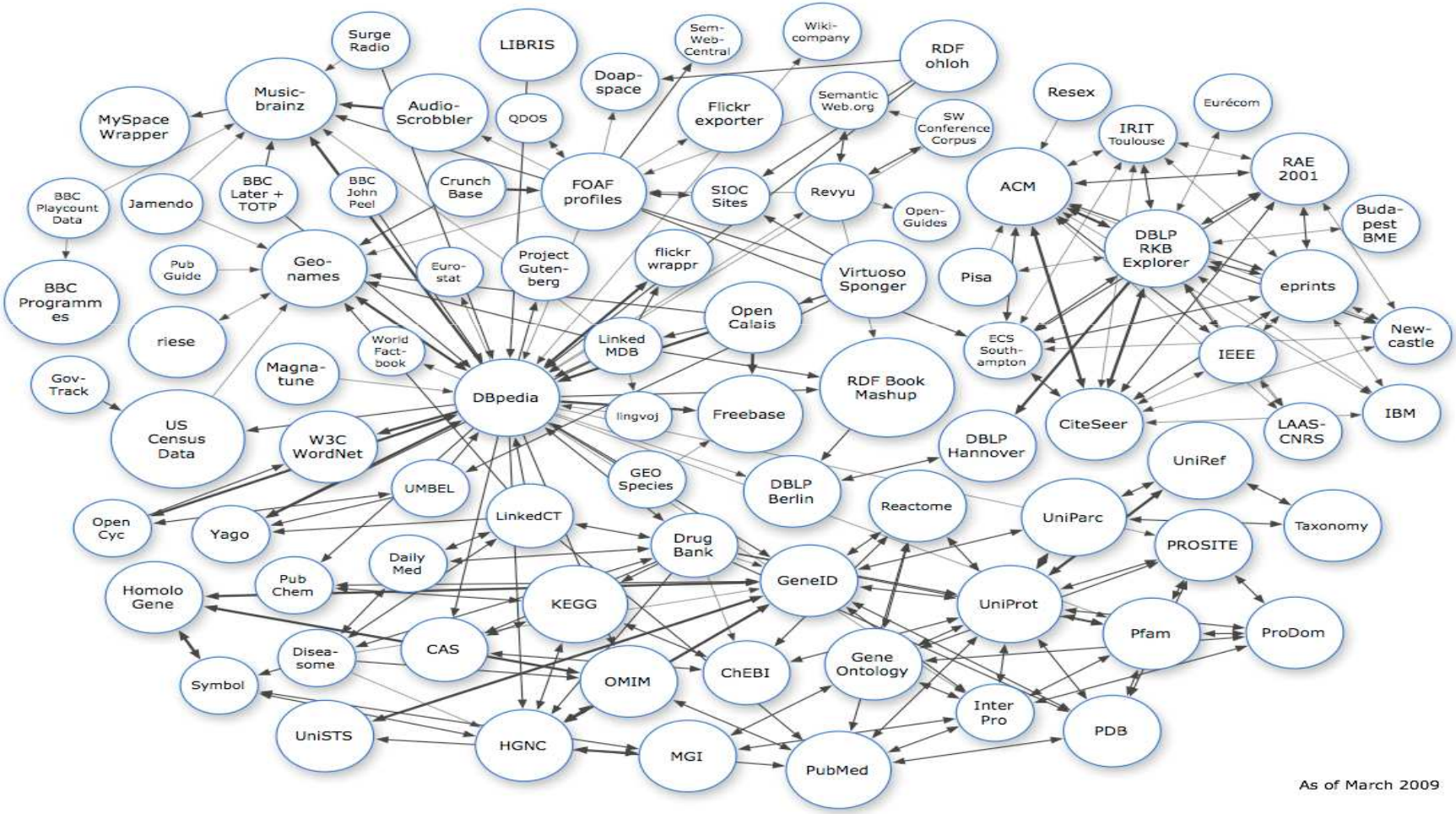
Linked (Open) Data

Freeing Data from the Tyranny of the Application

Brian McBride

A Web of Data/Information

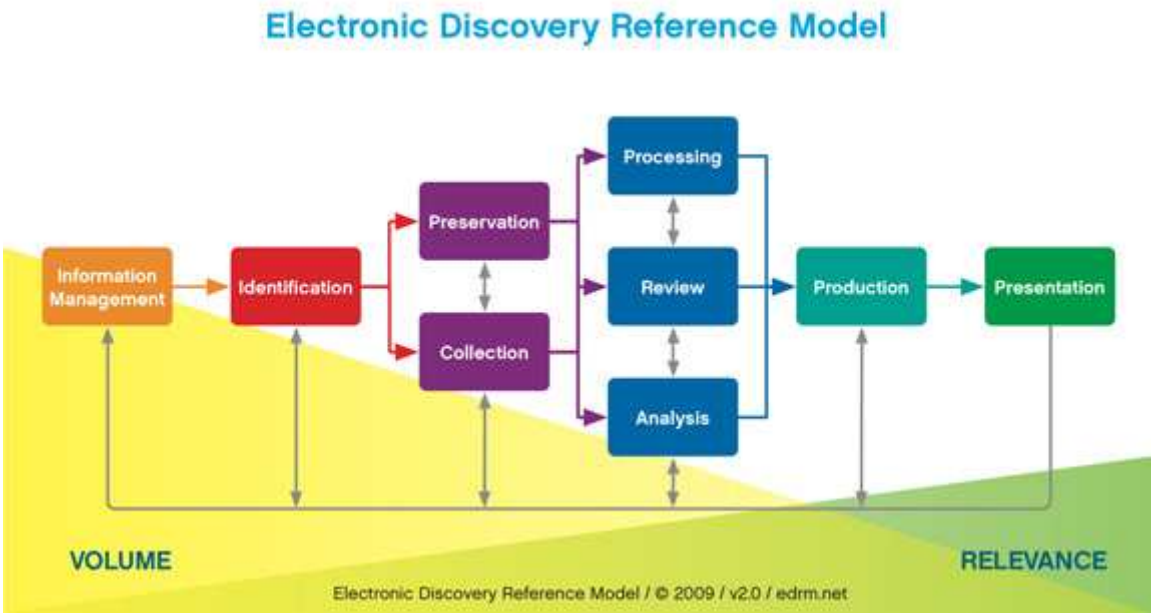


Source:
http://www4.wiwiss.fu-berlin.de/bizer/pub/lod-datasets_2009-03-05.html

e-discovery

- Producing evidence in the form of ESI
- Preserve, find, filter, produce
- Find the right people? How?

• Who committed code to the search Module?
• Who did the report to?
• Who was the most senior developer reporting to that manager?
• Who had access rights to commit the marketing materials?



Supply Chain Information Sharing
Sustainability Labelling

- Sustainability is a major issue
  - We need to change our behaviour
- Educate and Inform
- The Sustainability Consortium
  - Label products with e.g. their carbon footprint
- Publish the data
  - Compute your data from that of your suppliers
  - Find suppliers with better processes
  - Improve your footprint

Government

- Informing the citizen – democracy in the internet age
  - Keeping the government honest
  - Forestalling the lobbyists (e.g. Obama and healthcare)
- Information is the lubricant of the economy
  - The better it flows – the better off we will be
- Priming a knowledge economy

Yes Minister

Gov minister:	Humphrey, I want you to publish all our data.
Sir Humphrey: (smiling)	That would be a very bold move Minister.
Gov minister: (alarmed)	Oh would it? Oh dear. The Prime Minister wants us to publish our data!
Sir Humphrey:	Don't worry minister. My colleagues and I have agreed to set up an inter-departmental committee with a brief to identify all the information that might be published by government now or in the future and to agree a rich an extensible data model to fully express that information, fully interlinked, and able to represent all department's viewpoints on the data and efficiently support all likely queries, following which we will initiate an activity to harmonize that data model with those produced by similar initiatives in Europe.
Gov minister:	You mean you've buried it Humphrey?
Sir Humphrey:	Yes minister.

Publishing Data Web Style

- Just publish it
  - No need to agree a schema
- But we also want to link it together
  - Just putting some spreadsheets on the web doesn't make it easy to link the data up

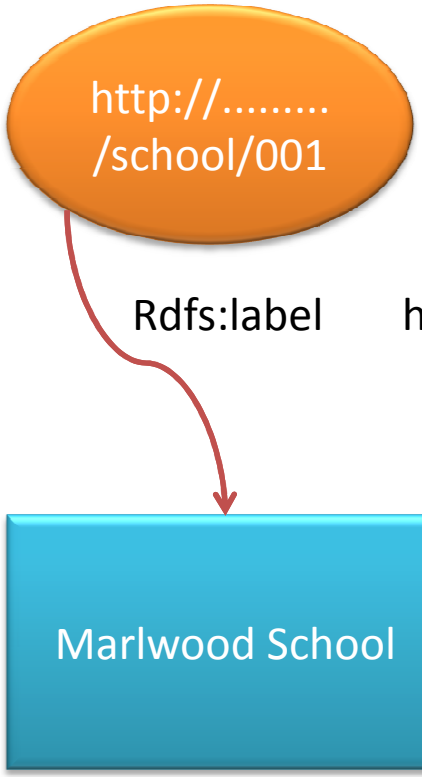
Linked Open Data Principles (Tim Berners-Lee)

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- Include links to other URIs. so that they can discover more things.

Source: http://www.w3.org/DesignIssues/LinkedData

The RDF Data Model
Name ‘things’ with URIs

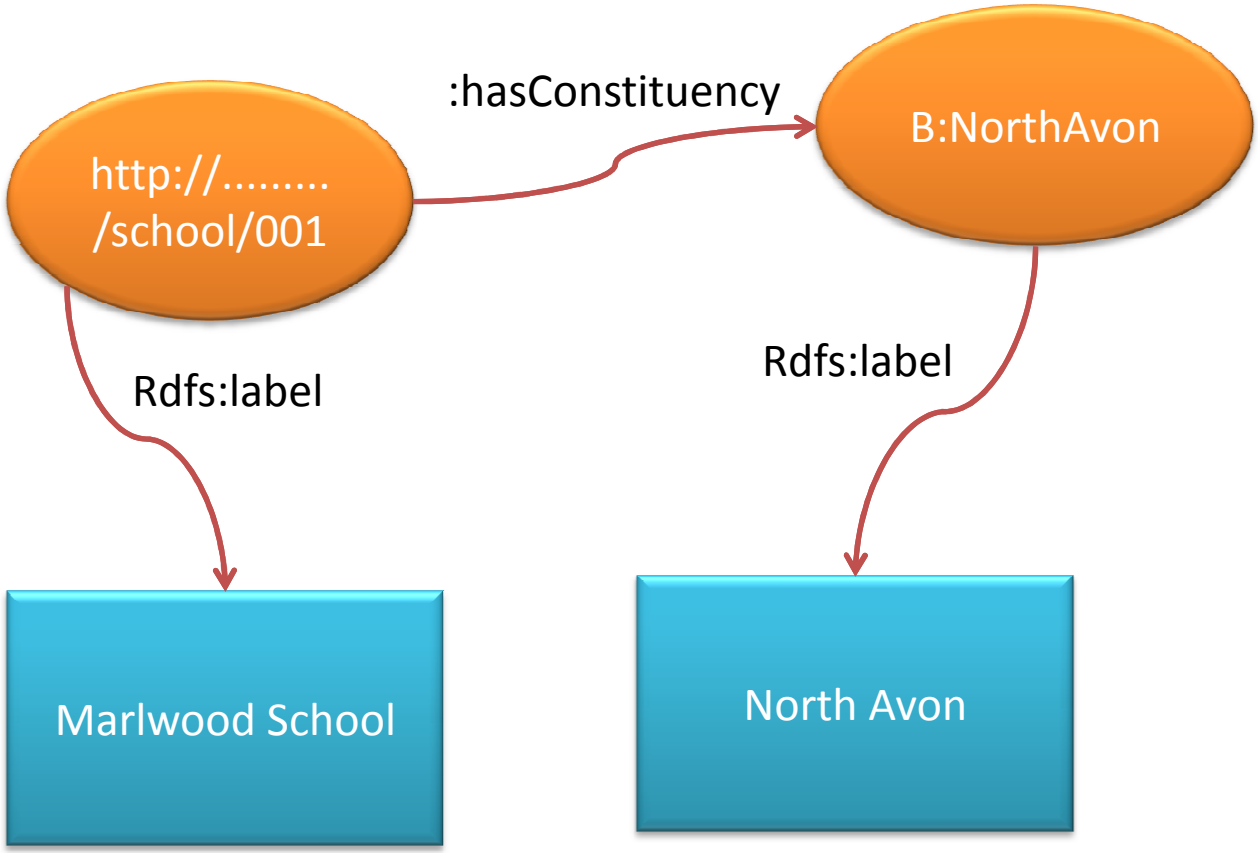
Resources have Properties which are named by URIs



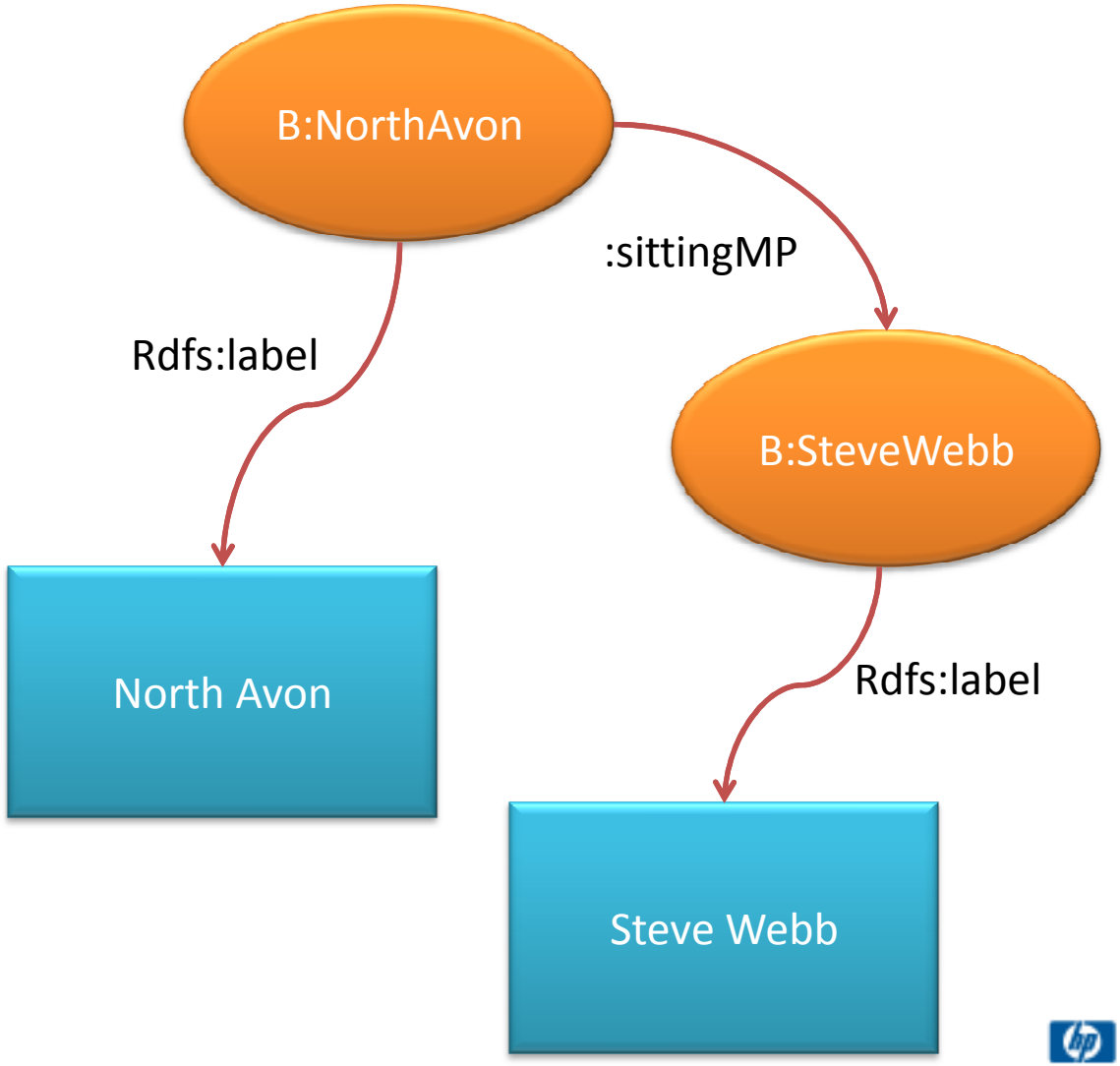
Unlike in Object Oriented Programming Languages, properties are first class entities.

http://www.w3.org/2000/01/rdf-schema#label

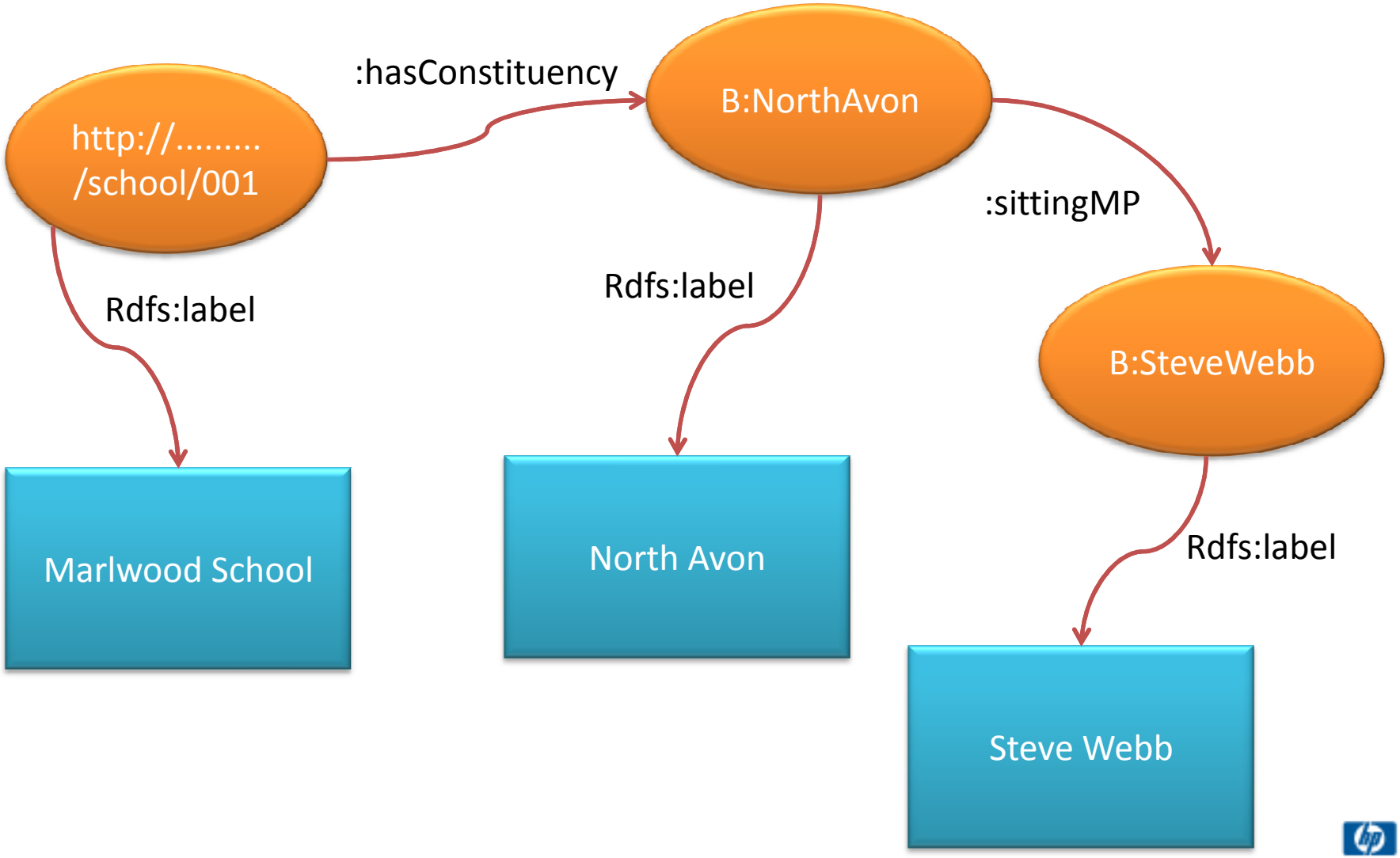
Property Values can be resources too



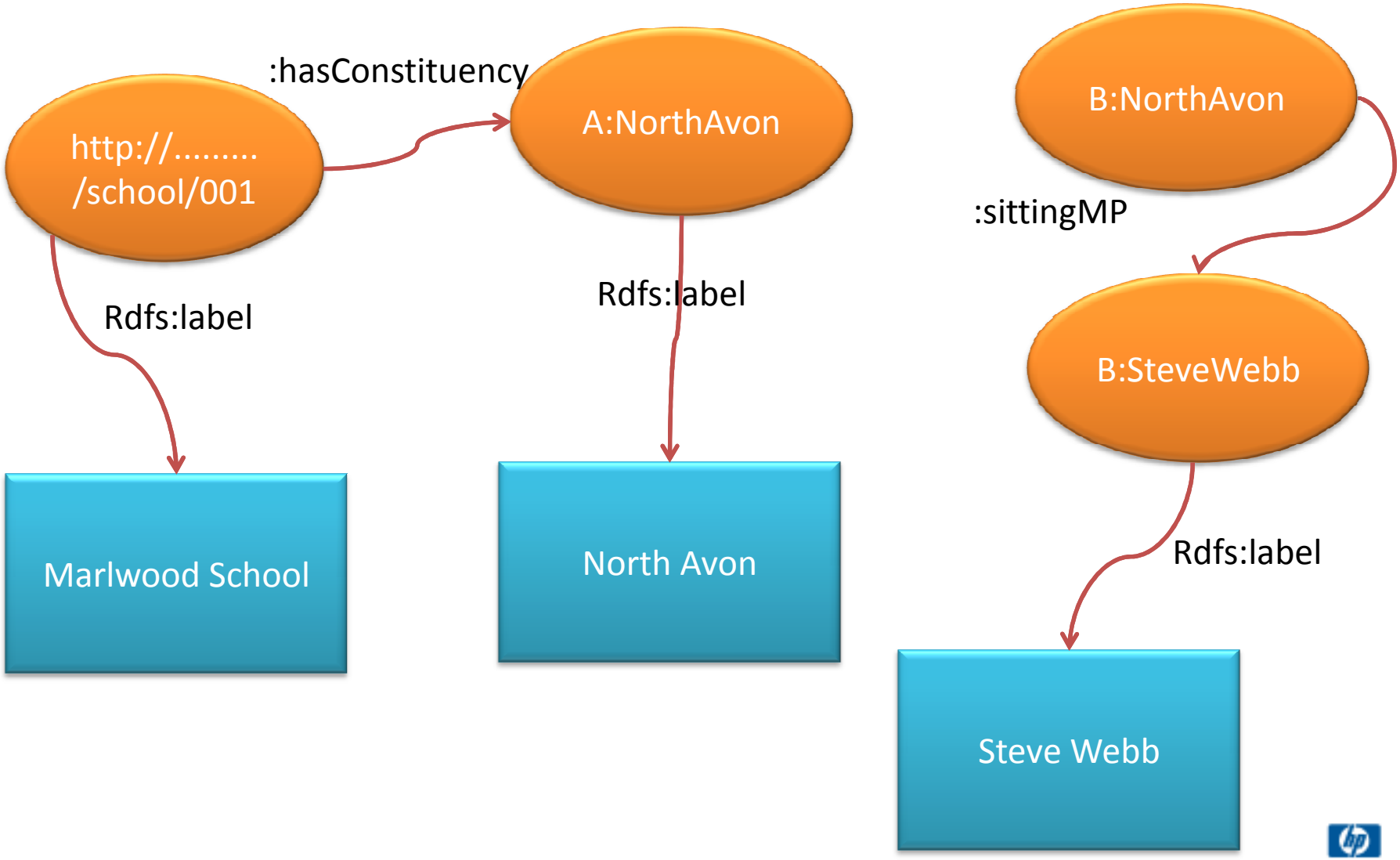
Reuse existing URIs for resources



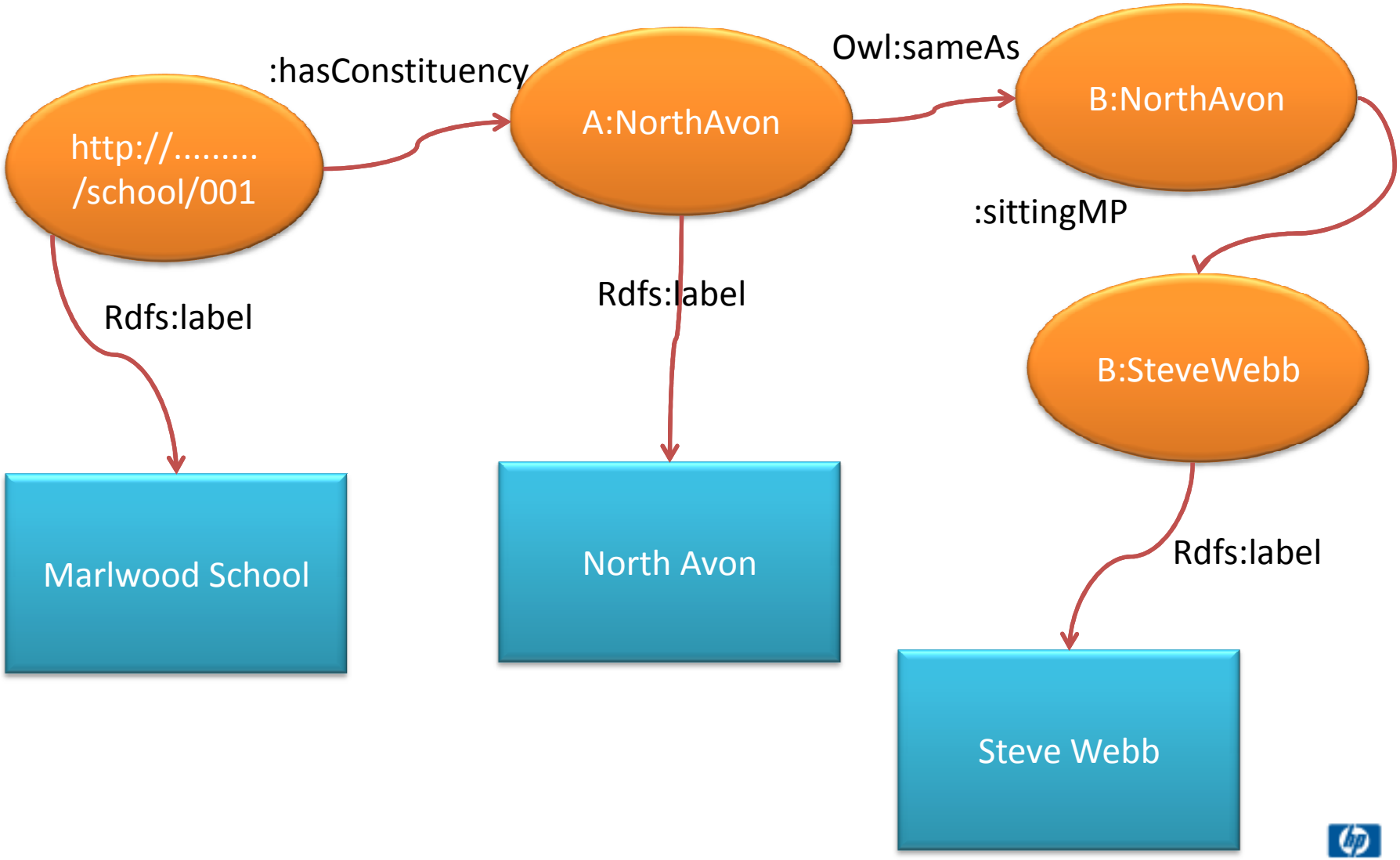
And good things happen



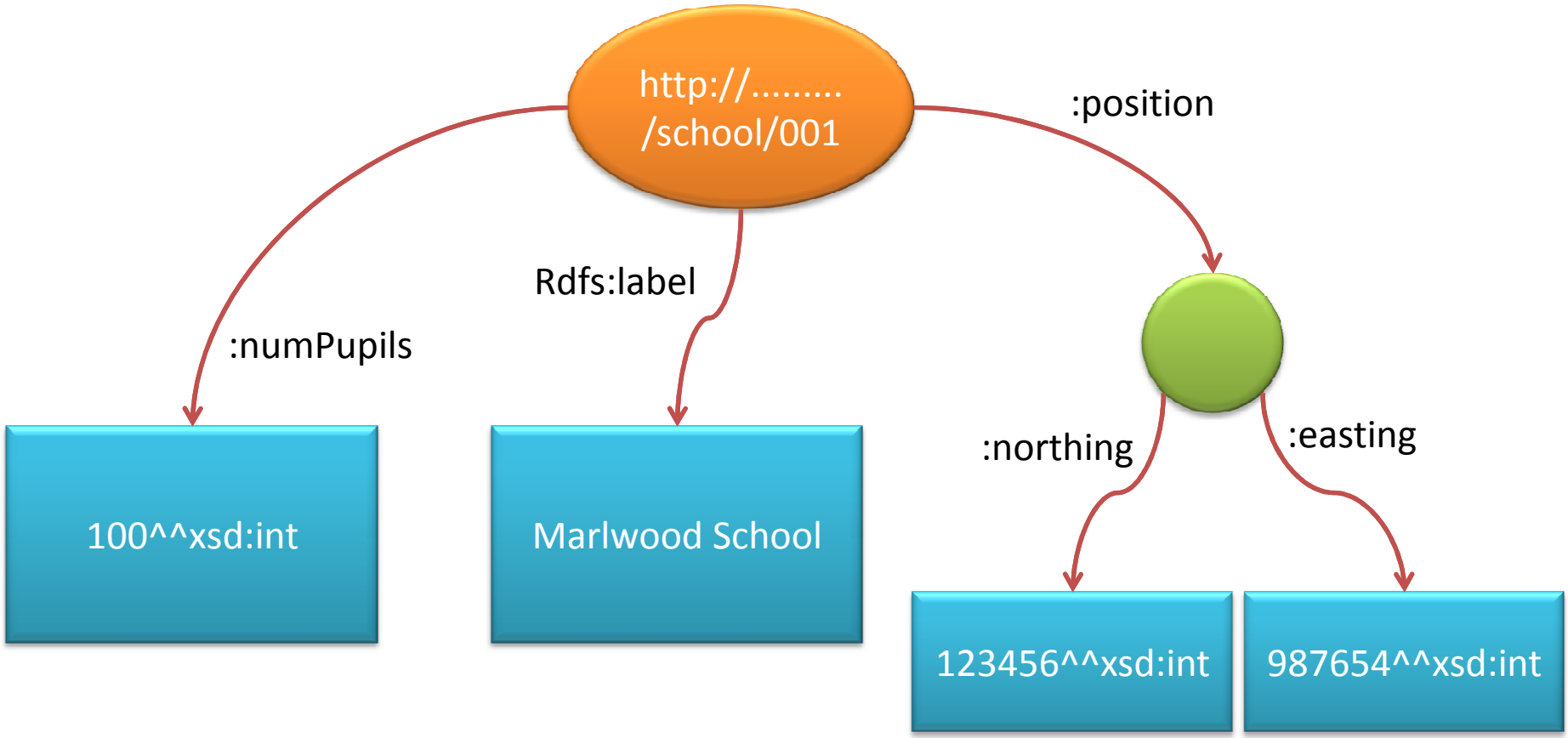
And if they didn't



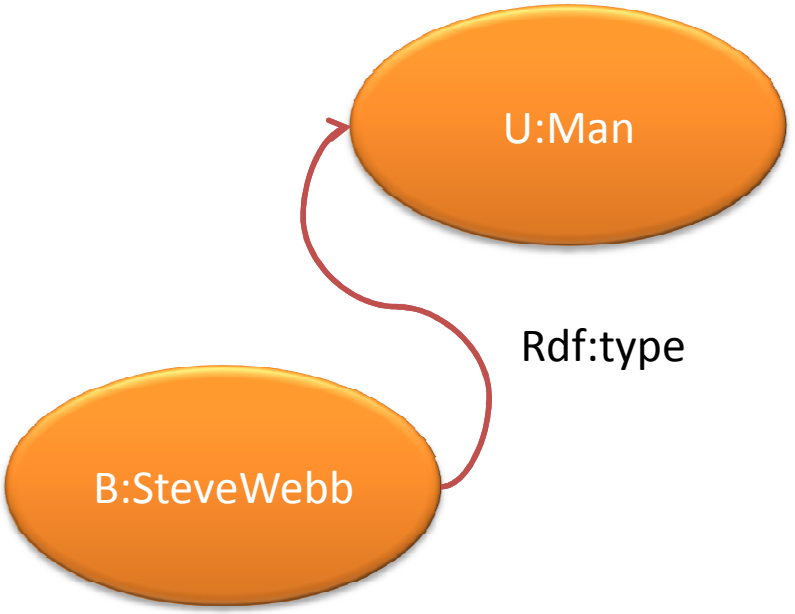
Use owl:sameAs



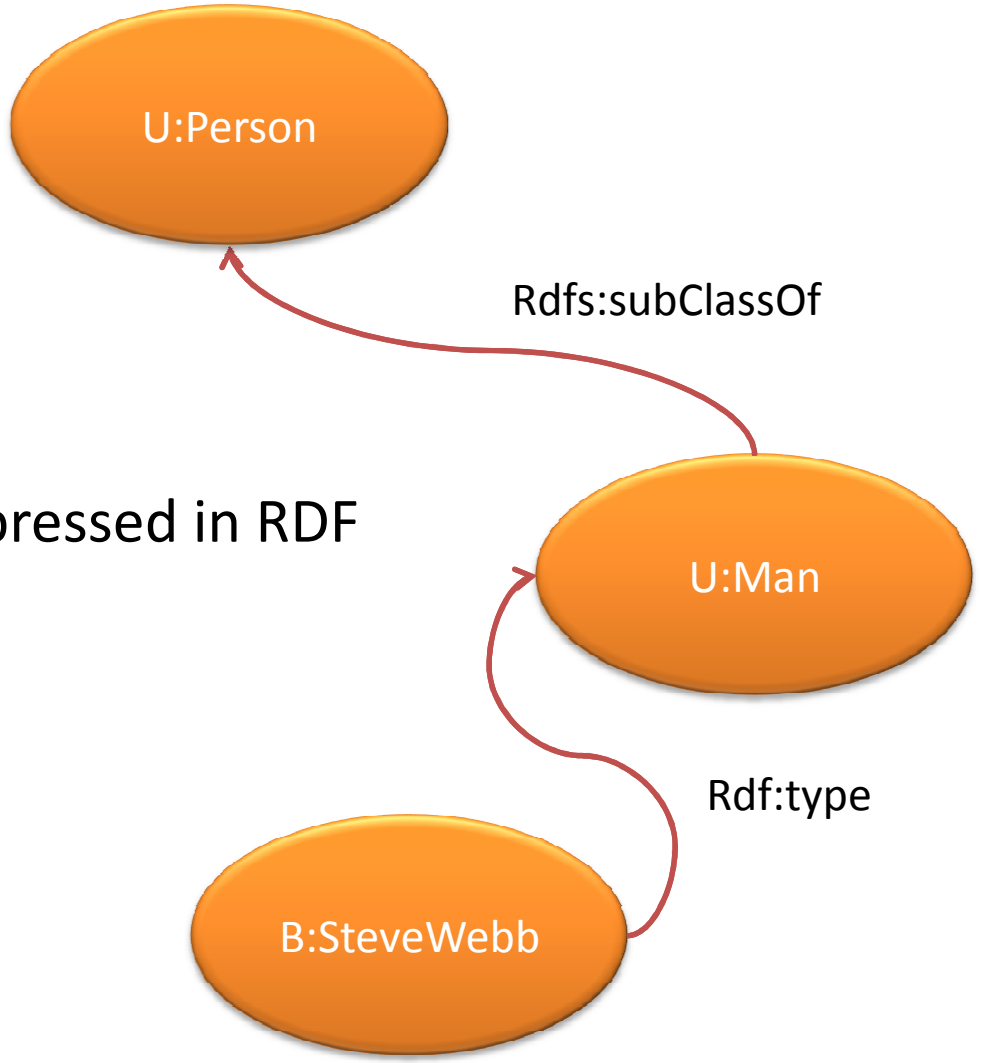
Datatypes, blank nodes and structured values



RDF Schema
A Simple Modeling Language

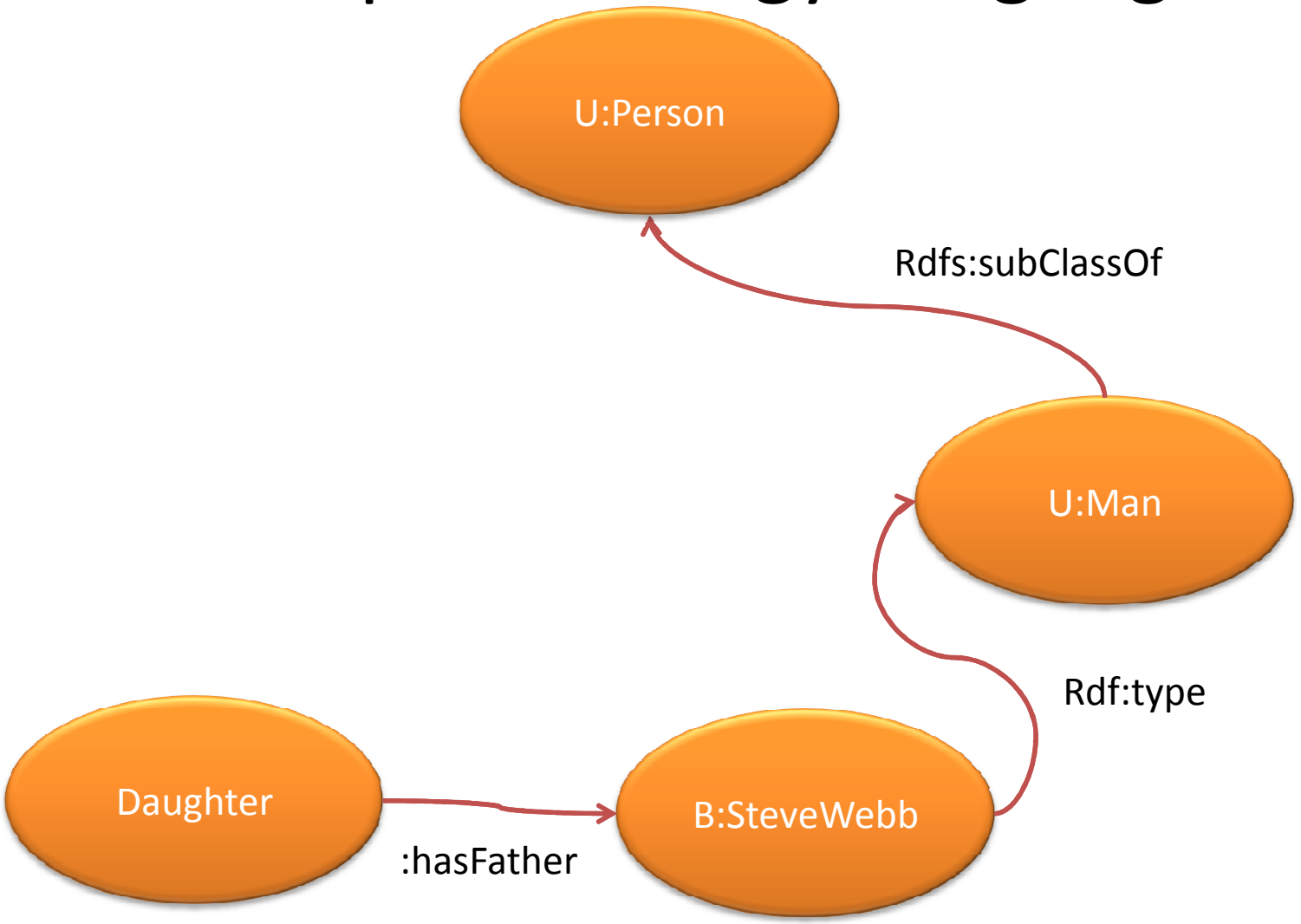


RDF Schema
Subclass

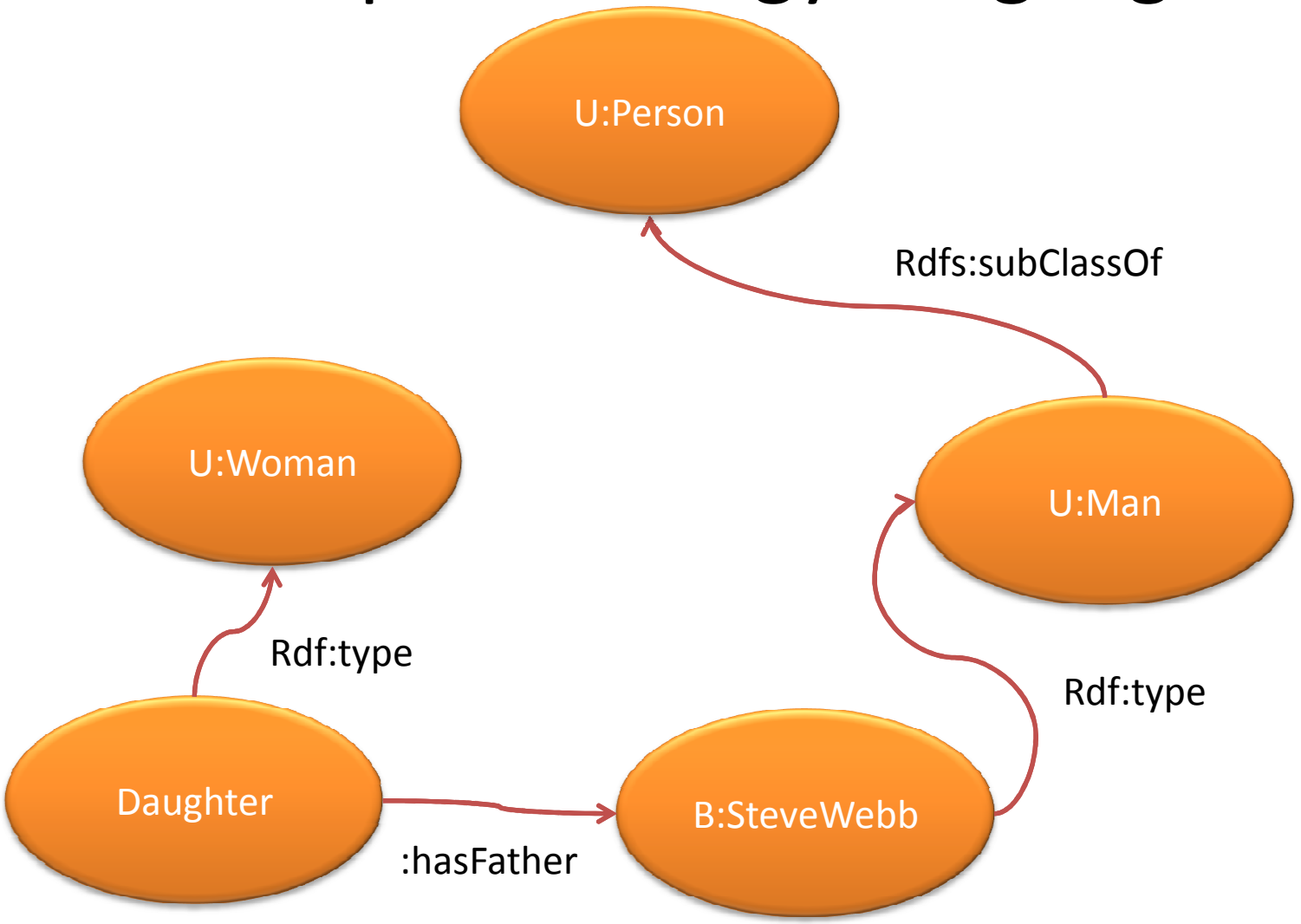


Note:
RDF Schema is itself expressed in RDF

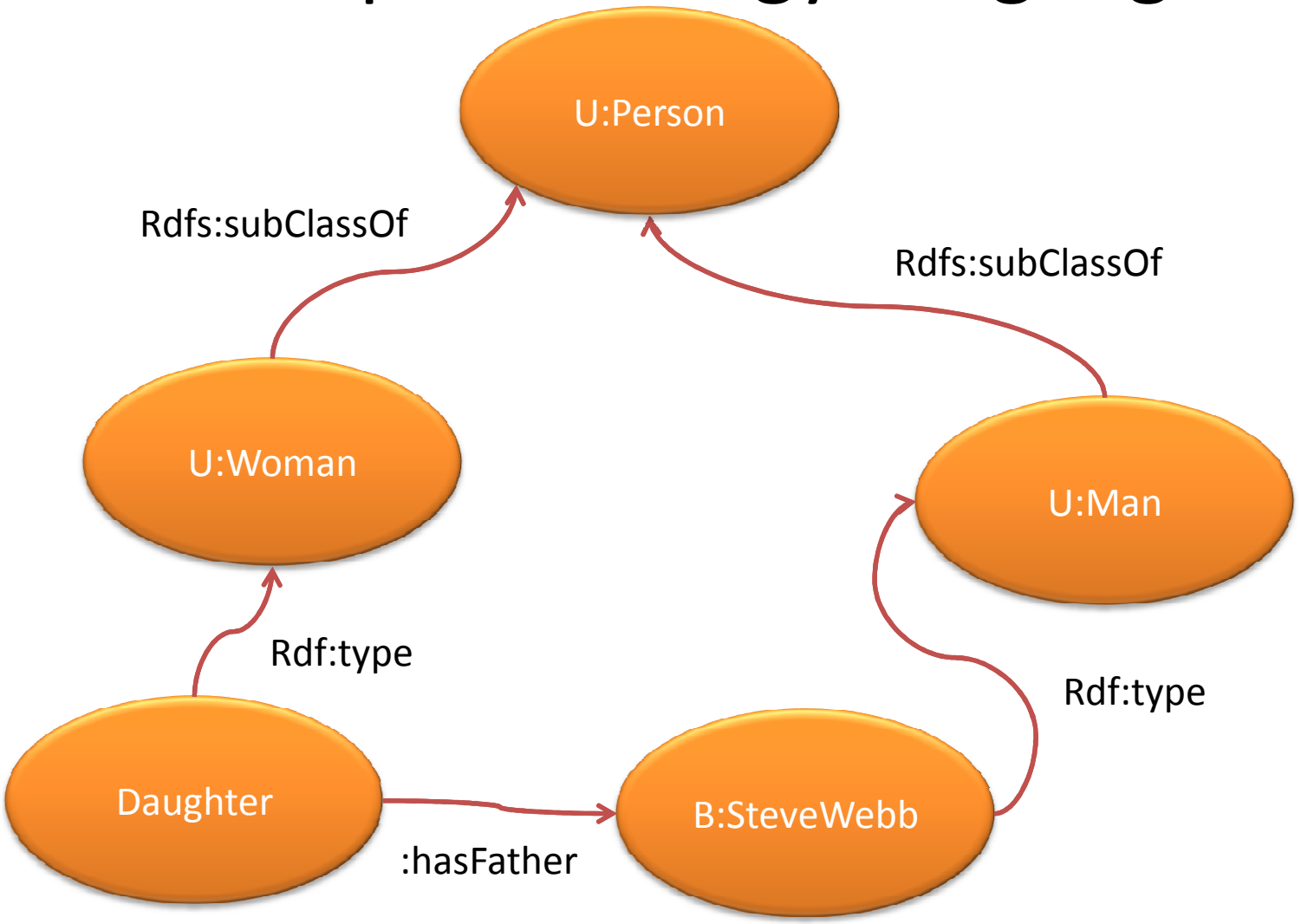
RDF Schema
A Simple Ontology Language



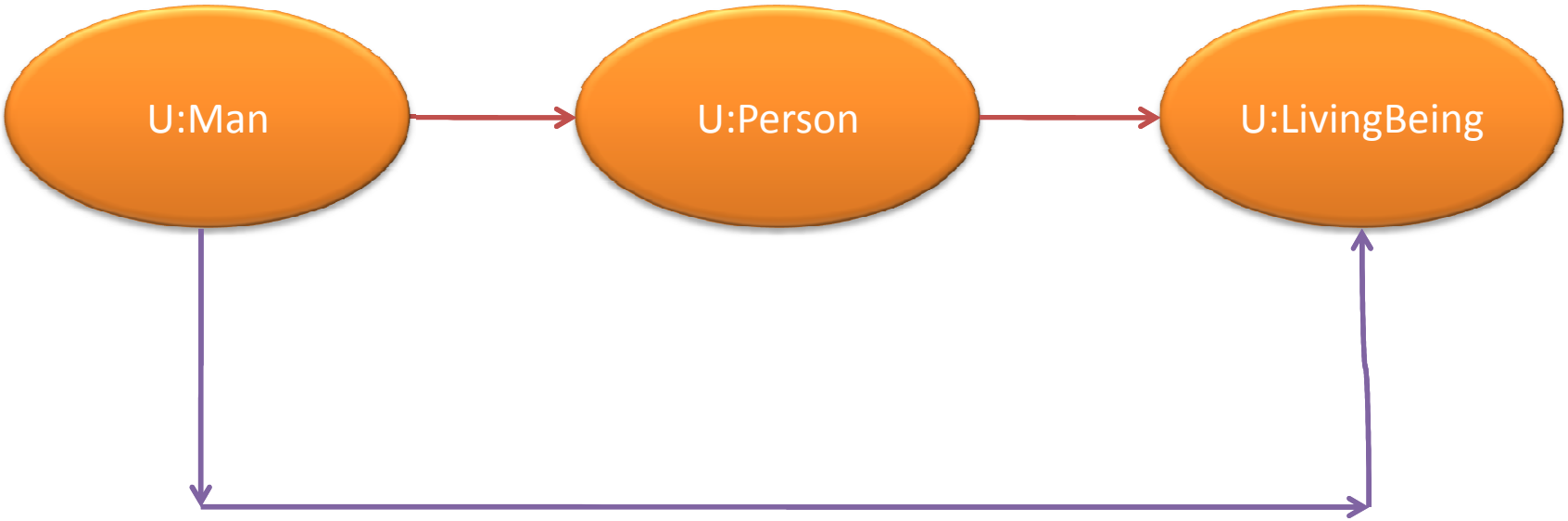
RDF Schema
A Simple Ontology Language



RDF Schema
A Simple Ontology Language



RDF Schema Inference
Subclass Inference



RDF Schema
Domain, Range, subProperty

- Range: defines the type of the value of a property – can be a datatype or a class
- Domain: defines the type of the thing at the blunt end of the arrow
- subPropertyOf: hasFather is a subProperty of hasParent:
  - X :hasFather Y => X hasParent Y
- hasFather and hasParent have different ranges

OWL: Web Ontology Langauge

- RDFS is expressively weak
  - No negation – no contradiction
- OWL is a more powerful language
  - Class expressions – e.g. Union, intersection, disjoint
  - Property types – inverse, transitive, functional, ...
  - ...

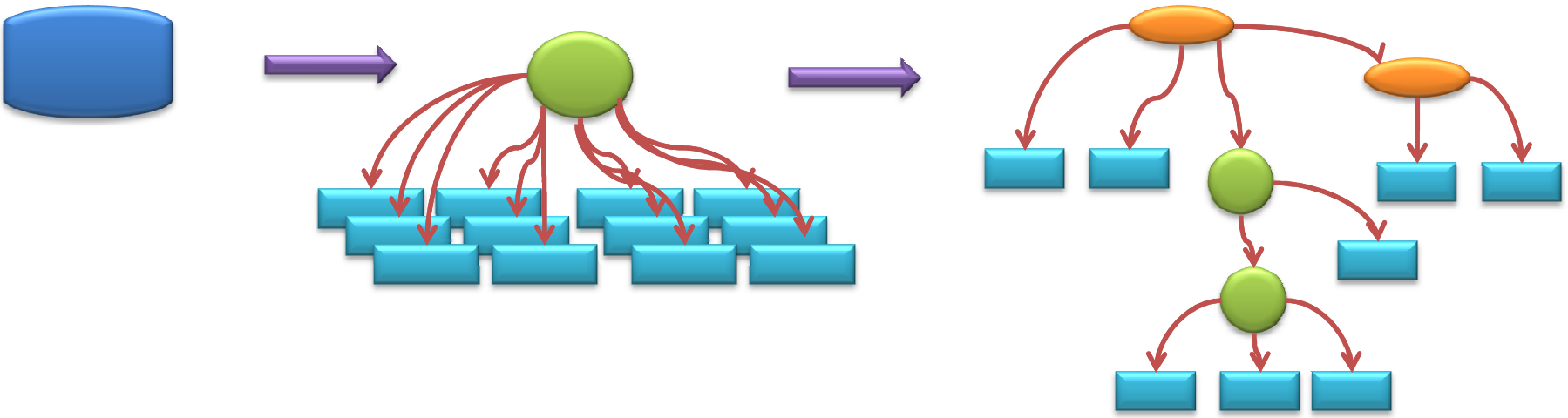
A Worked Example
Publish the EduBase Dataset LOD Style

- Basic reference data about schools in the UK
- Website http://www.edubase.gov.uk/home.xhtml
- CSV File
  - 218 columns
  - 66k rows – 1 per school
- Looks a bit like:

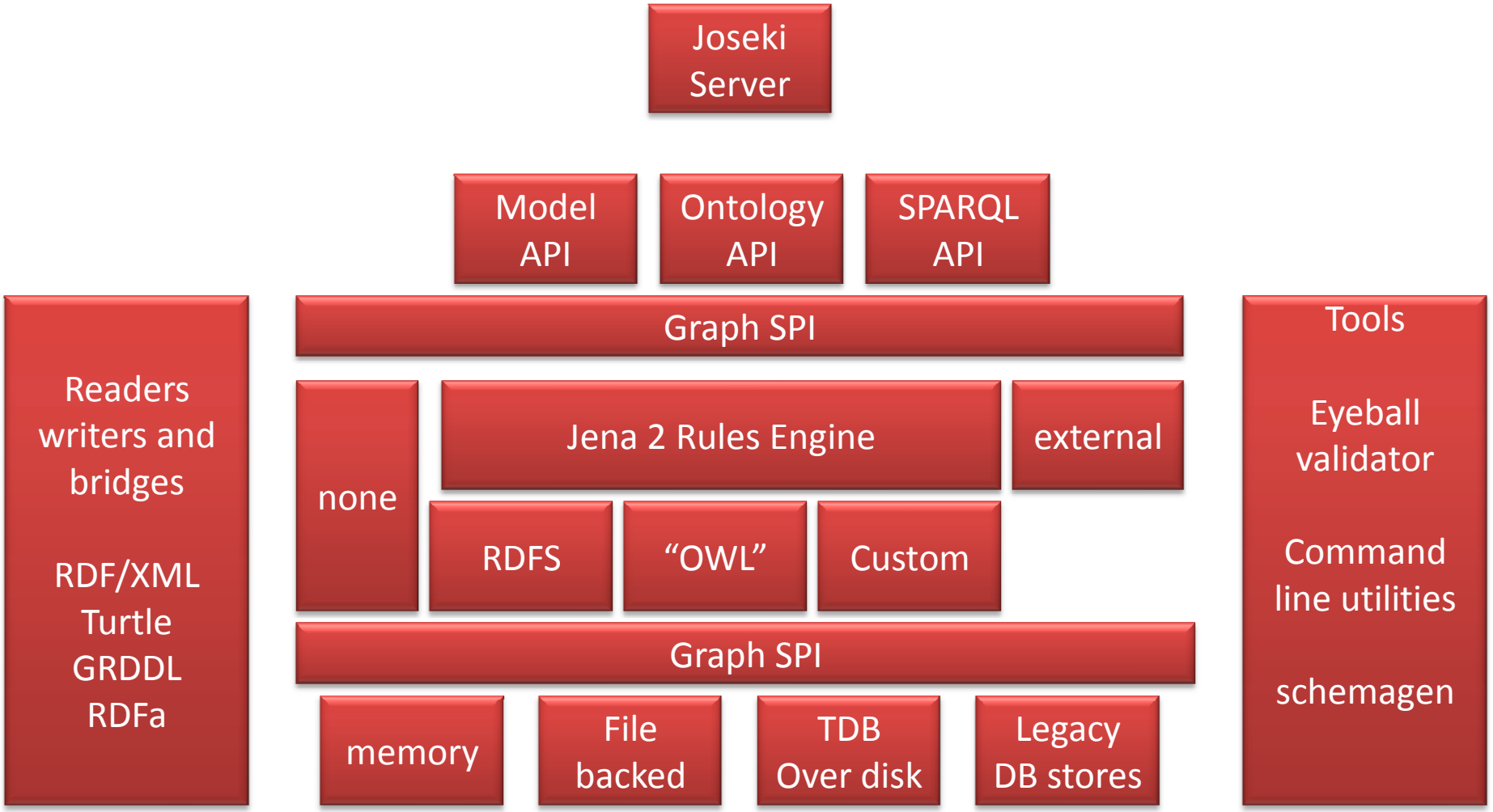
URN	LA code	LA	Status	Name	Type	...
100000	201	City of London	Open	School name	Voluntary Aided	...

Translation process

- Could operate in text mode with perl, awk, sed whatever to translate from CSV to an RDF concrete syntax such as RDF/XML or TURTLE.
- Also need to produce an ontology
- - use RDF tools



Jena Library Overview



Graph SPI

- Node s = Node.createResource("http://...");
- Node p = Node.createResource("http://...#label");
- Node o = Node.createLiteral("10", http://...#int);
- Triple t = new Triple(s,p,o);
- Graph g = new Graph();
- g.add(t); // or g.add(s,p,o);
- Iterator<Triple> iter = g.find(null, null, null);

Model API
Convenience API after JDom

- Model m = ModelFactory.createDefaultModel();
- m.createResource()
- .addProperty(SCHOOL.numPupils, 100)
- .addProperty(RDFS.label(“Marlwood School);
- m.list(null, null, null);
- r.getProperty(RDFS.label).getString();

Input File Analysis

- Column headings massaged to produce property class names etc
- Automatic analysis identifies probable patterns
  - String valued properties
  - Datatype valued properties
  - Controlled vocabulary terms
  - Types/boolean valued properties
- Then manually tweak – to produce an ontology

Semi-automatic production of the ontology

```
:establishmentName a owl:DatatypeProperty;
        rdfs:label 'establishment name';
        rdfs:domain :School;
        rdfs:range  xsd:string;
        meta:columnName 'EstablishmentName';
        meta:columnCategory 'SIMPLE_STRING'.
```

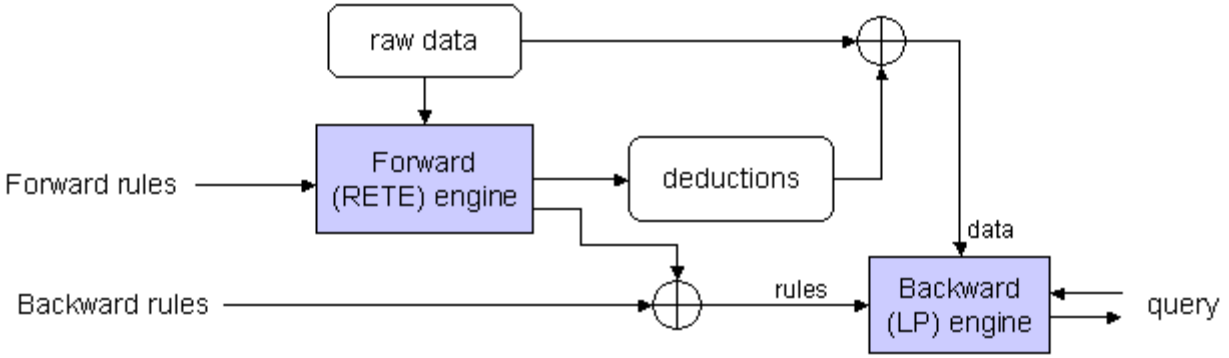
A Class

- :TypeOfEstablishment_LA_Nursery_School
- a owl:Class;
- rdfs:subClassOf :School;
- rdfs:label 'LA Nursery School';
- rdfs:comment 'A class used to indicate a LA Nursery School type of establishment';
- meta:columnName 'TypeOfEstablishment (name)'.

Pseudo Boolean

- :officialSixthForm a owl:DatatypeProperty;
- rdfs:label 'official sixth form';
- rdfs:domain :School;
- rdfs:range xsd:boolean;
- meta:columnName 'OfficialSixthForm (name)';
- meta:columnCategory 'PSEUDO_BOOLEAN';
- meta:descriptionIfTrue 'Has a sixth form';
- meta:descriptionIfFalse 'Does not have a sixth form'.

The Jena 2 Rules Engine



- Hybrid Forward and Backward Chaining Engine
- Rules can fire both ways
- Forward engine can add rules for the backward engine
- Can update – add new triples – get new deductions

Forward Chaining Rule

- (cs1 cp1 co1),
- (cs2 cp2 co2)
- ->
- (ds1 dp1 do2),
- (ds2 dp2 do2)

- Can have functors in the object position
  - (ds1 dp1 functor(cp1 cp2 co1 co2))
- Small extensible set of built in functions
  - makeTemp(?temp), makeList etc

Example Translation Rule For an ontology pattern

- [datatypeRule:
- (?rawSchool cont:raw2new ?school),
- (?rawSchool ?p ?o),
- (cont:columnRec ?p ?columnName),
- (?np meta:columnName ?columnName),
- (?np meta:columnCategory “SIMPLE_DATATYPE),
- (?np rdfs:range ?dataType),
- makeTypedLiteral(?o, ?dataType, ?dtValue),
- ->
- (?school ?np ?dtp)
-]

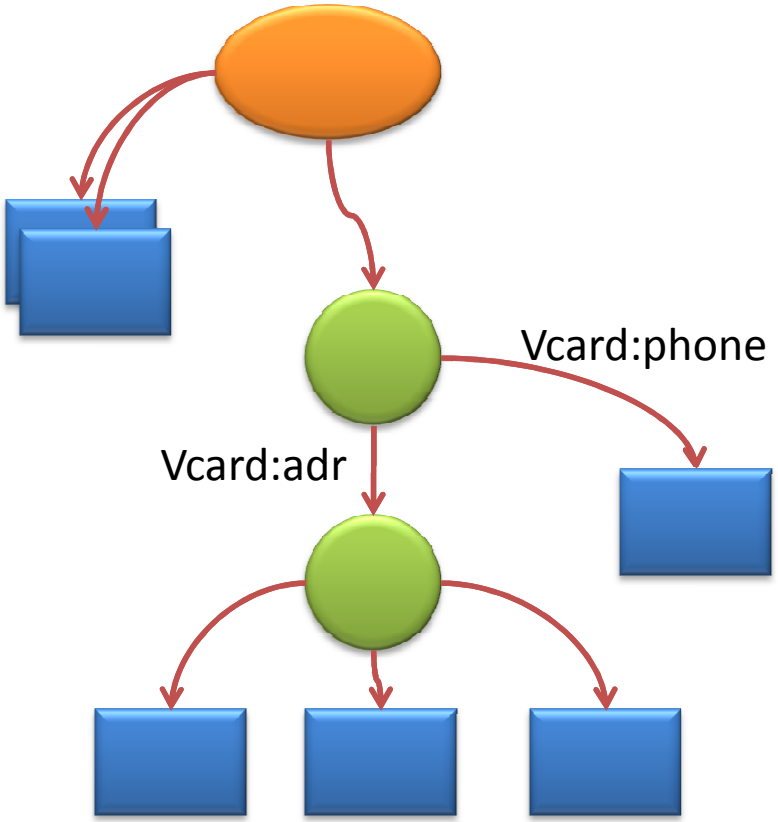
Compile Phase creates Control Triples

- [datatypeCompileRule:
- (?rawSchool ?p ?columnName),
- (?np meta:columnName ?columnName),
- (?np meta:columnCategory “SIMPLE_DATATYPE),
- (?np rdfs:range ?dataType),
- ->
- (?p cont:datatype ?dataType),
- (?p cont:newProp ?np)
-]

- [dataTypeTransformRule:
- (?rawSchool ?p ?o),
- (?p cont:datatype ?dataType),
- (?p cont:newProp ?np)
- makeTypedLiteral(?o, ?dataType, ?dtValue),
- ->
- (?school ?np ?dtp)
-]

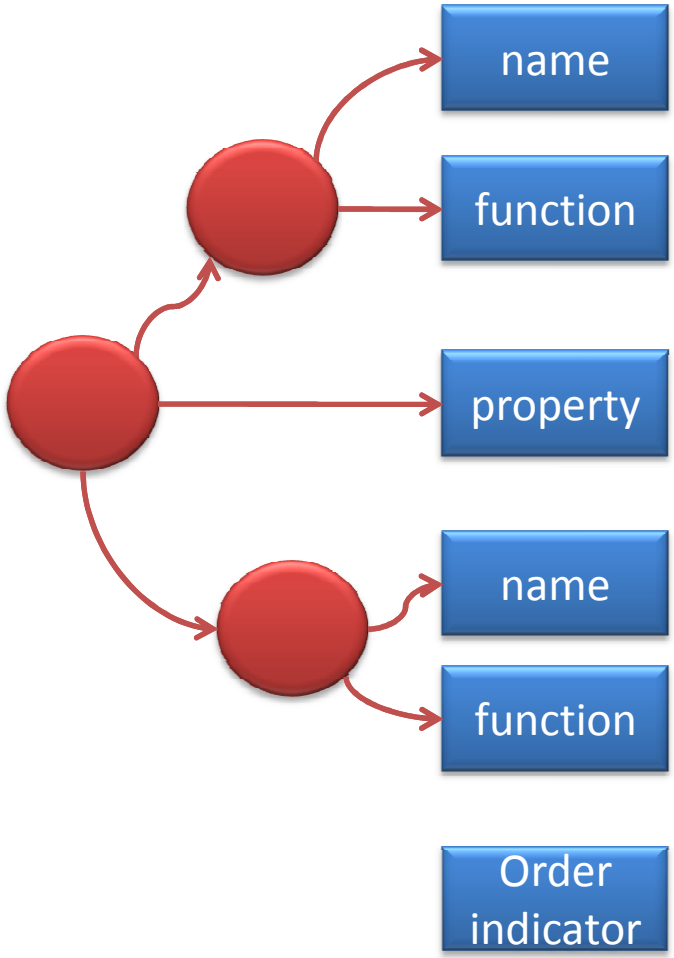
Creating Graphs

- Not all properties are hung of the root node
- Create substructures
- Name the nodes
- Don’t build structure for which there is no data



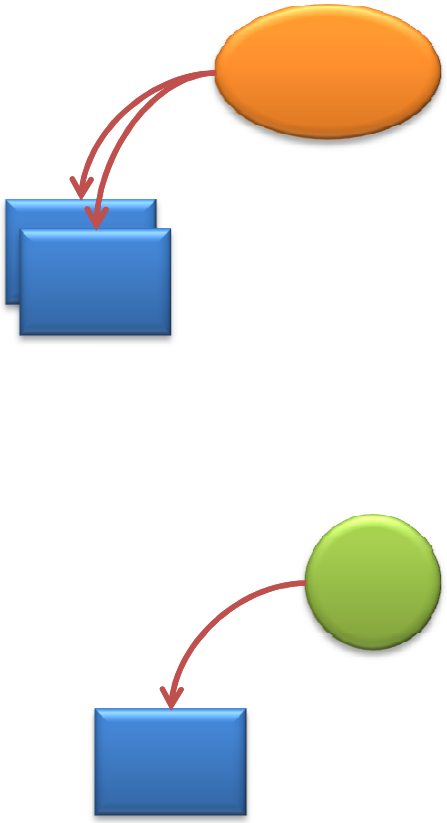
Translation Instruction

- Instruction structure
  - Subject predicate object
  - Subject and object have:
    - A name (optional)
    - A function (optional)
- Instruction Execution
  - Evaluate the object
  - if there is a subject and property
    - Evaluate the subject
    - Add the triple (s p o)



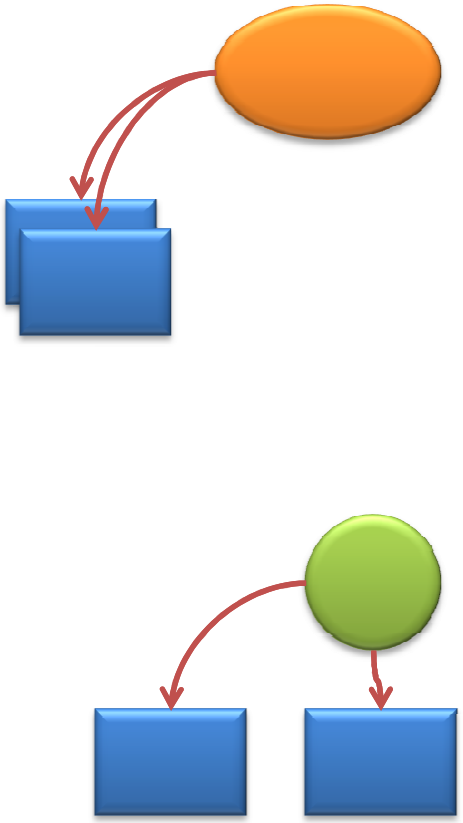
Build the graph structure bottom up

- Instruction
  - Subject name “addr”
  - Subject Fn: make bNode
  - Property: prop
  - Object name: none
  - Object Fn: get value



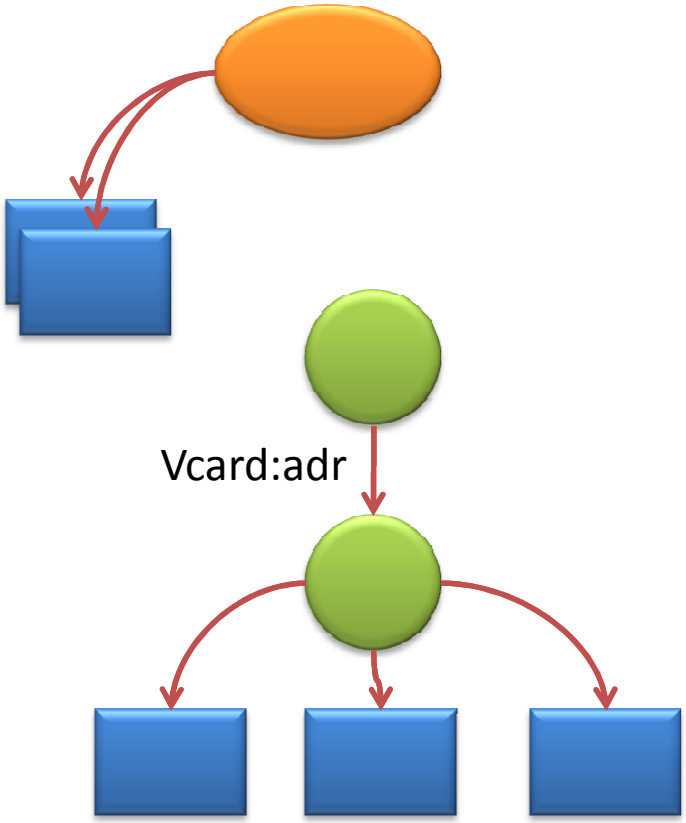
Build the graph structure bottom up

- Instrution
  - Subject name “addr”
  - Subject Fn: make bNode
  - Property: prop
  - Object name: none
  - Object Fn: get value



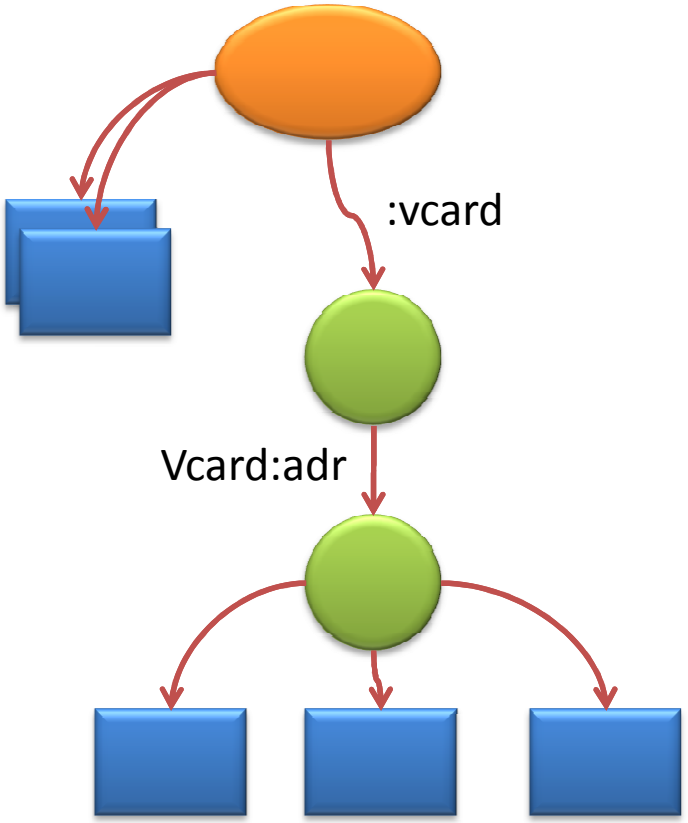
Build the graph structure bottom up

- Instrution
  - Subject name “vcard”
  - Subject Fn: make bNode
  - Property: vcard:adr
  - Object name: addr
  - Object Fn: none

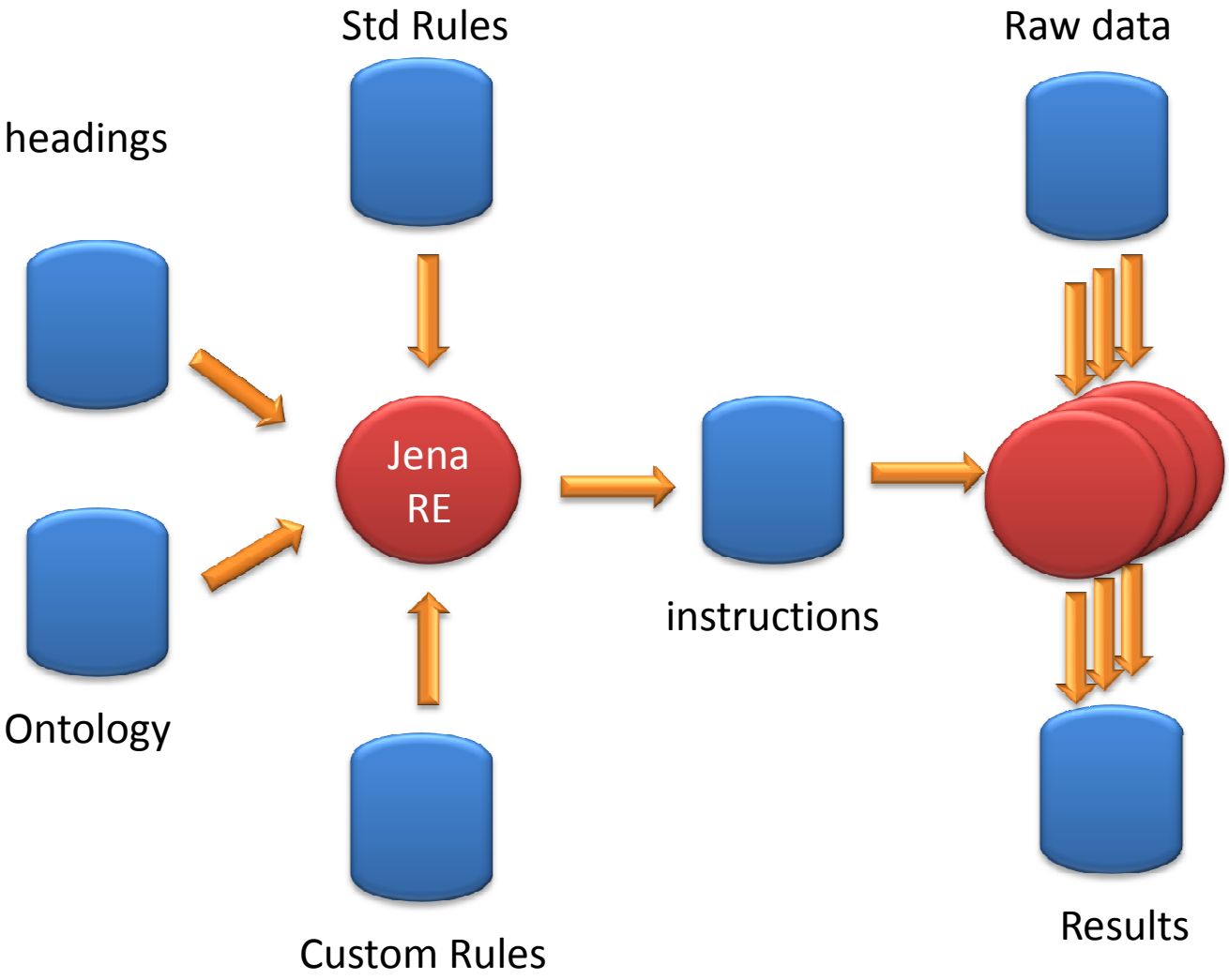


Build the graph structure bottom up

- Instrution
  - Subject name “root”
  - Subject Fn: ...
  - Property: :vcard
  - Object name: vcard
  - Object Fn: none



Interpreter



How are we doing on the principles

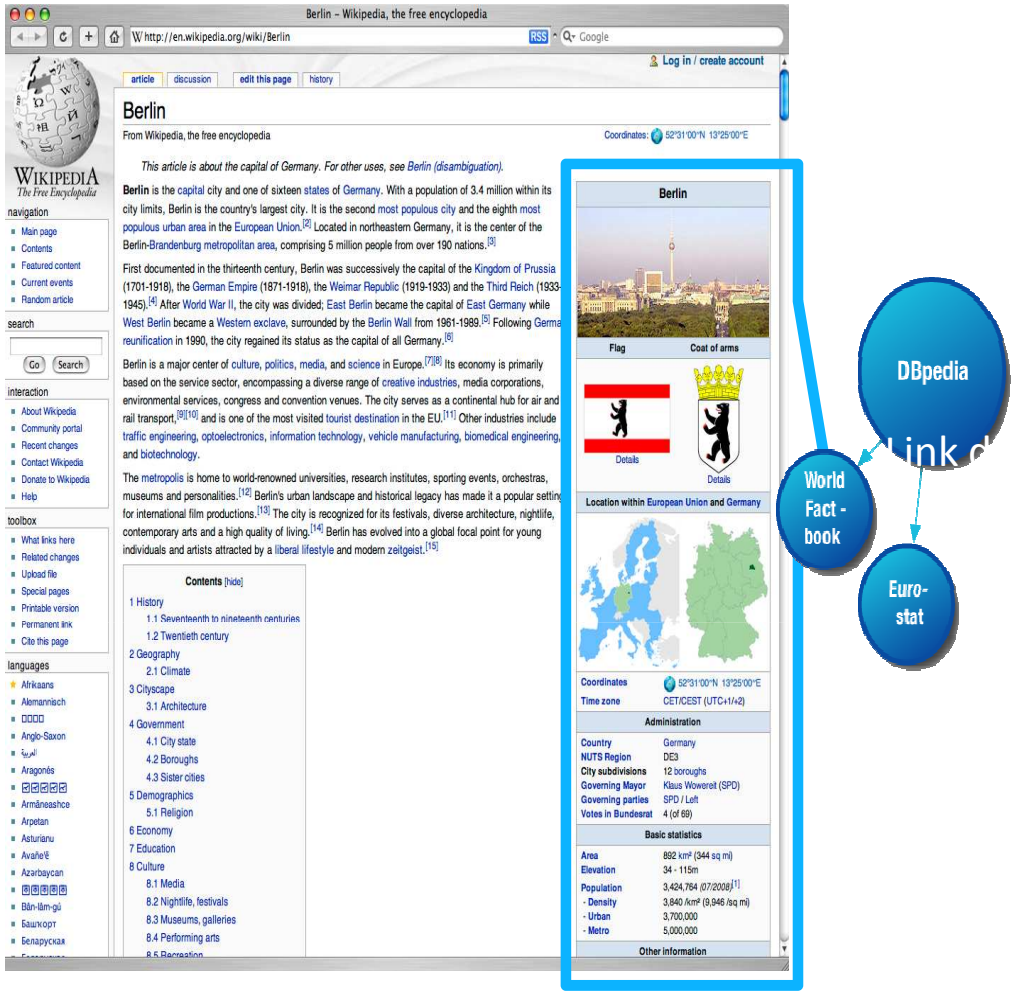
- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.

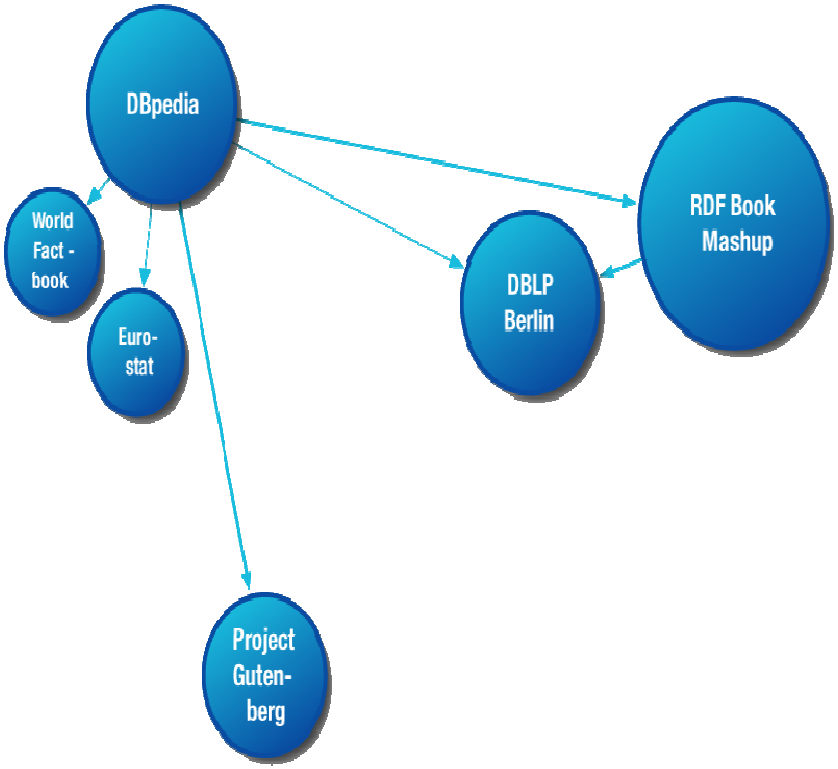
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- Include links to other URIs. so that they can discover more things.

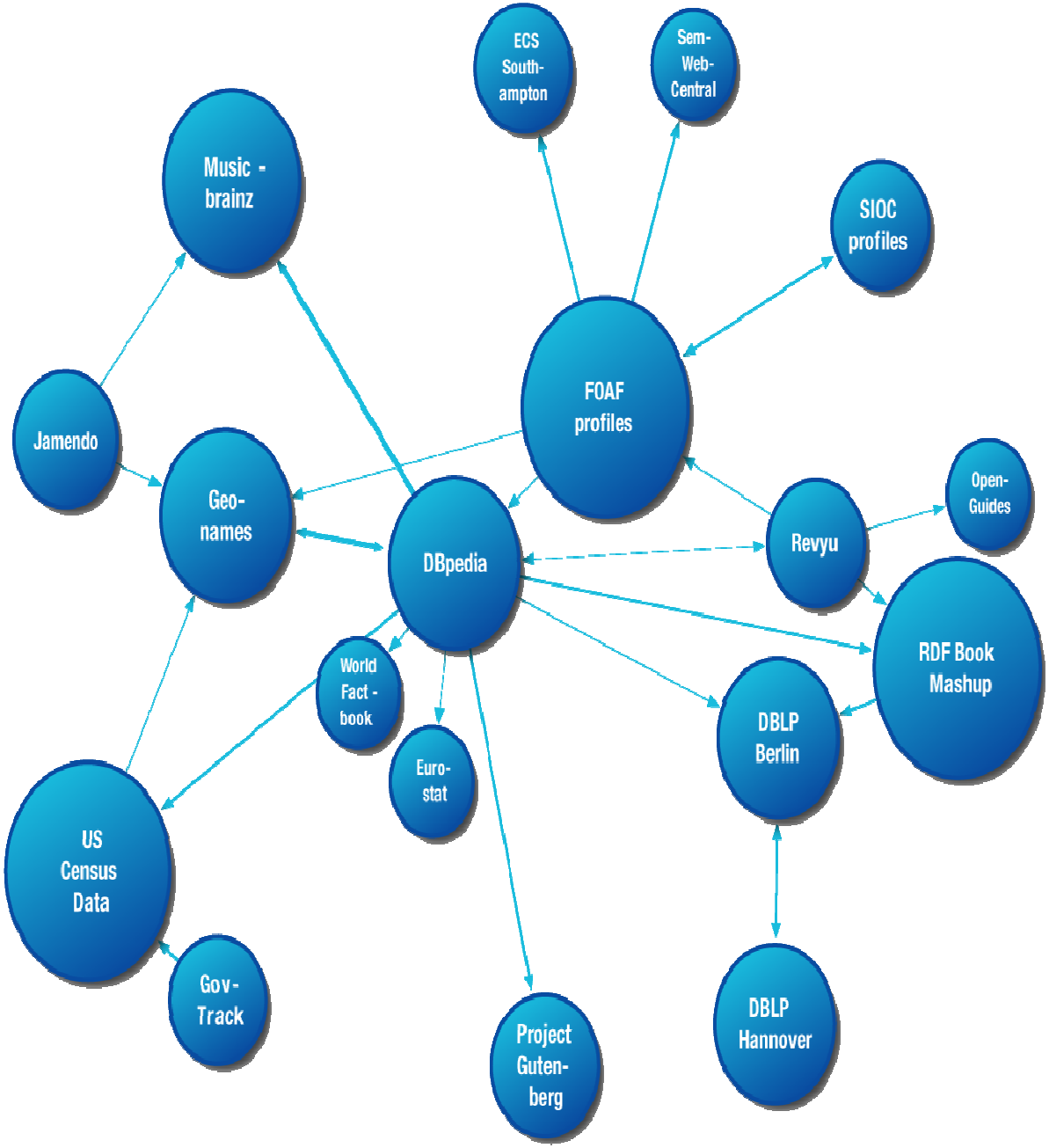
DBPedia

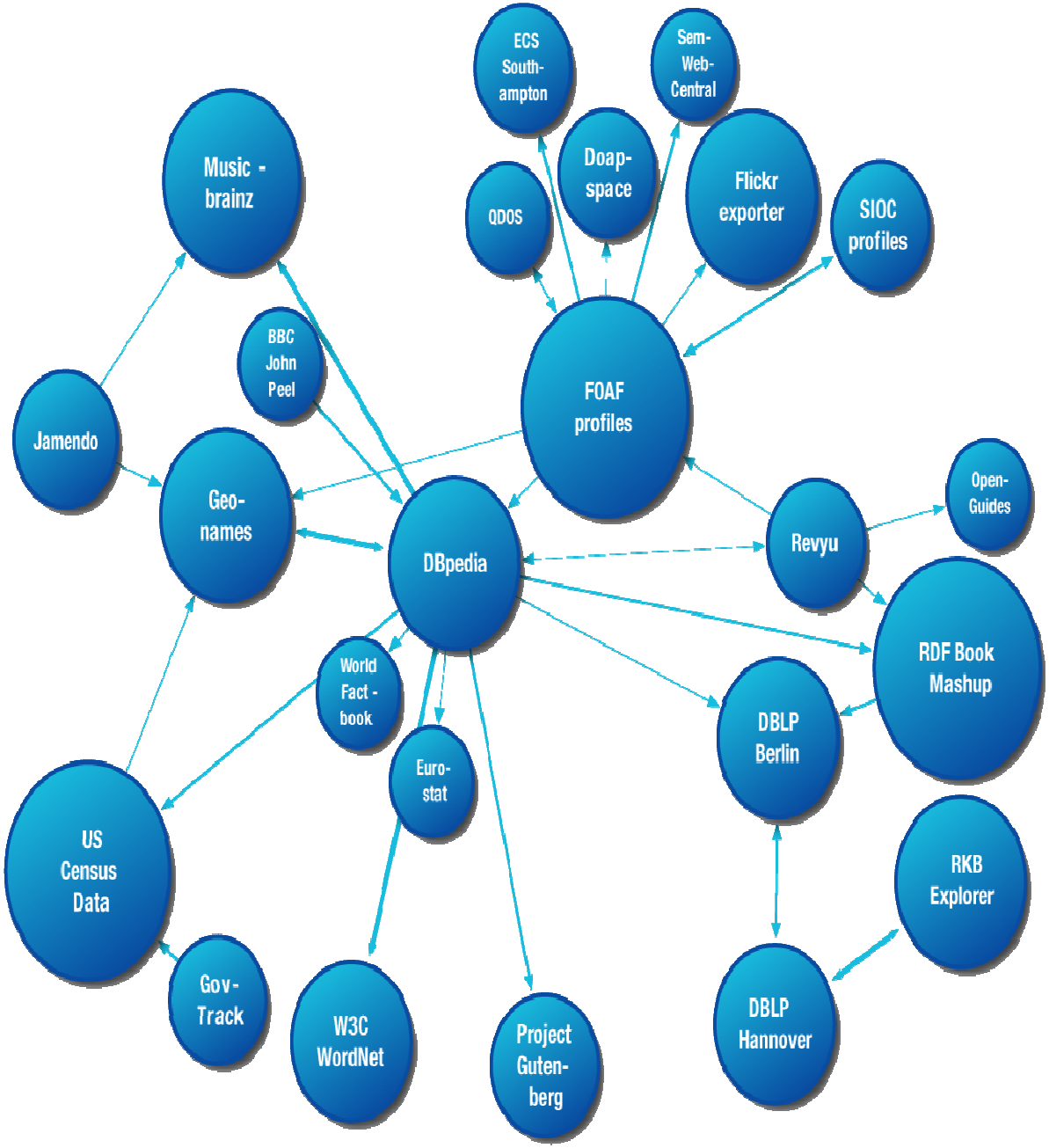
- We all know about Wikipedia
- DBPedia project
  - Extracts information from wikipedia
  - Publishes it Linked Open Data style

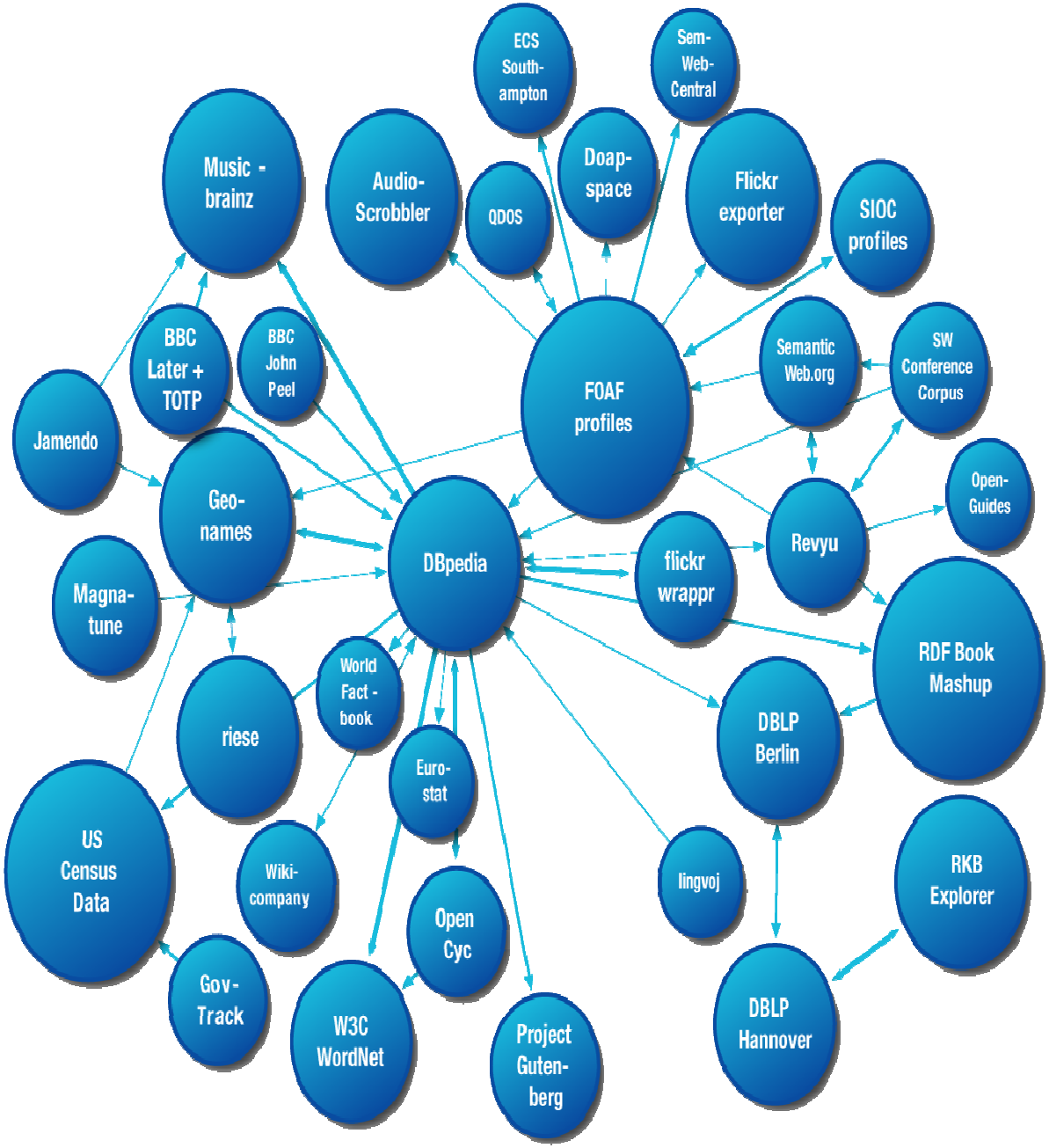
```
<http://dbpedia.org/resource/Northavon_%28UK_Parliament_constituency%29>
    rdfs:label "Northavon" ;
    dbpprop:mp <http://dbpedia.org/resource/Steve_Webb> .

<http://dbpedia.org/resource/Steve_Webb>
    rdfs:label "Steven John Webb" ;
    dbpprop:name "Steven John Webb" .
```

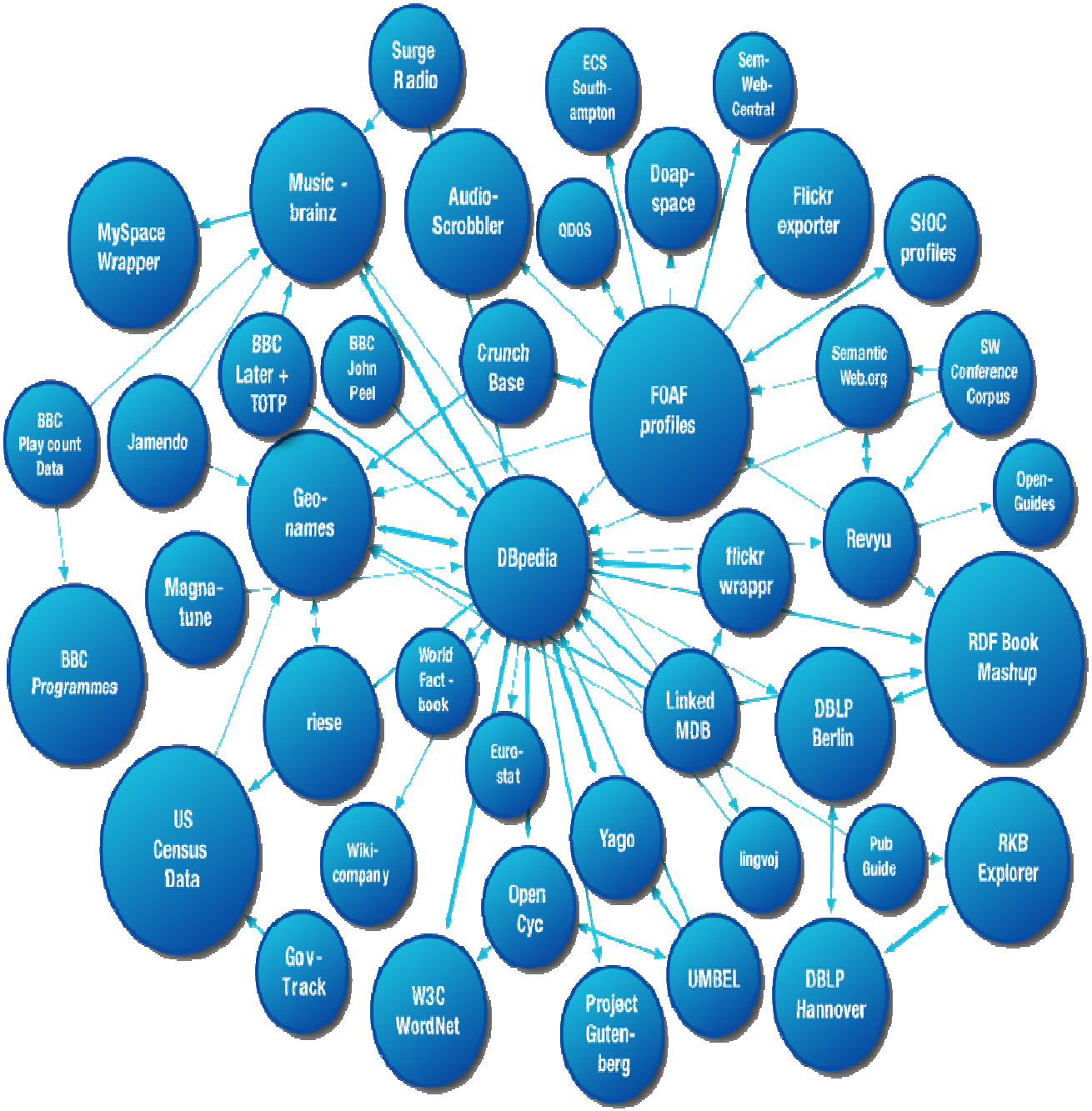

Berlin – Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Berlin
Google
Log in / create account
article
discussion
edit this page
history
Berlin
From Wikipedia, the free encyclopedia
Coordinates: 52°31′00″N 13°25′00″E
This article is about the capital of Germany. For other uses, see Berlin (disambiguation).
Berlin is the capital city and one of sixteen states of Germany. With a population of 3.4 million within its city limits, Berlin is the country's largest city. It is the second most populous city and the eighth most populous urban area in the European Union.[2] Located in northeastern Germany, it is the center of the Berlin-Brandenburg metropolitan area, comprising 5 million people from over 190 nations.[3]
First documented in the thirteenth century, Berlin was successively the capital of the Kingdom of Prussia (1701-1918), the German Empire (1871-1918), the Weimar Republic (1919-1933) and the Third Reich (1933-1945).[4] After World War II, the city was divided; East Berlin became the capital of East Germany while West Berlin became a Western exclave, surrounded by the Berlin Wall from 1961-1989.[5] Following Germa reunification in 1990, the city regained its status as the capital of all Germany.[6]
Berlin is a major center of culture, politics, media, and science in Europe.[7][8] Its economy is primarily based on the service sector, encompassing a diverse range of creative industries, media corporations, environmental services, congress and convention venues. The city serves as a continental hub for air and rail transport,[9][10] and is one of the most visited tourist destination in the EU.[11] Other industries include traffic engineering, optoelectronics, information technology, vehicle manufacturing, biomedical engineering, and biotechnology.
The metropolis is home to world-renowned universities, research institutes, sporting events, orchestras, museums and personalities.[12] Berlin's urban landscape and historical legacy has made it a popular settin for international film productions.[13] The city is recognized for its festivals, diverse architecture, nightlife, contemporary arts and a high quality of living.[14] Berlin has evolved into a global focal point for young individuals and artists attracted by a liberal lifestyle and modern zeitgeist.[15]
Contents [hide]
1 History
1.1 Seventeenth to nineteenth centuries
1.2 Twentieth century
2 Geography
2.1 Climate
3 Cityscape
3.1 Architecture
4 Government
4.1 City state
4.2 Boroughs
4.3 Sister cities
5 Demographics
5.1 Religion
6 Economy
7 Education
8 Culture
8.1 Media
8.2 Nightlife, festivals
8.3 Museums, galleries
8.4 Performing arts
8.5 Recreation
navigation
Main page
Contents
Featured content
Current events
Random article
search
Go
Search
interaction
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Donate to Wikipedia
Help
toolbox
What links here
Related changes
Upload file
Special pages
Printable version
Permanent link
Cite this page
languages
Afrikaans
Alemannisch
Anglo-Saxon
Aragonés
Armâneashce
Arpetan
Asturianu
Azərbaycan
Bân-lâm-gú
Башкорт
Беларуская
Berlin
Flag
Coat of arms
Details
Details
Location within European Union and Germany
Coordinates 52°31′00″N 13°25′00″E
Time zone CET/CEST (UTC+1/+2)
Administration
Country Germany
NUTS Region DE3
City subdivisions 12 boroughs
Governing Mayor Klaus Wowereit (SPD)
Governing parties SPD / Left
Votes in Bundesrat 4 (of 69)
Basic statistics
Area 892 km² (344 sq mi)
Elevation 34 - 115m
Population 3,424,764 (07/2008)[1]
- Density 3,840 /km² (9,946 /sq mi)
- Urban 3,700,000
- Metro 5,000,000
Other information
DBpedia
World Fact - book
Euro- stat


DBpedia
World Fact - book
Euro- stat
DBLP Berlin
RDF Book Mashup
Project Guten- berg


ECS South-ampton
Sem-Web-Central
Music - brainz
SIOC profiles
FOAF profiles
Jamendo
Geo-names
DBpedia
Revyu
Open-Guides
RDF Book Mashup
World Fact - book
Euro-stat
DBLP Berlin
US Census Data
Gov-Track
Project Guten-berg
DBLP Hannover


ECS South-ampton
Sem-Web-Central
Music - brainz
Doap-space
Flickr exporter
QDOS
SIOC profiles
BBC John Peel
FOAF profiles
Jamendo
Geo-names
DBpedia
Revyu
Open-Guides
RDF Book Mashup
World Fact - book
DBLP Berlin
Euro-stat
US Census Data
RKB Explorer
Gov-Track
W3C WordNet
Project Guten-berg
DBLP Hannover


ECS South-ampton
Sem-Web-Central
Music - brainz
Audio-Scrobbler
QDOS
Doap-space
Flickr exporter
SIOC profiles
BBC Later + TOTP
BBC John Peel
FOAF profiles
Semantic Web.org
SW Conference Corpus
Jamendo
Geo-names
DBpedia
Open-Guides
Revyu
flickr wrappr
Magna-tune
RDF Book Mashup
World Fact - book
riese
DBLP Berlin
Euro-stat
US Census Data
Wiki-company
lingvoj
RKB Explorer
Open Cyc
Gov-Track
W3C WordNet
Project Guten-berg
DBLP Hannover


Surge Radio
ECS South-ampton
Sem-Web-Central
Music - brainz
Audio-Scrobbler
Doap-space
Flickr exporter
SIOC profiles
MySpace Wrapper
QDOS
BBC Later + TOTP
BBC John Peel
Crunch Base
FOAF profiles
Semantic Web.org
SW Conference Corpus
BBC Play count Data
Jamendo
Geo-names
DBpedia
Revyu
Open-Guides
flickr wrappr
Magna-tune
RDF Book Mashup
BBC Programmes
World Fact-book
Linked MDB
DBLP Berlin
riese
Euro-stat
US Census Data
Yago
Pub Guide
RKB Explorer
Wiki-company
lingvoj
Open Cyc
Gov-Track
UMBEL
DBLP Hannover
W3C WordNet
Project Guten-berg

Linking Data

- Schools data specifies the parliamentary constituency its in
- Link to:
  - OS administrative geography data
  - DBPedia information about the constituency
- The schools data has a text field with the name of the constituency – not a URI.
- So get all the DbPedia constituencies and do a text match on the names

The SPARQL Query We'd like to do

- @PREFIX dbpedia: < ... >
- @PREFIX skos: <...>

- CONSTRUCT {
- ?constituency rdfs:label ?label .
- }

- WHERE {
- ?constituency rdf:type dbpedia:UkParliamentaryConstituency .
- ?constituency rdfs:label ?label .
- FILTER (lang(?label) = "en") }
- }

- This DOESN'T work!

Jena Local and remote SPARQL Query

```
CONSTRUCT {
   ?constituency rdfs:label ?label .
}
WHERE  {
 SELECT ?concept {
     ?concept (skos:broader)+
         category:United_Kingdom_Parliamentary_constituencies .
     OPTIONAL {(?narrower skos:broader ?concept)}
     FILTER(!BOUND(?narrower))
 }
 SERVICE <http://dbpedia.org/sparql> {
   ?constituency skos:subject ?concept .
   ?constituency rdfs:label ?label  .
   FILTER (lang(?label) = "en“)
 }
}
```

Name Matchers built using SecondString

- configMatcher(“const”, "logNoMatch", "false"),
- configMatcher(“const”, "preprocessor", "remove", "
  (UK Parliament constituency)"),
- configMatcher(“const”, "preprocessor", "dropChars",
  ".,-\""),
- configMatcher(“const”, "preprocessor", "tolowercase"),
- configMatcher(“const”, “algorithm”, “bagOfWords”),
- configMatcher(“const”, "load" ,
  <file:data/dbPediaConstituencies.n3>, rdfs:label, "N3")

Custom Rule to create link

- [linkConstituencyToDbPedia:
- (?p cont:newProp school:parliamentaryConstituency),
- (school:parliamentaryConstituency meta:columnName ?nodeName),
- makeTemp(?node),
- makeList(?p, cont:ParliamentaryConstituency, ?args)
- ->
- (?node cont:linkProp owl:sameAs),
- (?node cont:nodeName ?nodeName),
- (?node cont:op config:lookup),
- (?node cont:args ?args)
-]

Modeling needs care …

- OS have data describing the geo properties of various adminstrative regions
  - <rdf:Description rdf:about="http://data.ordnancesurvey.co.uk/id/7000000000024920">
  - <rdf:type rdf:resource="http://www.ordnancesurvey.co.uk/ontology/admingeo/WestminsterConstituency"/>
  - <rdfs:label>Northavon</rdfs:label>
  - <foaf:name>Northavon</foaf:name>
  - <admingeo:hasUnitID>24920</admingeo:hasUnitID>
  - <admingeo:hasAreaCode>WMC</admingeo:hasAreaCode>
  - <admingeo:hasArea>47032.424</admingeo:hasArea>
  - </rdf:Description>
- But beware – the OS constituency is a geographic area – not an administrative body
- Don’t use owl:same has – use :hasRegion or some such

How are we doing on the principles

Use URIs as names for things

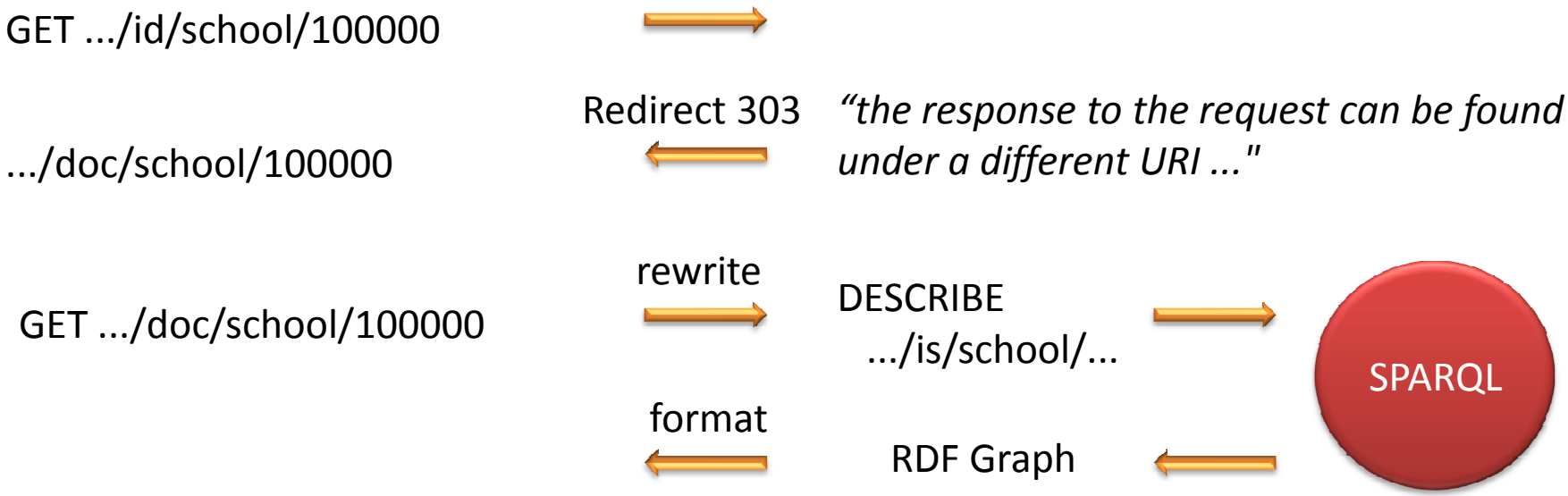
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)

Include links to other URIs. so that they can discover more things.

Publishing the data

- LoD says:
  - Name things – like schools – with URIs
  - Use http URIs to make them de-referencable
- Big web architecture debate about whether http URIs can name things or only documents
- http://www.ihmc.us/users/phayes/PatHayesAbout.html
  - States that this URI denotes Pat Hayes the man – not the document you get back if you do a GET on it
  - But surely it must name the document
  - And since men and documents are disjoint ...

A Simple Publishing Scheme



How are we doing on the principles

Use URIs as names for things

- Use HTTP URIs so that people can look up those names.

- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)

Include links to other URIs. so that they can discover more things.

Now that we have linked data ...

```
SELECT ?mpName (sum(?teenMotherPlaces) as ?total)
WHERE {
 SERVICE <http://dbpedia.org/sparql> {
   ?cons dbpprop:mp ?mp .
   ?mp rdfs:label ?mpName .
 }
?dbpediaLinks rdfs:label “DBPedia Links” .
Graph ?dbpediaLinks {
    ?c2 owl:sameAs ?cons .
}
 ?school edu:establishmentName ?schoolName;
        edu:hasTeenageMothers "true"^^xsd:boolean;
        edu:teenageMotherPlaces ?teenMotherPlaces;
        edu:parliamentaryConstituency ?c2 .

} GROUP BY  ?mpName ORDER BY ?total
```

MP	#
Steve Webb	0
…	0
…	5
…	…

So what else could you do with the data

- Houses for sale/rent
  - web page can dynamically link to local schools
  - Show key information – performance data, truancy rates, specialisms etc

UK Governement appeal to try out new data site and exploit the data

- “From today we are inviting developers to show government how to get the future public data site right - how to find and use public sector information.”
- 1000+ datasets (not all/many in LoD form - YET)
- Including a SPARQL endpoint on the EduBase dataset
- http://blogs.cabinetoffice.gov.uk/digitalengagement/

Discussion