Detailed introduction into RDF and the Semantic Web

Search & Find Workshop
Ghent, Belgium, 2008-08-22

Ivan Herman, W3C

Towards a Semantic Web

- Tasks often require to combine data on the Web:
  - hotel and travel information may come from different sites
  - searches in different digital libraries
  - etc.
- Humans combine these information easily even if
  - different terminologies are used
  - the information is incomplete, or buried in images, videos, …

However…

- However: machines are ignorant!
  - partial information is unusable
  - difficult to make sense from, e.g., an image
  - drawing analogies automatically is difficult
  - difficult to combine information automatically
    - is `<foo:creator>` same as `<bar:author>`?
  - …

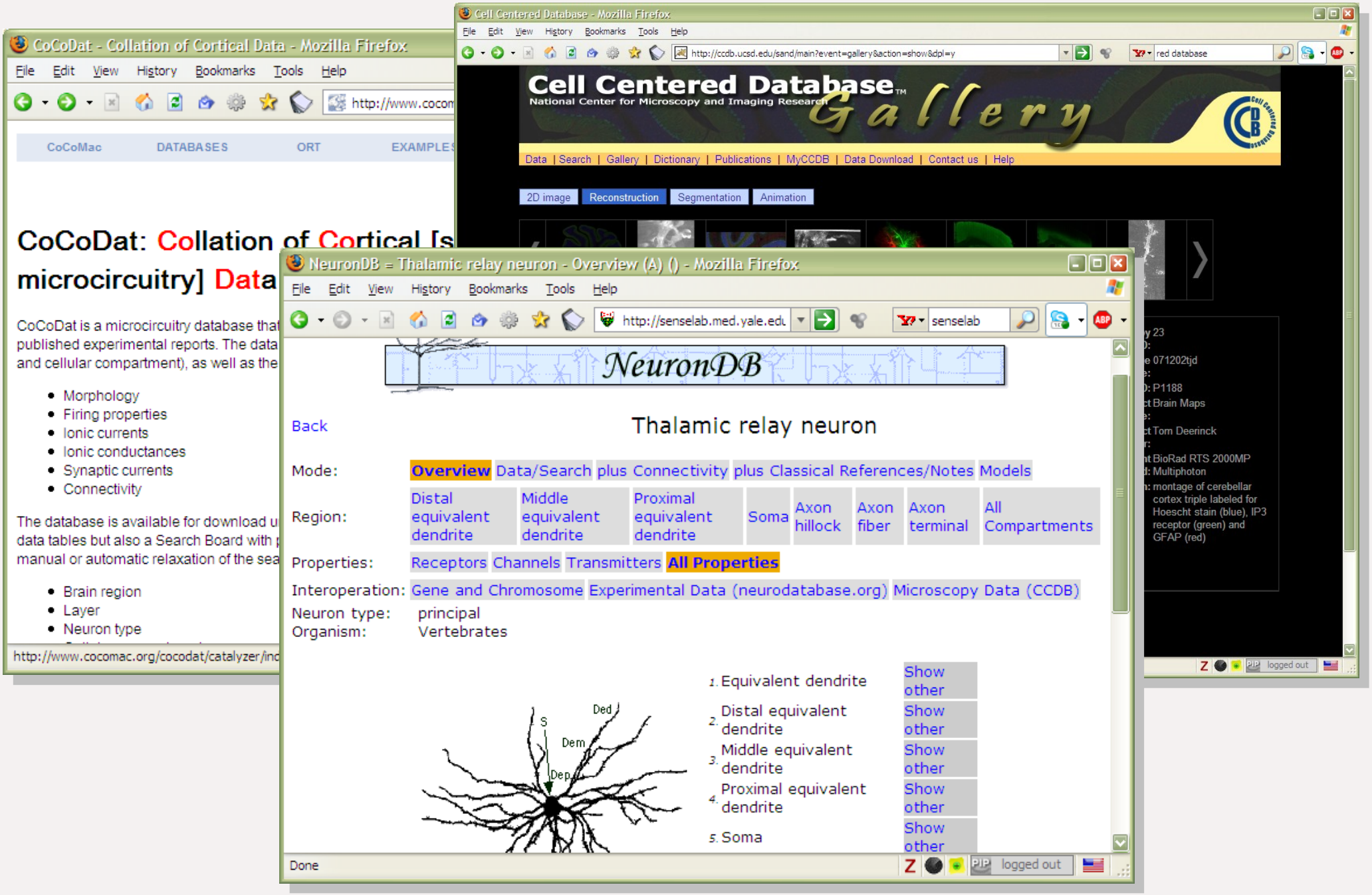
Example: automatic airline reservation

- Your automatic airline reservation
  - knows about your preferences
  - builds up knowledge base using your past
  - can combine the local knowledge with remote services:
    - airline preferences
    - dietary requirements
    - calendaring
    - etc
- It communicates with remote information (i.e., on the Web!)
  - (M. Dertouzos: The Unfinished Revolution)

Example: data(base) integration

- Databases are very different in structure, in content
- Lots of applications require managing several databases
  - after company mergers
  - combination of administrative data for e-Government
  - biochemical, genetic, pharmaceutical research
  - combination of online library data
  - etc.
- Most of these data are accessible from the Web (though not necessarily public yet)

And the problem *is* real…



Example: social networks

- Social sites are everywhere these days (LinkedIn, Facebook, Dopplr, Digg, Plexo, Zyb, …)
- Data is not interchangeable: how many times did you have to add your contacts?
- Applications should be able to get to those data via standard means
  - there are, of course, privacy issues…

Example: digital libraries

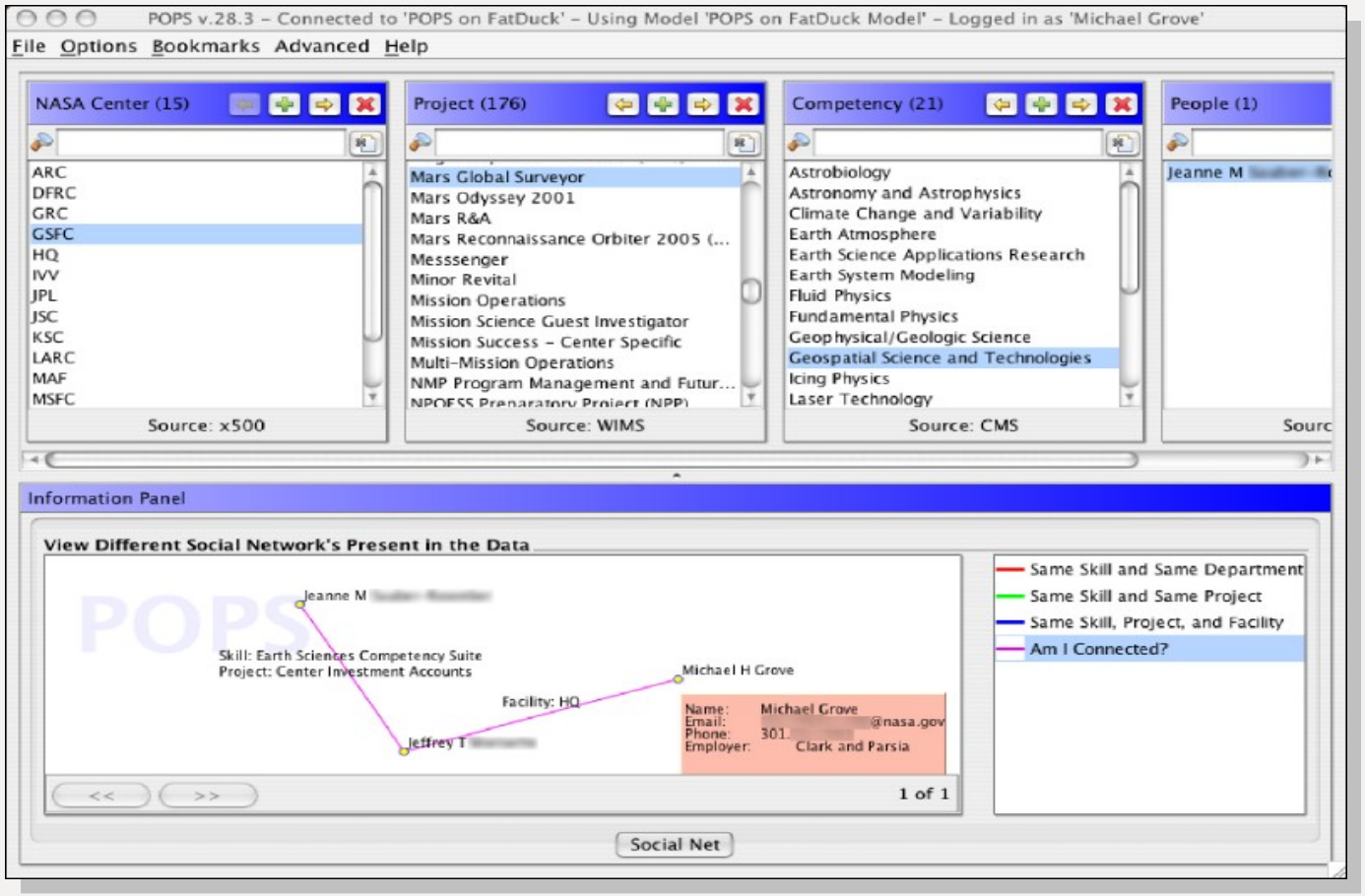
- Sort of catalogues on the Web
  - librarians have known how to do that for centuries
  - goal is to have this on the Web, World-wide
  - extend it to multimedia data, too
- But it is more: software agents should also be librarians!
  - e.g., help you in finding the right publications

What is needed?

- (Some) data should be available for machines for further processing
- Data should be possibly combined, merged on a Web scale
- Machines may also need to *reason* about that data
- Create a *Web of Data* (beyond the Web of Documents)

Find the right experts at NASA

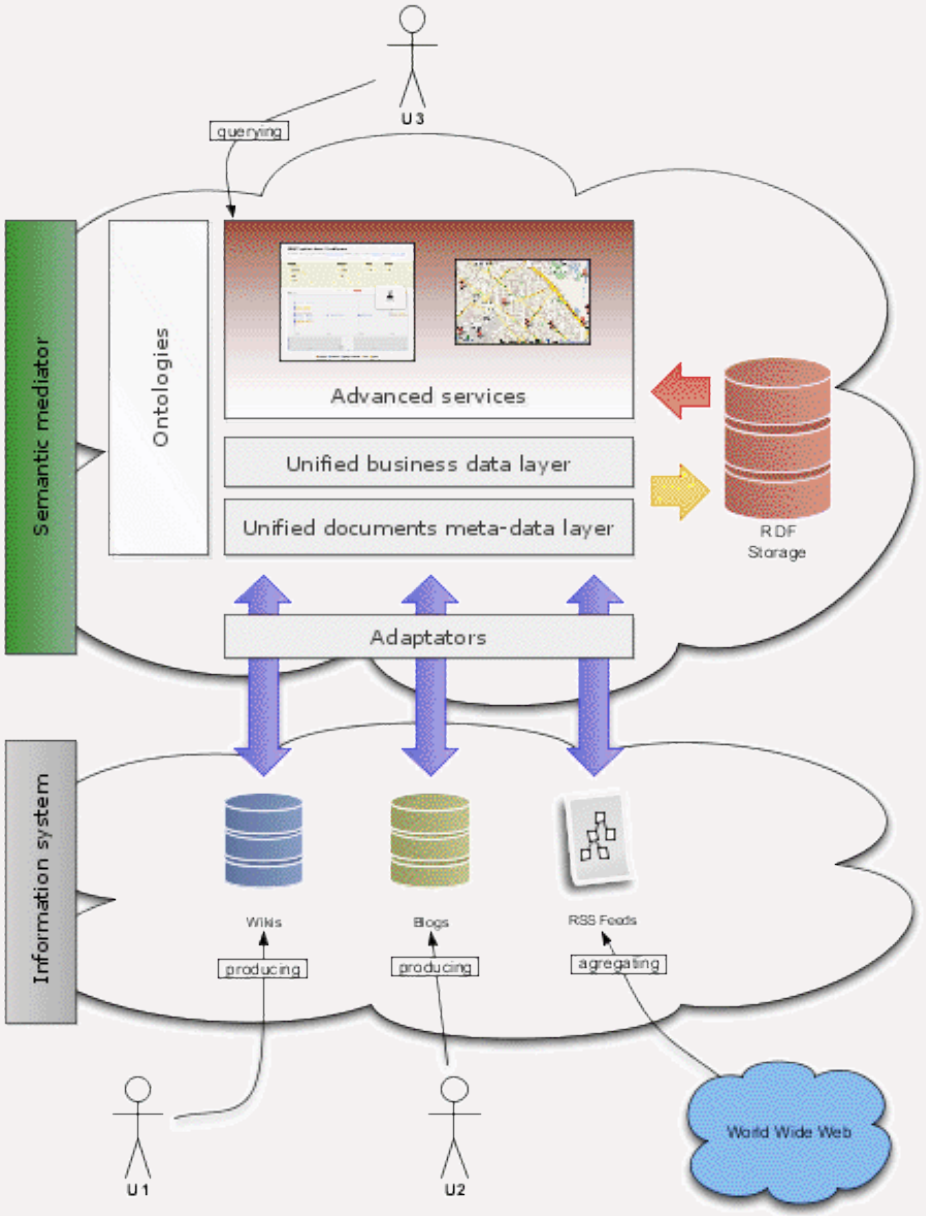
- Expertise locater for nearly 70,000 NASA civil servants, integrating 6 or 7 geographically distributed databases, data sources, and web services…



Michael Grove, Clark & Parsia, LLC, and Andrew Schain, NASA, (SWEO Case Study)

Integration of “social” software data

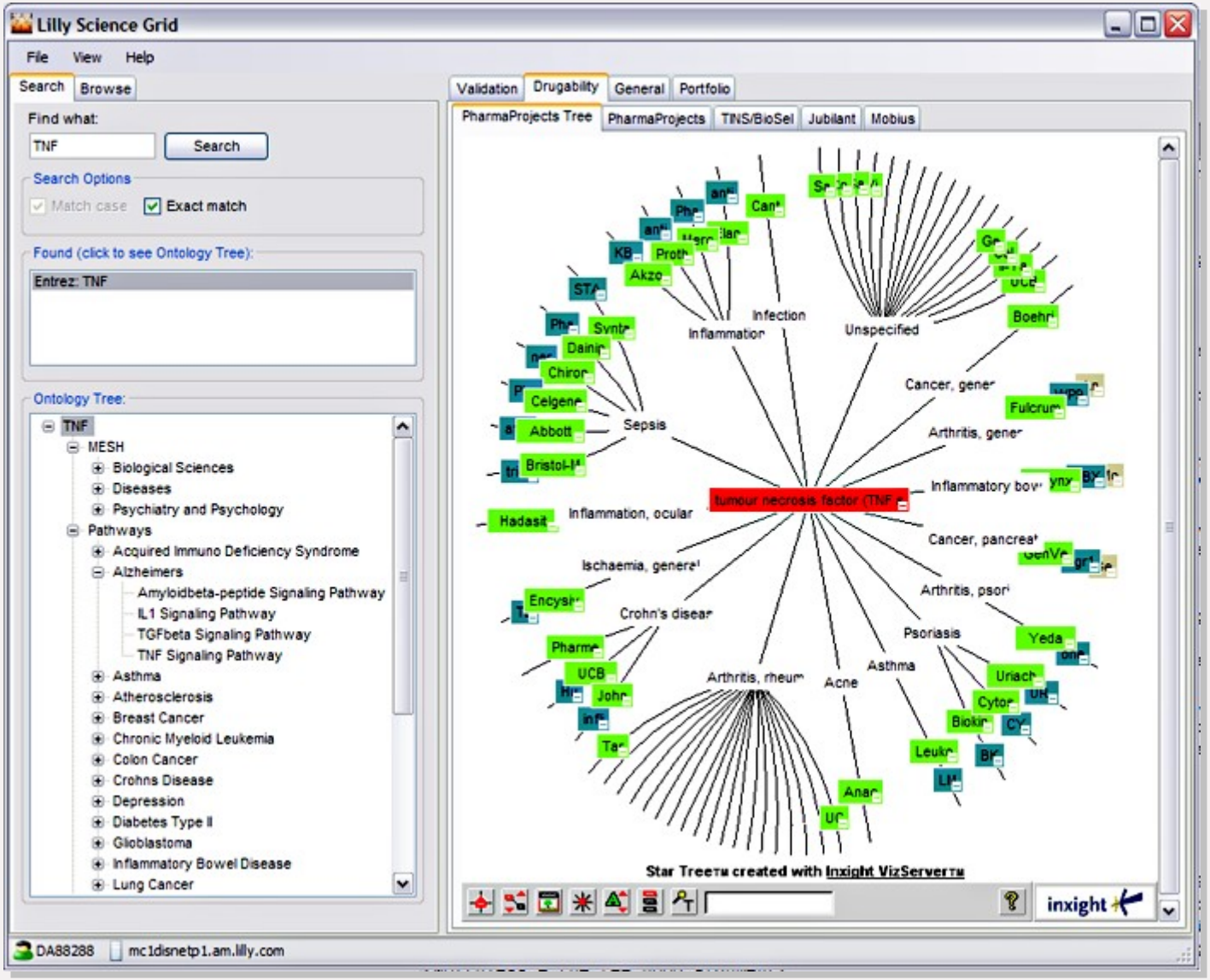
- Internal usage of wikis, blogs, RSS, etc, at EDF
  - uses:
    - public ontologies
    - public datasets for tagging
- Details are hidden from end users (via plugins, extra layers, etc)



Courtesy of Alexandre Passant, EDF R&D and LaLIC, Université Paris-Sorbonne, (SWEO Case Study)

Eli Lilly's Target Assessment Tool

- Prioritization of drug target, integrating data from different sources and formats
- Integration, search via ontologies (proprietary and public)



Courtesy of Susie Stephens, Eli Lilly (SWEO Case Study)

In what follows…

- We will use a simplistic example to introduce the main Semantic Web concepts
- We take, as an example area, data integration

The rough structure of data integration

1. Map the various data onto an abstract data representation
   - make the data independent of its internal representation…
2. Merge the resulting representations
3. Start making queries on the whole!
   - queries that could not have been done on the individual data sets

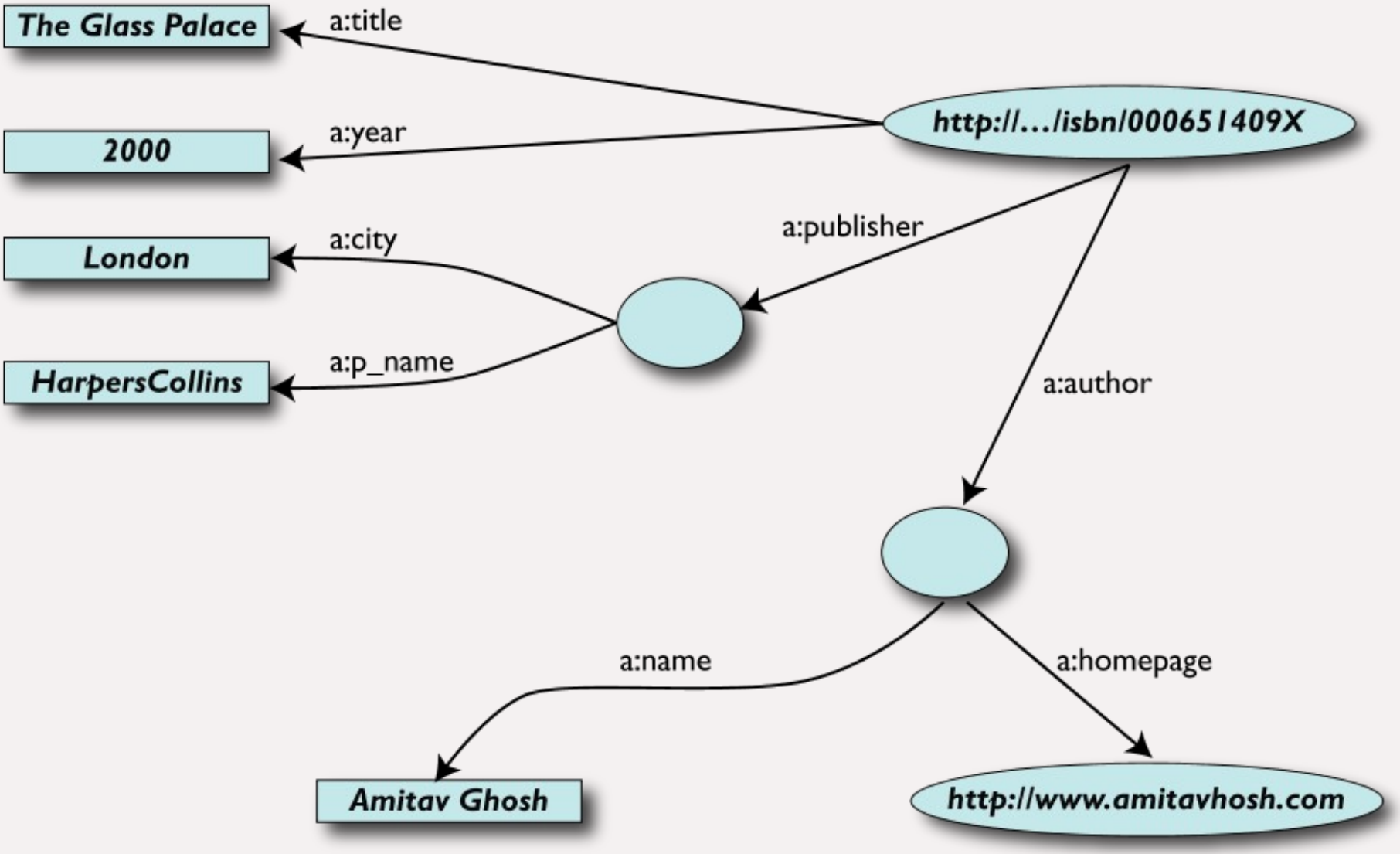
A *simplified* bookstore data (dataset "A")

ID	Author	Title	Publisher	Year
ISBN0-00-651409-X	id_xyz	The Glass Palace	id_qpr	2000

ID	Name	Home Page
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

ID	Publ. Name	City
id_qpr	Harpers Collins	London

1st: export your data as a set of *relations*



Some notes on the exporting the data

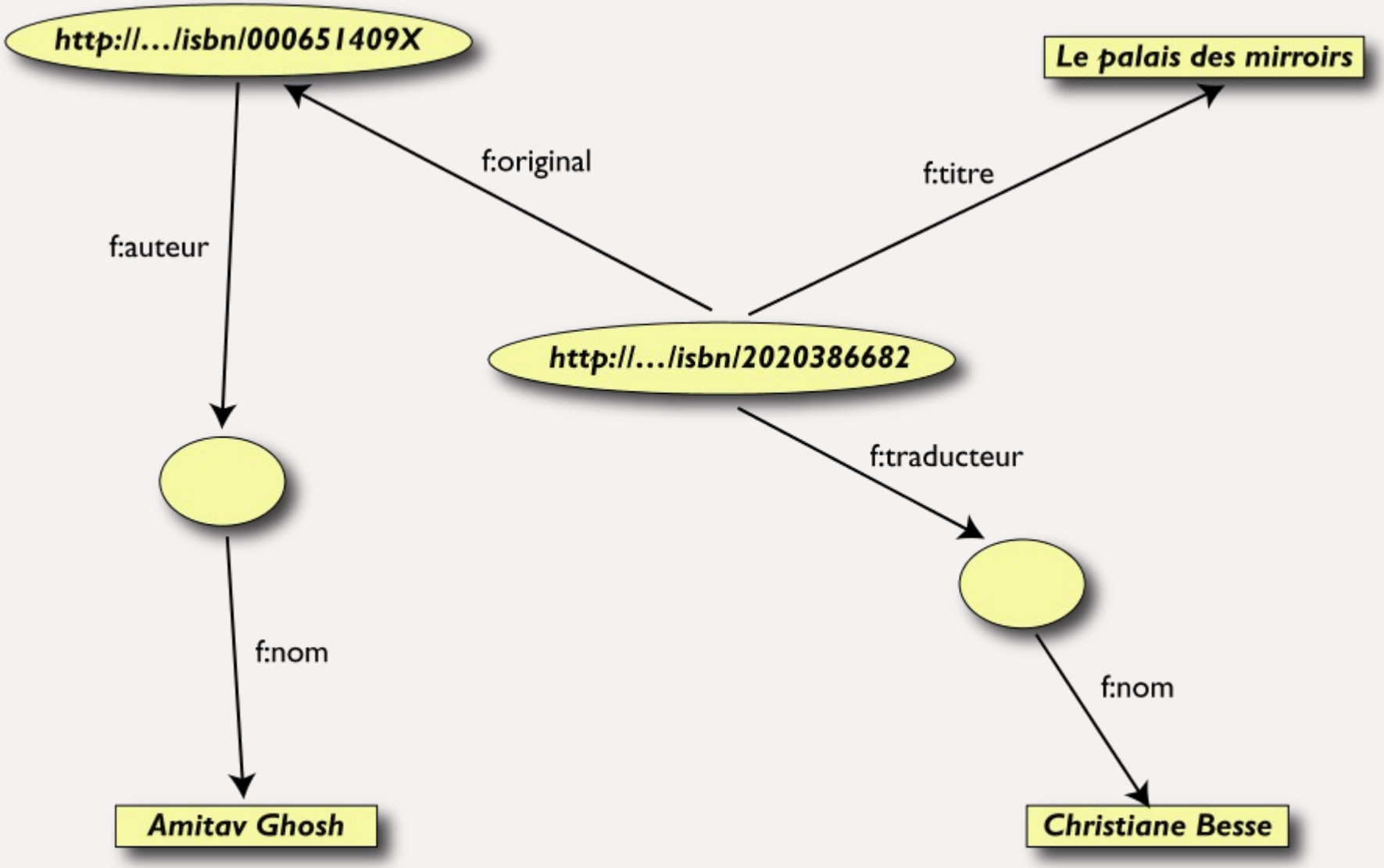
- Relations form a graph
  - the nodes refer to the “real” data or contain some literal
  - how the graph is represented in machine is immaterial for now
- Data export does *not* necessarily mean physical conversion of the data
  - relations can be generated on-the-fly at query time
    - via SQL “bridges”
    - scraping HTML pages
    - extracting data from Excel sheets
    - etc.
- One can export *part* of the data

Another bookstore data (dataset “F”)

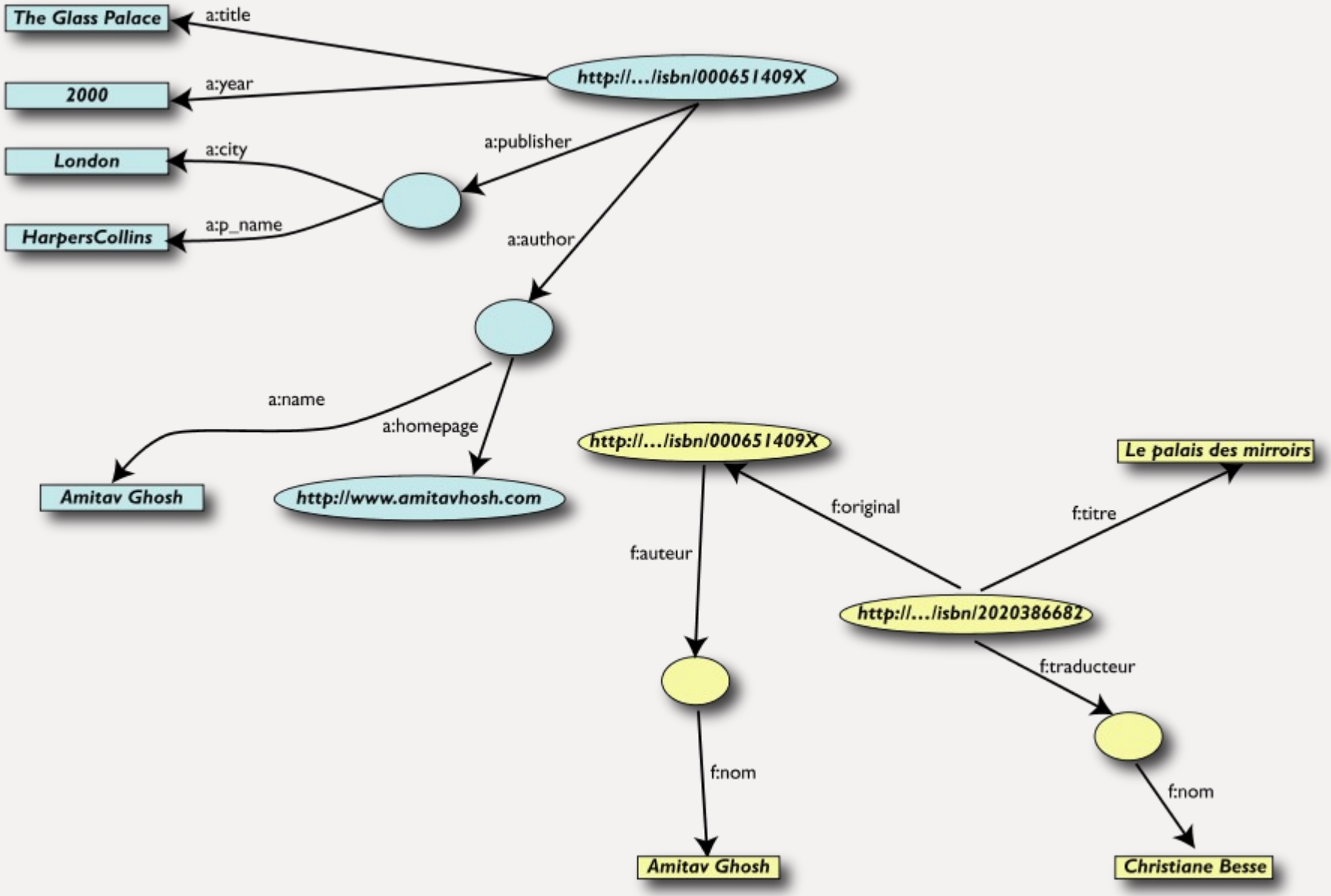
	A	B	C	D	E
1	**ID**	**Titre**	**Auteur**	**Traducteur**	**Original**
2	ISBN0 2020386682	Le Palais des miroirs	A7	A8	ISBN-0-00-651409-X
3					
4					
5					

	Nom
6	**Nom**
7	Ghosh, Amitav
8	Besse, Christianne

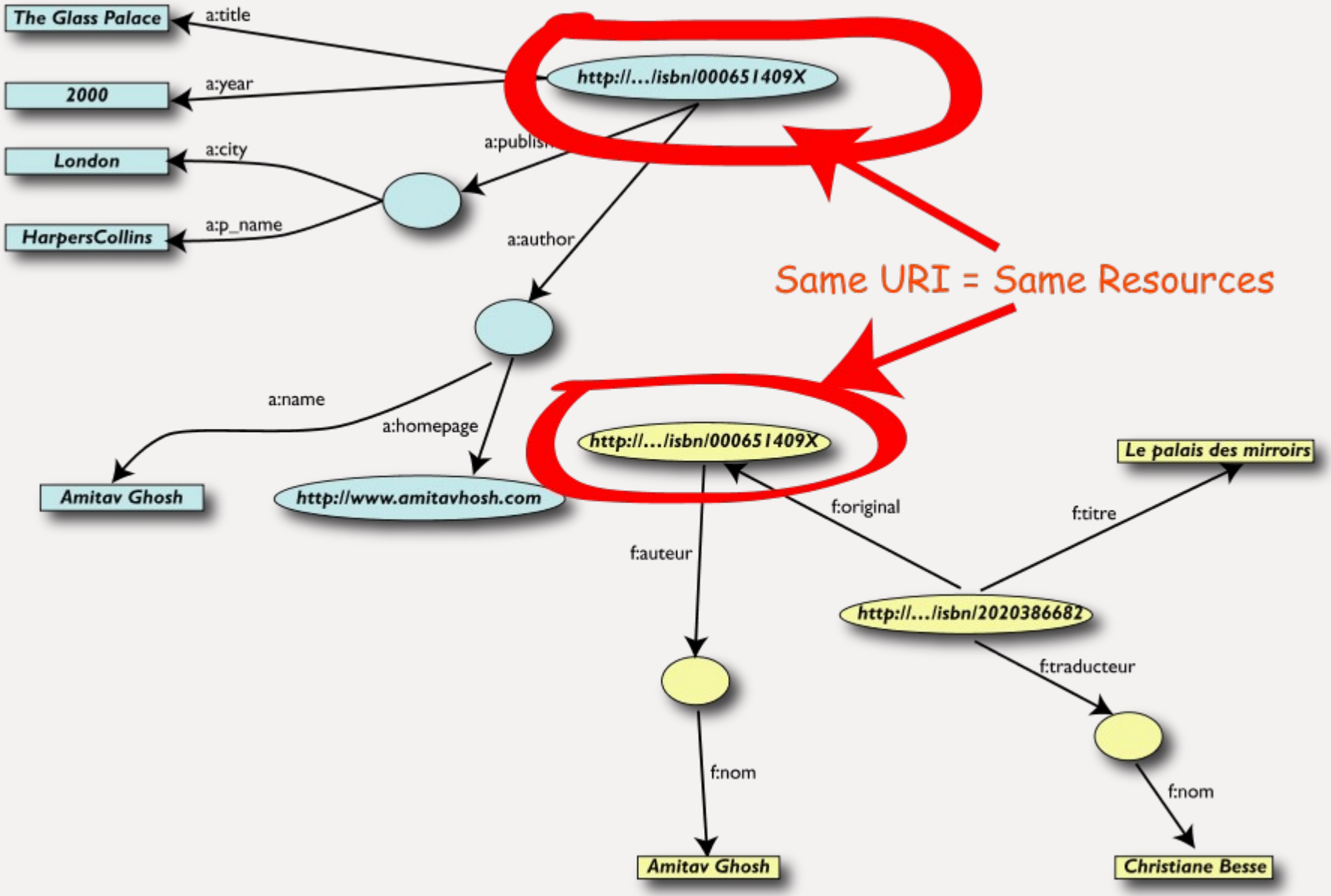
2nd: export your second set of data



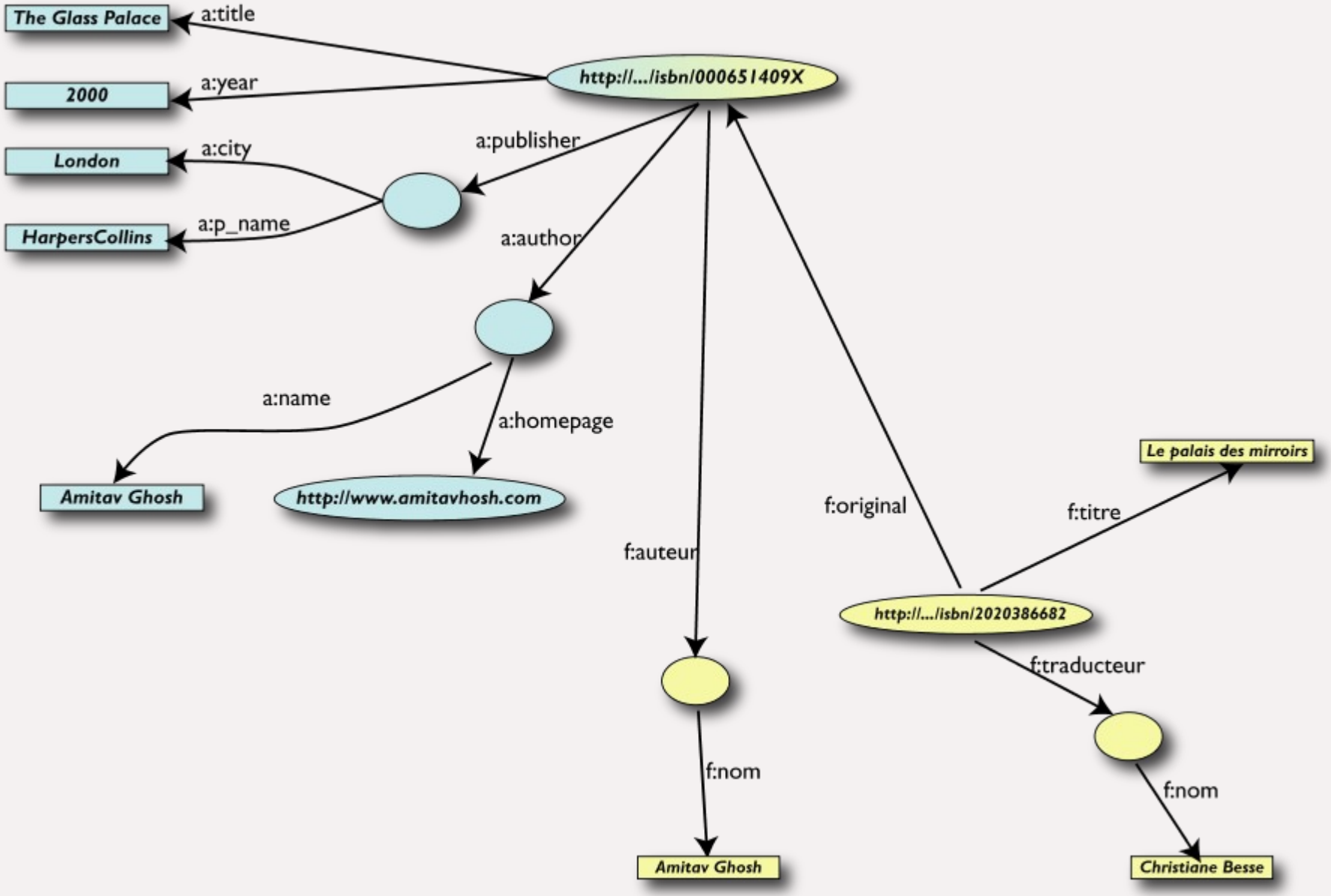
3rd: start merging your data



3rd: start merging your data (cont.)

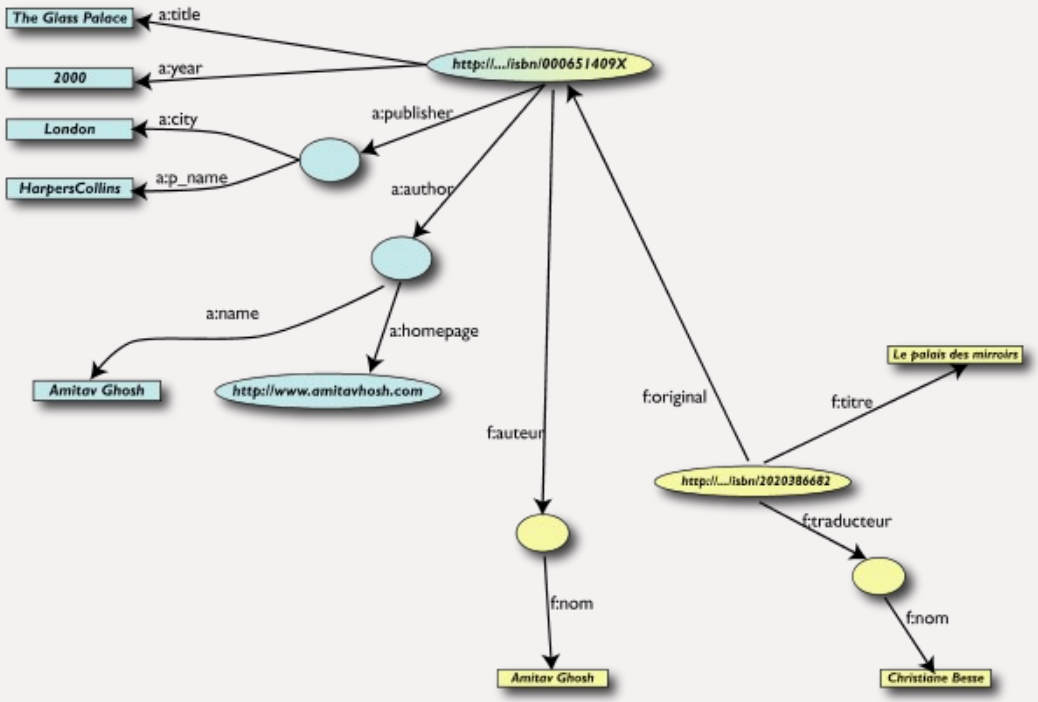


3rd: merge identical resources



Start making queries…

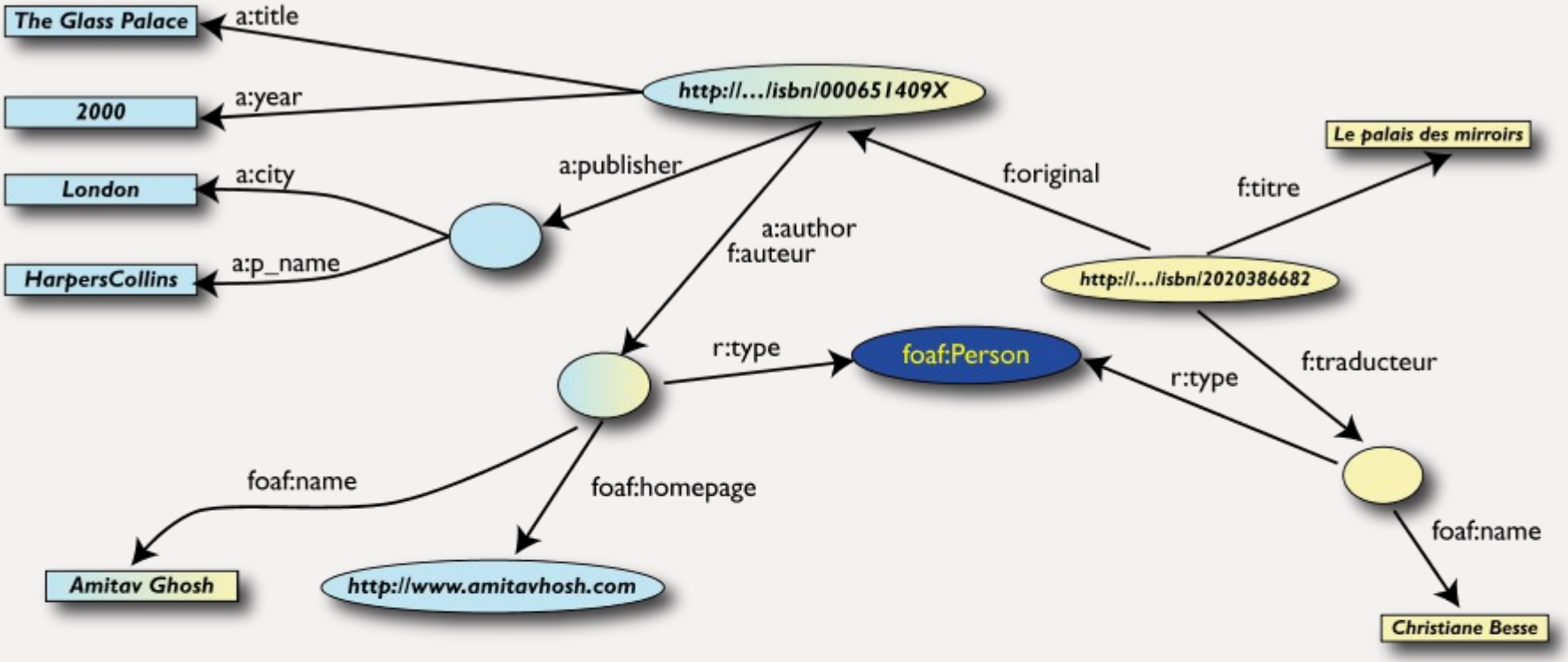
- User of data “F” can now ask queries like:
  - “give me the title of the original”
- This information is not in the dataset “F”…
- …but can be retrieved by merging with dataset “A”!

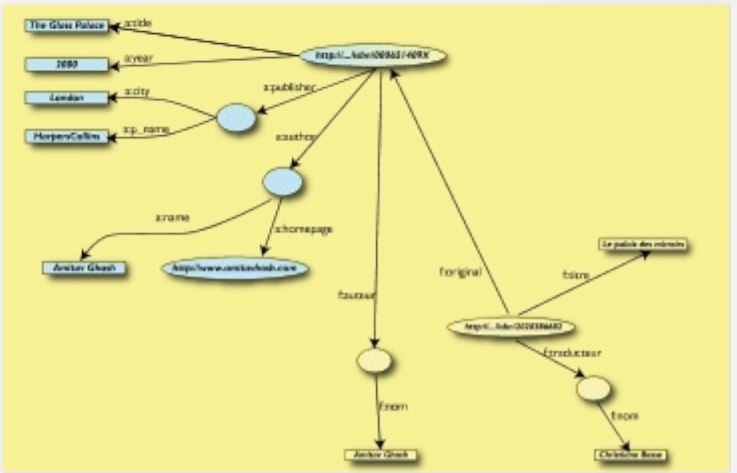


However, more can be achieved…

- We "feel" that **`a:author`** and **`f:auteur`** should be the same
- But an automatic merge does not know that!
- Let us add some extra information to the merged data:
  - **`a:author`** same as **`f:auteur`**
  - both identify a "Person"
  - a term that a community may have already defined:
    - a "Person" is uniquely identified by his/her name and, say, homepage
    - it can be used as a "category" for certain type of resources

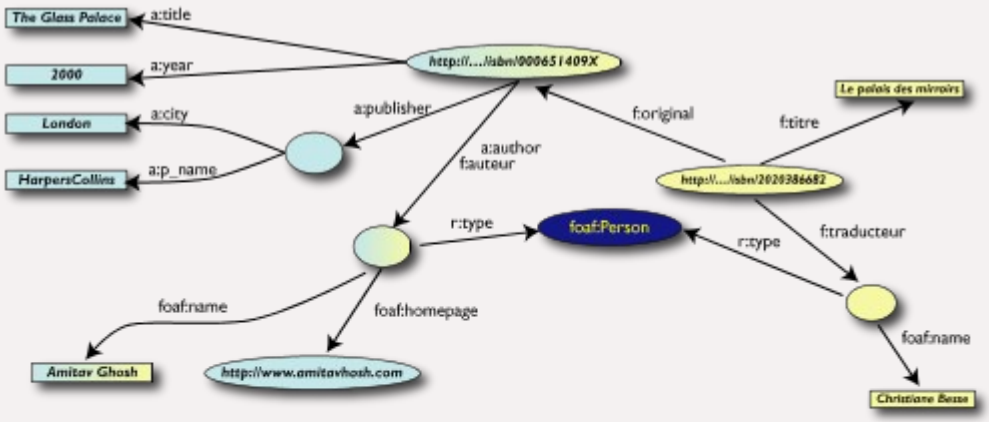
3rd revisited: use the extra knowledge





Start making richer queries!

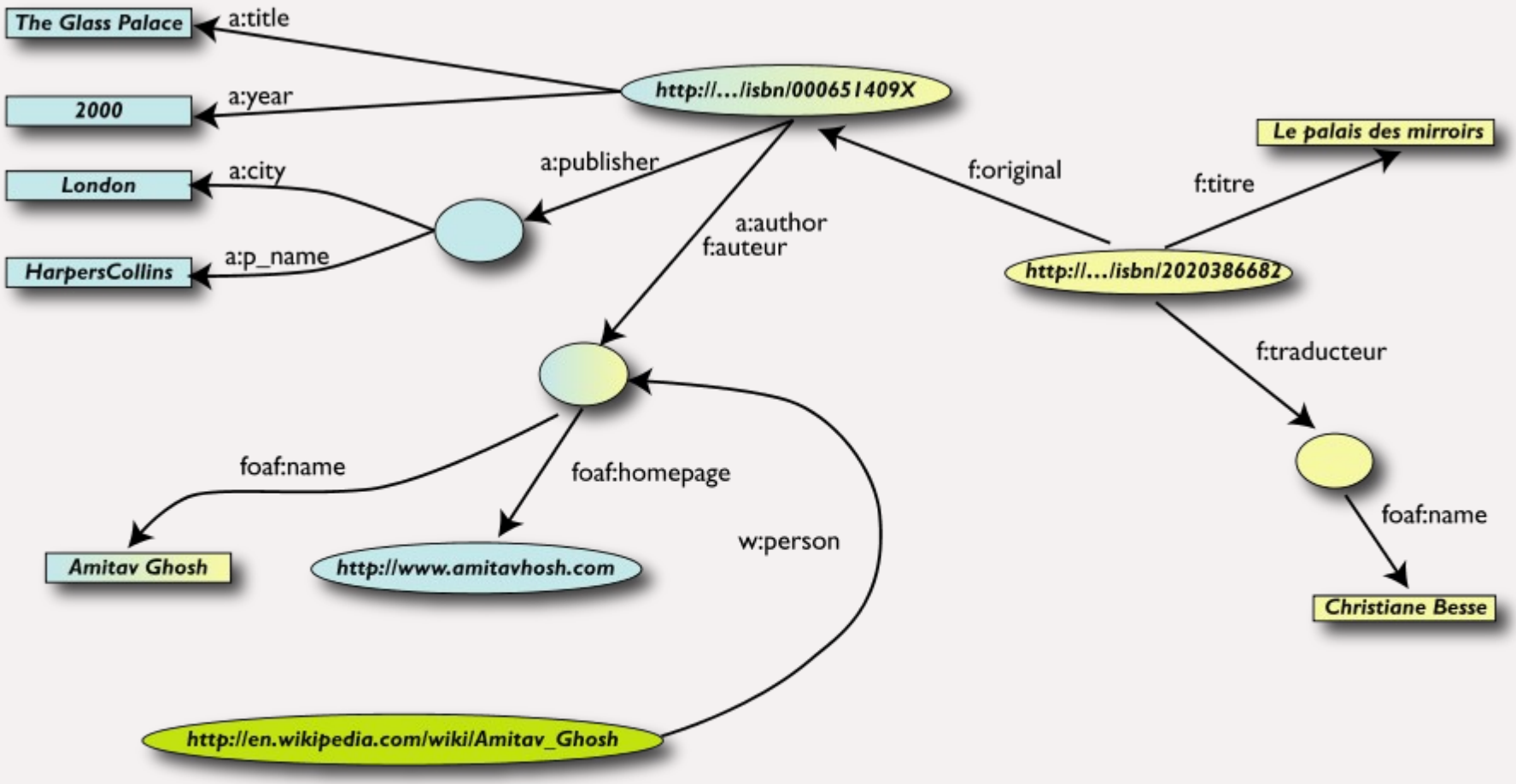
- User of dataset “F” can now query:
  - “give me the home page of the original’s author”
- The information is not in datasets “F” or “A”…
- …but was made available by:
  - merging datasets “A” and datasets “F”
  - adding three simple extra statements as an extra “glue”



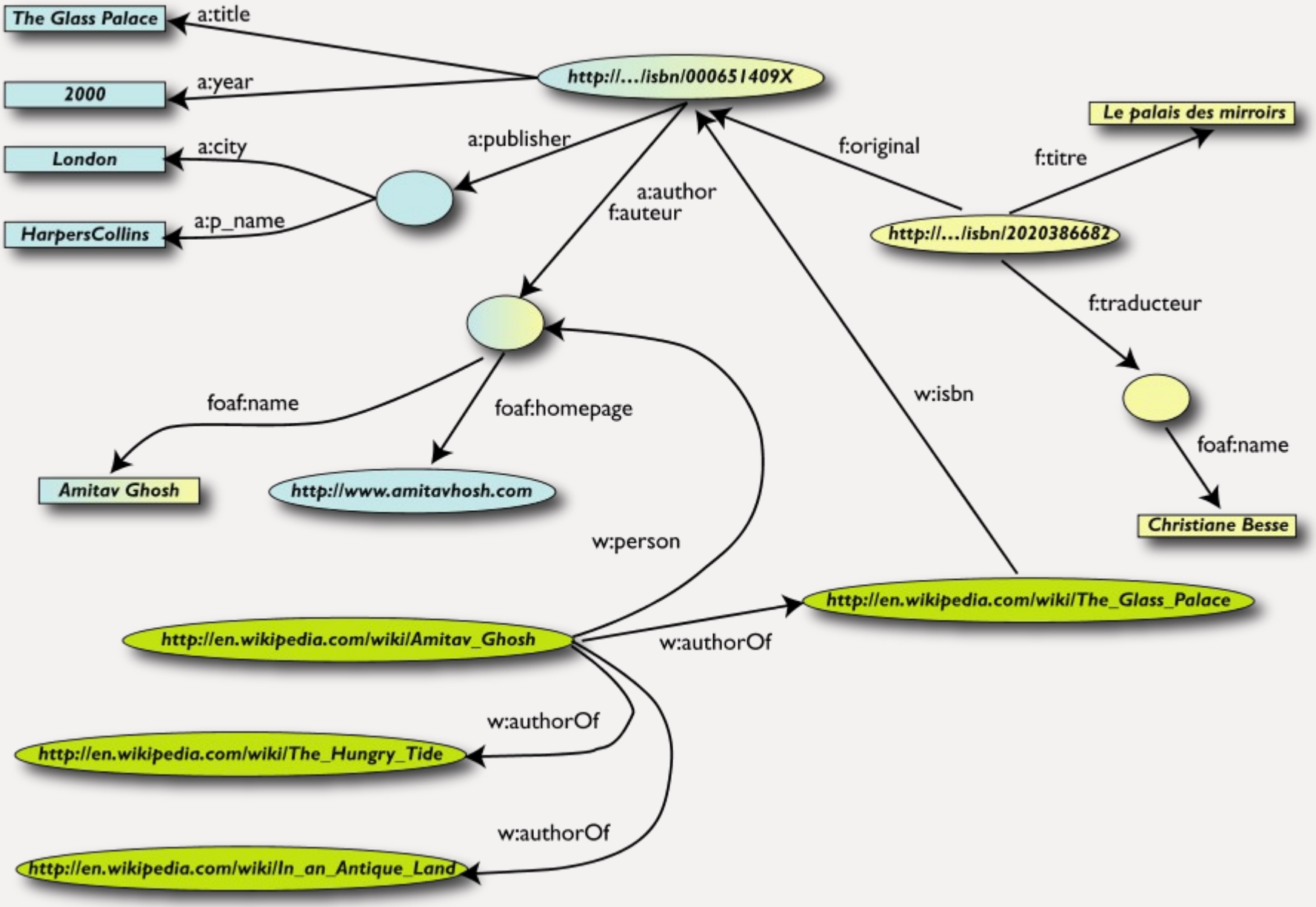
Combine with different datasets

- Using, e.g., the “Person”, the dataset can be combined with other sources
- For example, data in Wikipedia could be extracted using dedicated tools

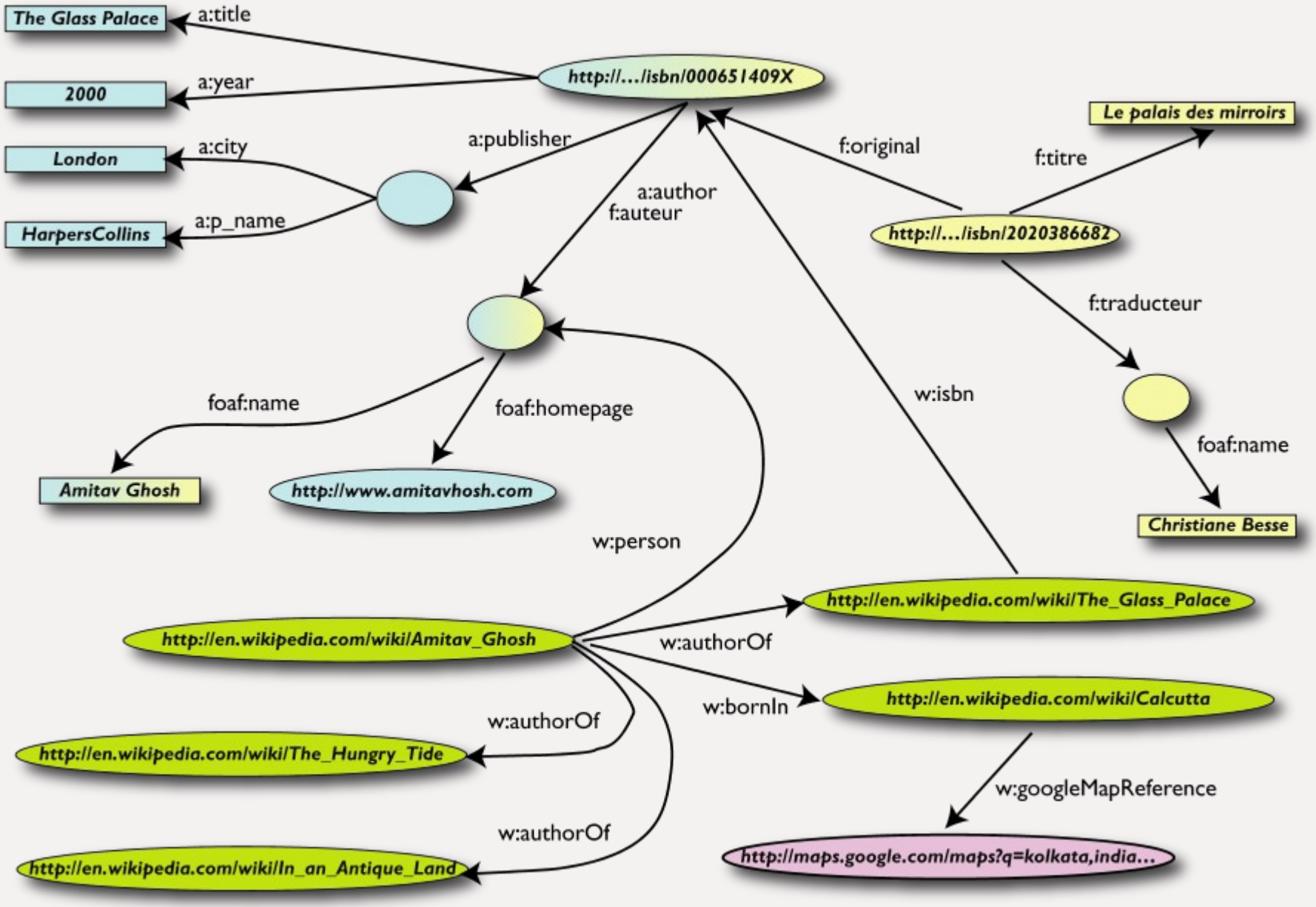
Merge with Wikipedia data



Merge with Wikipedia data



Merge with Wikipedia data



Is that surprising?

- Maybe but, in fact, no…
- What happened via automatic means is done all the time, every day by the users of the Web!
- The difference: a bit of extra rigour (e.g., naming the relationships) is necessary so that machines could do this, too

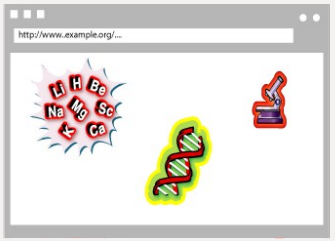
What did we do?

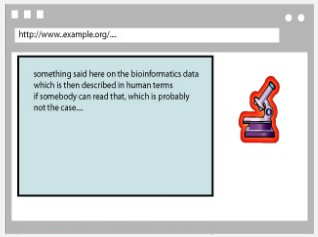
- We combined different datasets that
  - are somewhere on the web
  - are of different formats (mysql, excel sheet, XHTML, etc)
  - have different names for relations
- We could combine the data because some URI-s were identical (the ISBN-s in this case)
- We could add some simple additional information (the “glue”), including using common terminologies that a community has produced
- As a result, new relations could be found and retrieved

It could become even more powerful

- We could add extra knowledge to the merged datasets
  - e.g., a full classification of various types of library data
  - geographical information
  - etc.
- This is where *ontologies*, extra *rules*, etc, come in
  - ontologies/rule sets can be relatively simple and small, or huge, or anything in between…
- Even more powerful queries can be asked as a result

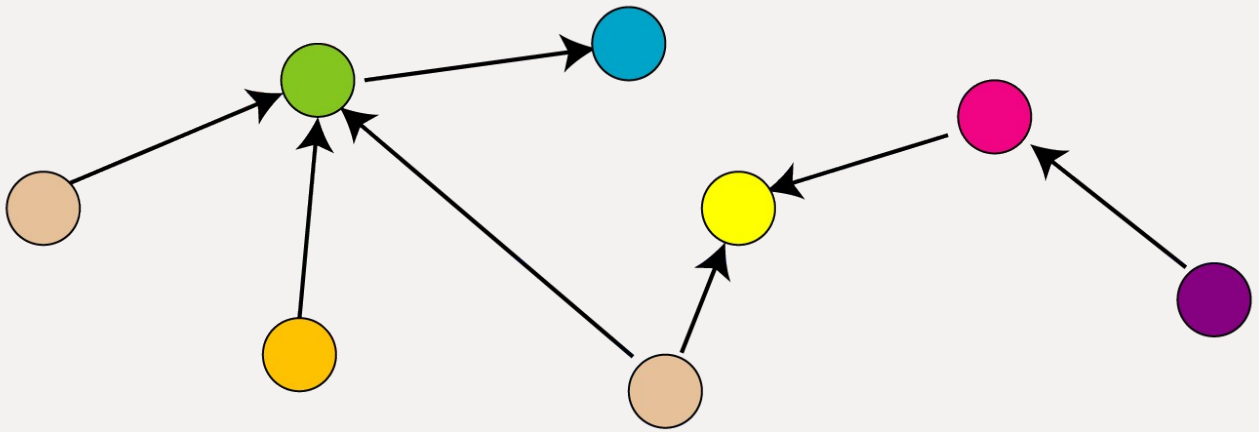
What did we do? (cont)





Applications

Query, Manipulate, etc.



Data represented in abstract format

Map, Expose, etc.

Data in various formats

So where is the Semantic Web?

- The Semantic Web provides technologies to make such integration possible!
- Hopefully you get a full picture at the end of the tutorial…

Basic RDF

RDF triples

- Let us begin to formalize what we did!
  - we “connected” the data…
  - but a simple connection is not enough… it should be named somehow
  - hence the RDF Triples: *a labelled connection between two resources*

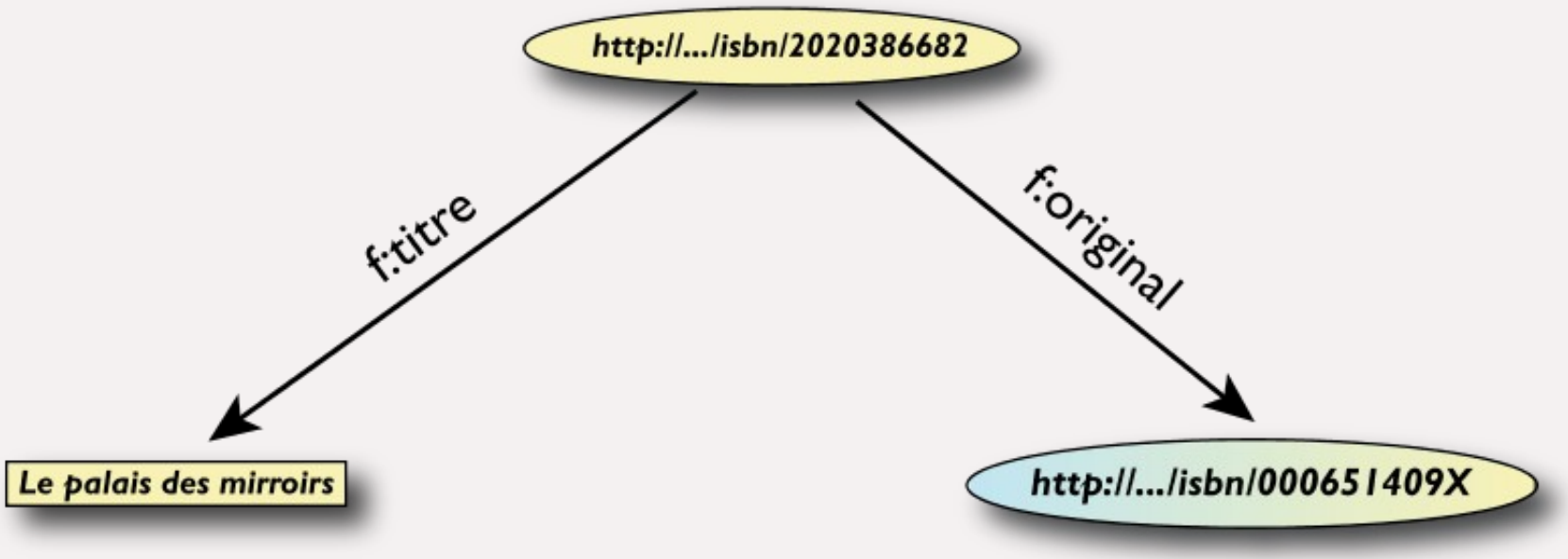
RDF triples (cont.)

- An RDF Triple **(s,p,o)** is such that:
  - "**s**", "**p**" are URI-s, ie, resources on the Web; "**o**" is a URI or a literal
    - "**s**", "**p**", and "**o**" stand for "subject", "predicate", and "object"
    - conceptually: "**p**" connects, or relates the "**s**" and the "**o**"
    - note that we use URI-s for naming: i.e., we can use **http://www.example.org/original**
  - here is the complete triple:

```
(<http://…isbn…6682>, <http://…/original>, <http://…isbn…409X>)
```

- *RDF* is a general model for such triples
  - with machine readable formats like RDF/XML, Turtle, N3, RXR, RDFa, …

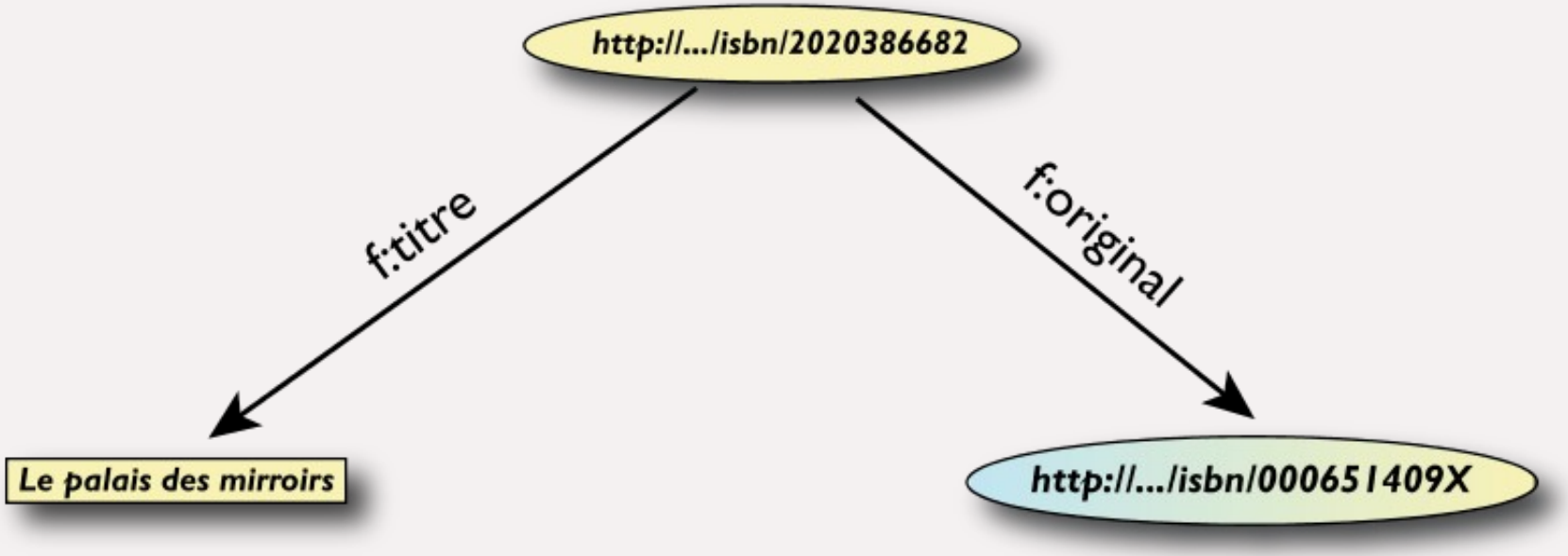
A simple RDF example (in RDF/XML)




```
<rdf:Description rdf:about="http://…/isbn/2020386682">
    <f:titre xml:lang="fr">Le palais des mirroirs</f:titre>
    <f:original rdf:resource="http://…/isbn/000651409X"/>
</rdf:Description>
```

(Note: namespaces are used to simplify the URI-s)

A simple RDF example (in Turtle)




```
<http://…/isbn/2020386682>
    f:titre "Le palais des mirroirs"@fr ;
    f:original <http://…/isbn/000651409X> .
```

URI-s play a fundamental role

- URI-s made the merge possible
- Anybody can create (meta)data on any resource on the Web
  - e.g., the same XHTML file could be annotated through other terms
  - semantics is added to existing Web resources via URI-s
  - URI-s make it possible to link (via properties) data with one another
- *URI-s ground RDF into the Web*
  - information can be retrieved using existing tools
  - this makes the “Semantic Web”, well… “Semantic *Web*”

RDF in programming practice

- For example, using Java+Jena (HP's Bristol Lab):
  - a "Model" object is created
  - the RDF file is parsed and results stored in the Model
  - the Model offers methods to retrieve:
    - triples
    - (property,object) pairs for a specific subject
    - (subject,property) pairs for specific object
    - etc.
  - the rest is conventional programming…
- Similar tools exist in Python, PHP, etc.

Vocabularies

Need for vocabulary specifications

- We saw the need and the power of vocabulary specification in our example
  - “Person”
  - “author” vs. “auteur”
  - etc
- Several technologies are provided for that purpose: RDFS, OWL, SKOS, …
  - trade-off between expressibility and simplicity (eg, for implementations)

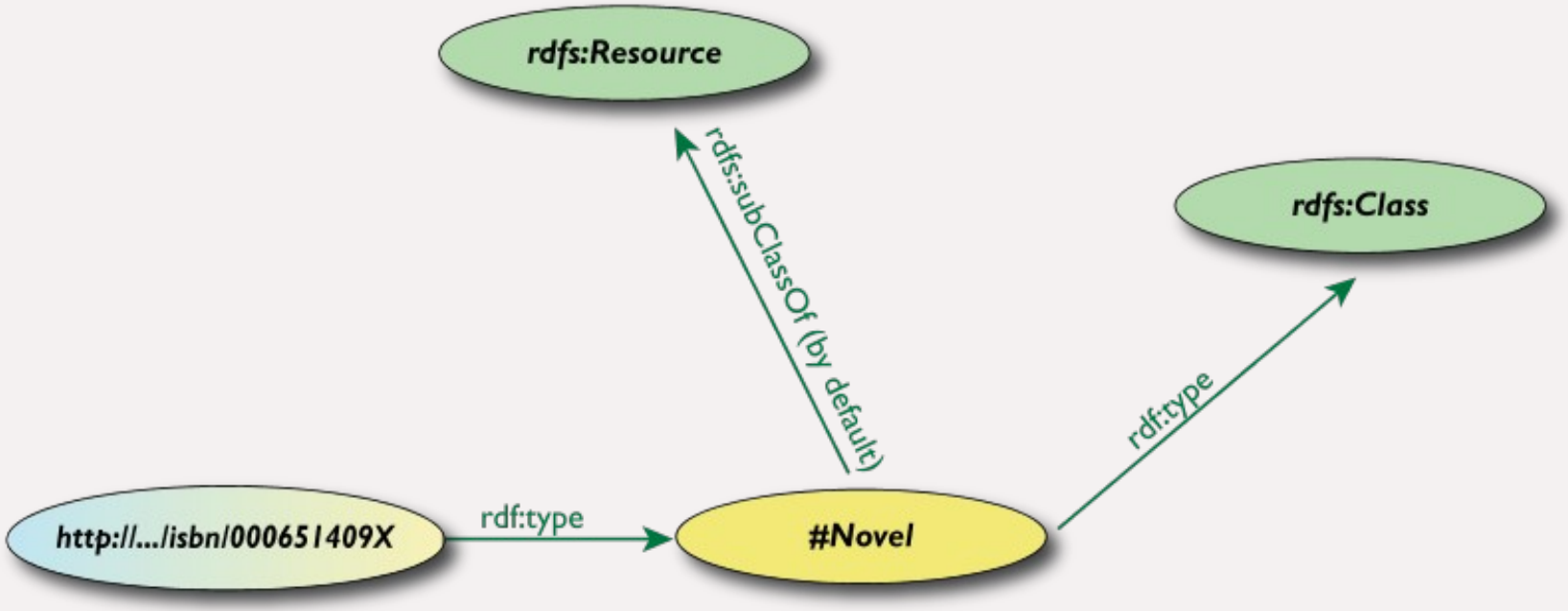
RDFS: the basics

- Think of well known traditional ontologies or taxonomies:
  - use the term “novel”
  - “every novel is a fiction”
  - “«The Glass Palace» is a novel”
  - etc.
- RDFS defines resources and classes:
  - everything in RDF is a “resource”
  - “classes” are also resources, but…
  - …they are also a collection of possible resources (i.e., “individuals”)
    - “fiction”, “novel”, …

Classes, resources, … (cont.)

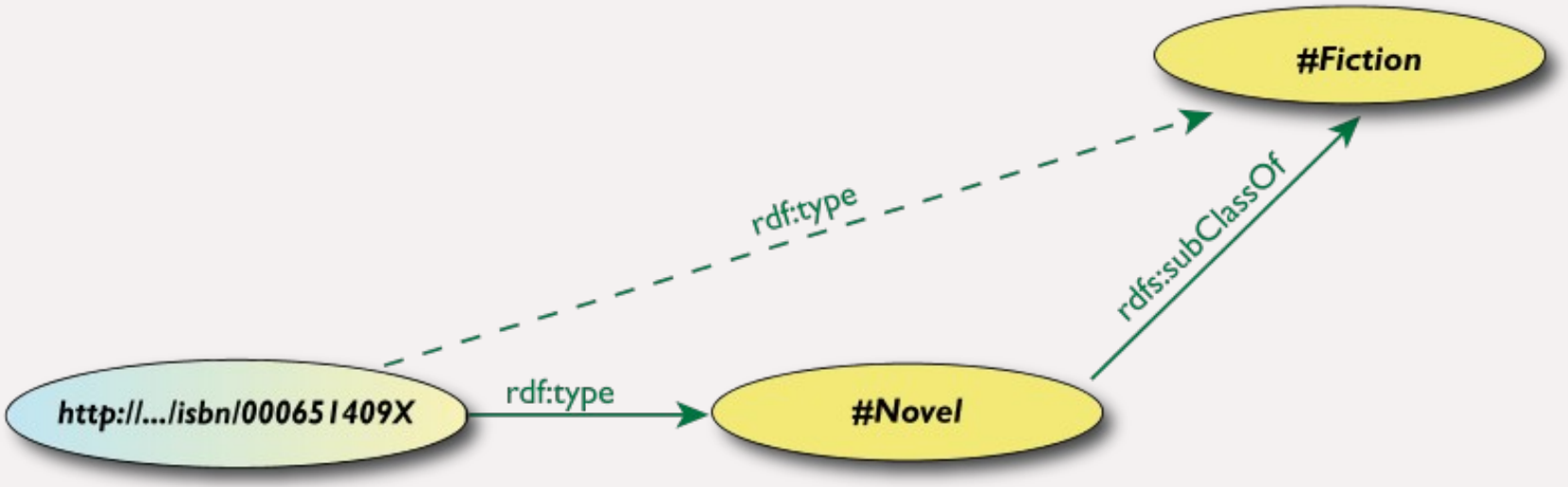
- Relationships are defined among classes/resources:
  - "typing": an individual belongs to a specific class ("«The Glass Palace» is a novel")
    - to be more precise: "«http://.../**000651409X**» is a novel"
  - "subclassing": *all* instances of one are also the instances of the other ("every novel is a fiction")
- *RDFS formalizes these notions in RDF*

Classes, resources in RDF(S)



- RDFS defines **`rdfs:Resource`**, **`rdfs:Class`** as nodes; **`rdf:type`**, **`rdfs:subClassOf`** as properties
  - (these are all special URI-s, we just use the namespace abbreviation)

Inferred properties




```
(<http://…/isbn/000651409X> rdf:type #Fiction)
```

- is not in the original RDF data…
- …but can be inferred from the RDFS rules
- "RDFS aware" RDF environments return that triple, too

Inference: let us be formal…

- The RDF Semantics document has a list of (44) *entailment rules*:
  - "if such and such triples are in the graph, add this and this triple"
  - do that recursively until the graph does not change
- The relevant rule for our example:

```
If:
  uuu rdfs:subClassOf xxx .
  vvv rdf:type uuu .
Then add:
  vvv rdf:type xxx .
```

Properties

- Properties' range and domain can be specified
  - i.e., what type of resources can serve as object and subject
- There is also a possibility for a "sub-property"
  - all resources bound by the "sub" are also bound by the other

Property specification example

- In RDF/XML:

```
<rdf:Property rdf:ID="title">
  <rdfs:domain rdf:resource="#Fiction"/>
  <rdfs:range rdf:resource="http://...#Literal"/>
</rdf:Property>
```

- In Turtle:

```
:title
  rdf:type     rdf:Property;
  rdfs:domain  :Fiction;
  rdfs:range   rdfs:Literal.
```

What does this mean?

- Again, new relations can be deduced. Indeed, if

```
:title
  rdf:type     rdf:Property;
  rdfs:domain  :Fiction;
  rdfs:range   rdfs:Literal.

<http://…/isbn/000651409X> :title "The Glass Palace" .
```

- then the system can *infer* that:

```
<http://…/isbn/000651409X> rdf:type :Fiction .
```

A bit of RDFS can take you far…

- Remember the power of merge?
- We could have used, in our example:
  - `f:auteur` is a sub-property of `a:author` and vice versa (although we will see other ways to do that…)
- Of course, in some cases, more complex knowledge is necessary

Ontologies

- Complex applications may want more possibilities:
  - identification of objects with different URI-s
  - disjointness or equivalence of classes
  - construct classes, not only name them
  - more complex classification schemes
  - can a program reason about some terms? E.g.:
    - “if «Person» resources «A» and «B» have the same «`foaf:email`» property, then «A» and «B» are identical”
  - etc.
- This is where OWL comes in

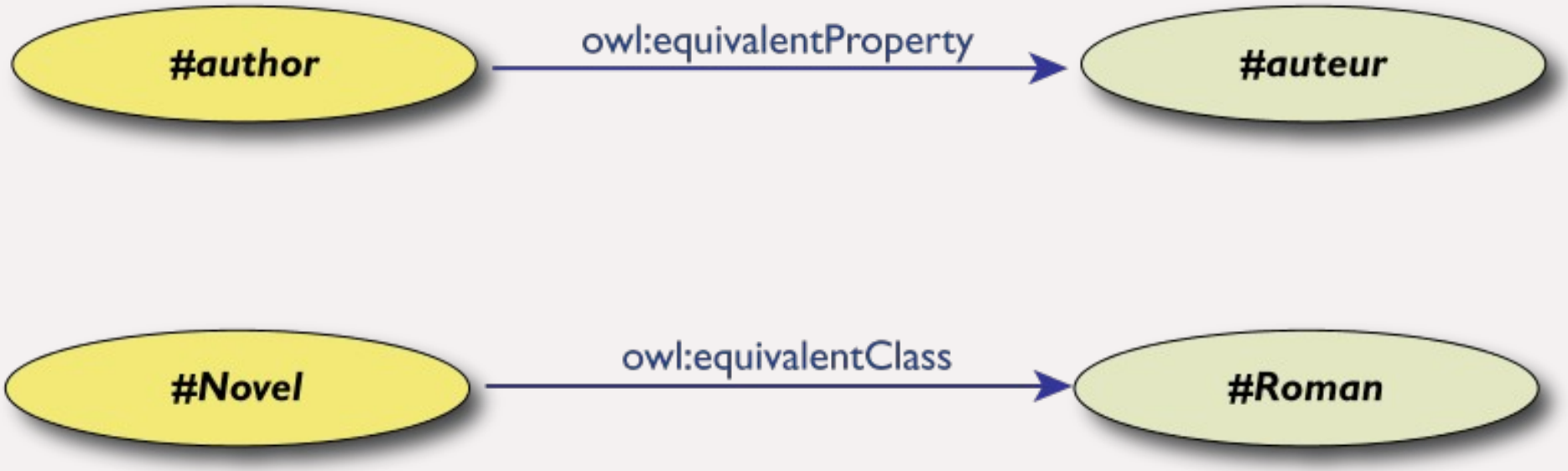
Web Ontology Language = OWL

- OWL is an extra layer, a bit like RDF Schemas
  - own namespace, own terms
  - it relies on RDF Schemas
- Needs extra tools, inference engines, etc, to implement it
  - that is why it is a separate recommendation
  - use it only if you need it…

OWL examples: term equivalence

- For classes:
  - **`owl:equivalentClass`**: two classes have the same individuals
  - **`owl:disjointWith`**: no individuals in common
- For properties:
  - **`owl:equivalentProperty`**
    - remember the **`a:author`** vs. **`f:auteur`**?
- For individuals:
  - **`owl:sameAs`**: two URIs refer to the same concept ("individual")
  - **`owl:differentFrom`**: negation of **`owl:sameAs`**

Example: connecting to French

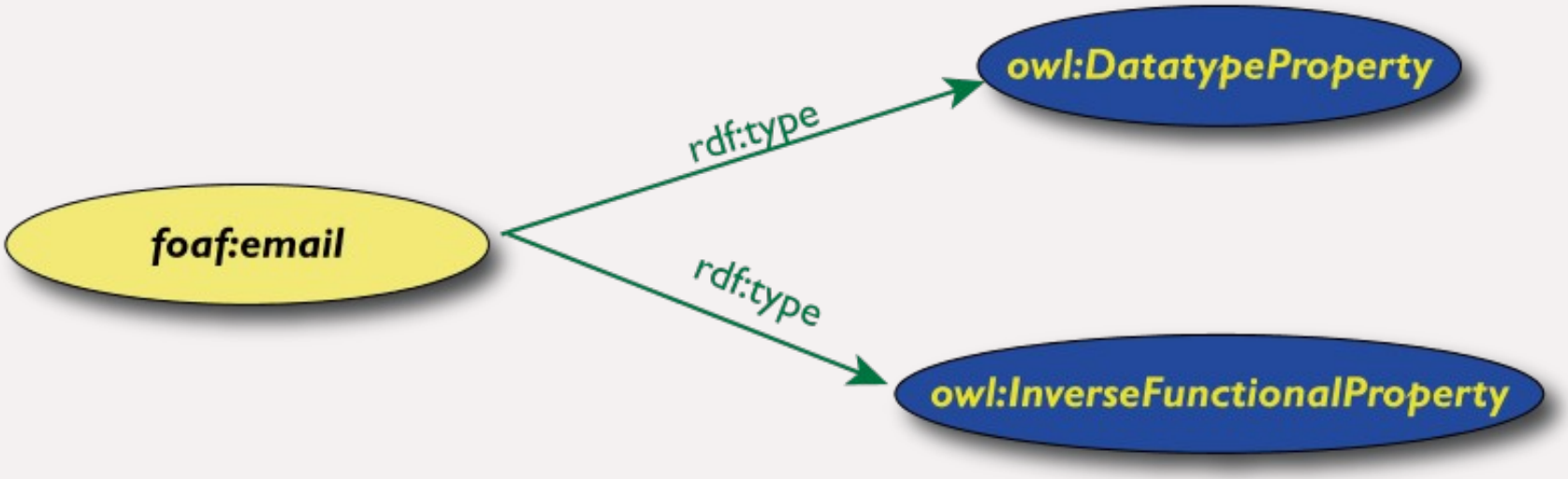


Property characterization

- Characterize the behaviour of properties: symmetric, transitive, functional, inverse functional…

Characterization example

- "`foaf:email`" is inverse functional (i.e., two different subjects cannot have identical objects)



What this means is…

- If the following holds in our triples:

```
:email rdf:type owl:InverseFunctionalProperty.
<A> :email "mailto:a@b.c".
<B> :email "mailto:a@b.c".
```

- then the following holds, too:

```
<A> owl:sameAs <B>.
```

- I.e., *new relationships* were discovered again (beyond what RDFS could do)

Classes in OWL

- In RDFS, you can subclass existing classes… that's all
- In OWL, you can *construct* classes from existing ones:
  - enumerate its content
  - through intersection, union, complement
  - etc

What we have so far…

- The OWL features listed so far are already fairly powerful
- E.g., various databases can be linked via **`owl:sameAs`**, functional or inverse functional properties, etc.
- It is still possible to find inferred relationships using a traditional rule engine

However… even that may not be enough

- Very large vocabularies (ontologies) might require even more complex features
  - one major issue is the way classes (i.e., “concepts”) are defined.
  - for example, define a class by characterizing the value a property can have on its instances
    - eg: allowed prices should have their currency set to €…
- OWL includes those extra features but… the inference engines become (much) more complex 

OWL has “subspecies”

- Some features of OWL require complex programs
- OWL has defined “subspecies”: OWL Lite, OWL DL, OWL Full
  - applications choose a level that is good enough for them
  - current work on OWL may define some more

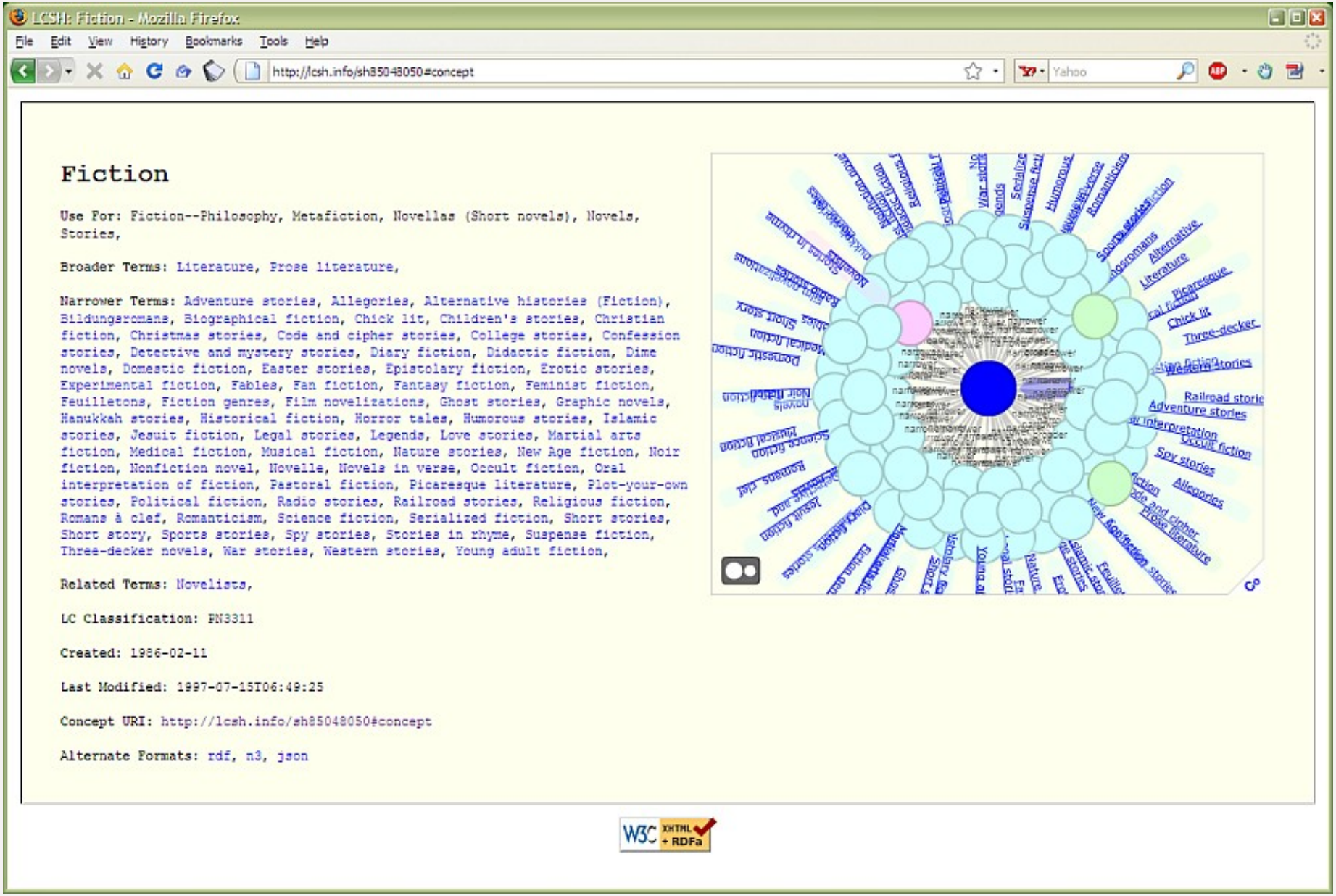
Ontologies examples

- Large ontologies are being developed
  - eClassOwl: eBusiness ontology for products and services, 75,000 classes and 5,500 properties
  - National Cancer Institute's ontology: about 58,000 classes
  - Open Biomedical Ontologies Foundry: a collection of ontologies, including the Gene Ontology to describe gene and gene product attributes in any organism or protein sequence and annotation terminology and data (UniProt)
  - BioPAX: for biological pathway data

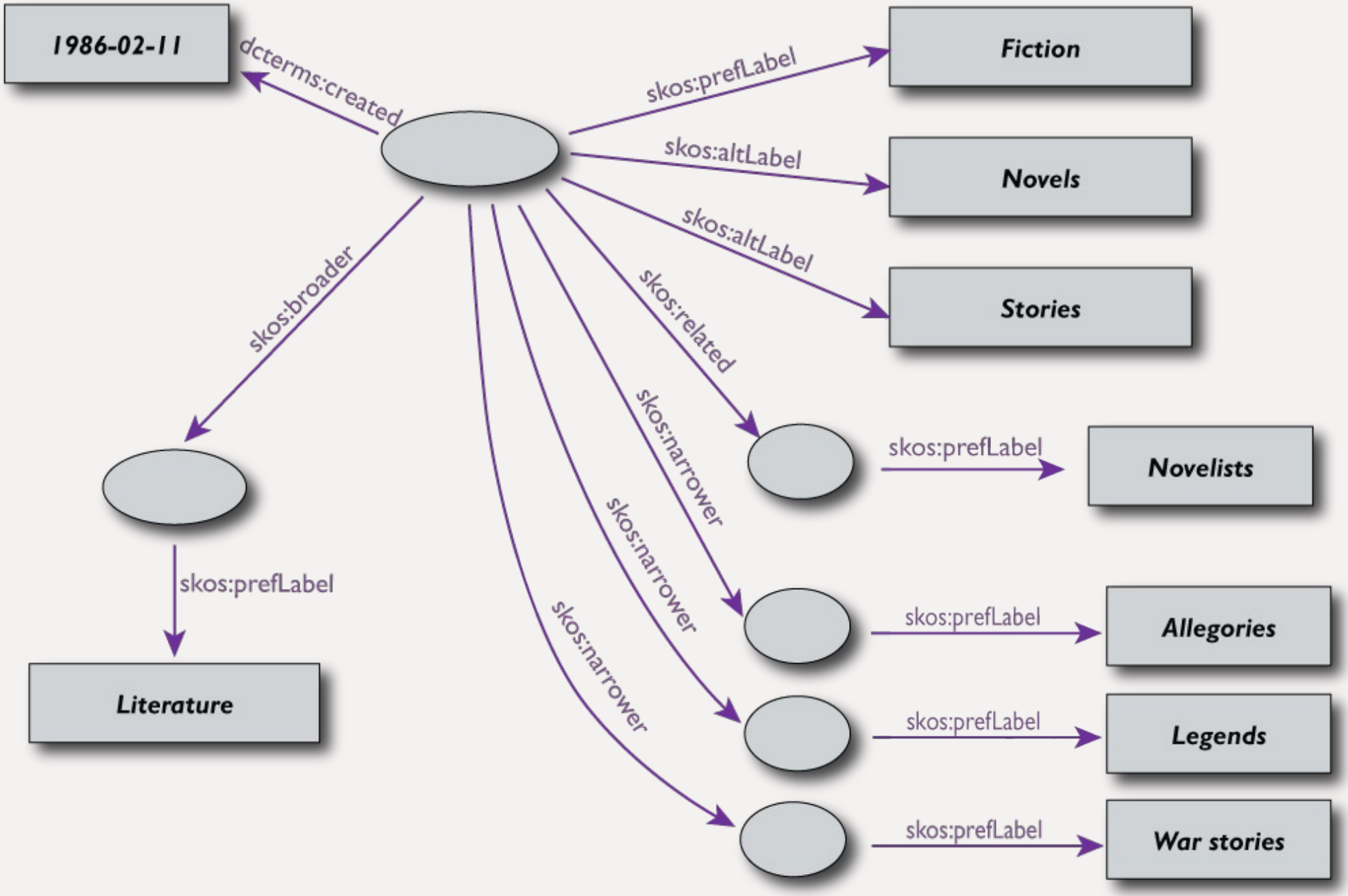
Another vocabulary specification tool: SKOS (Simple Knowledge Organization System)

- Goal: represent and share classifications, glossaries, thesauri, etc, as developed in the “Print World”.
  - for example:
    - Dewey Decimal Classification, Art and Architecture Thesaurus, ACM classification of keywords and terms…
    - US Library of Congress subject headings
- Allow for a quick port of this traditional data, combine it with other data
- No need for complex inferencing

Example: Library of Congress subject headings



The structure can be translated into RDF



SKOS and OWL

- SKOS is geared towards some specific (though large) use cases, like
  - taxonomies, glossaries, …
  - annotations of complex structures
- SKOS is a based on a very simple usage of OWL
  - roughly on the rule based level
  - no complex inference possibilities
- Ideal tool to create “bridges” between the library world and the (Semantic) Web
  - eg, by providing standard “terms” (ie, their URI-s)

How to get to RDF(S) data?

How to get to RDF(S) data?

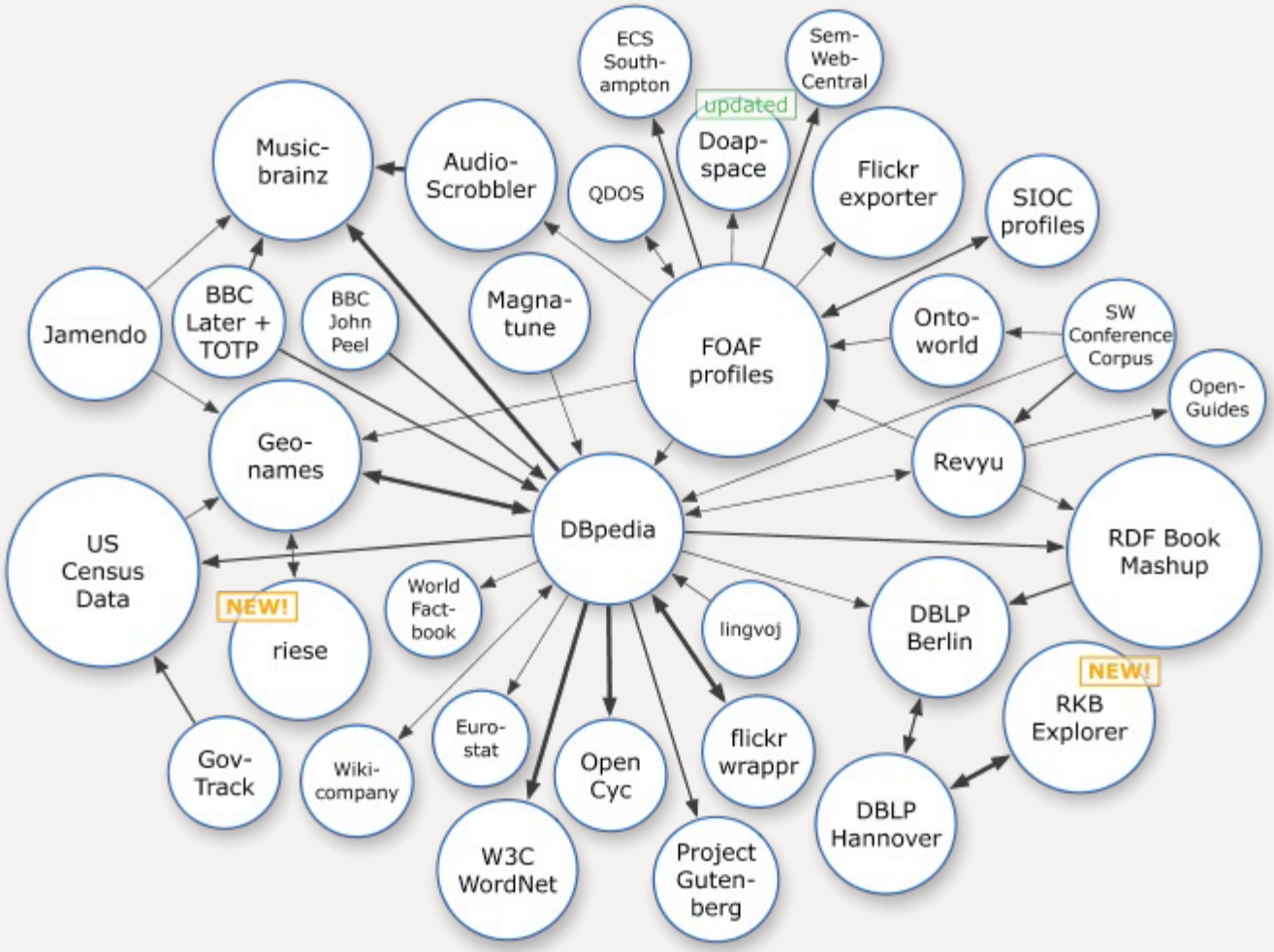
- Write RDF/XML or Turtle “manually”
  - in some cases that is necessary, but it really does not scale…
- Embed RDF data into HTML (using, eg, RDFa)
- Extract RDF from XML data (using, eg, GRDDL)
- …
- But this does not scale

Access to databases

- “Bridges” are defined to get to RDBS content
  - the same database system can be used in a traditional environment
  - but its content can be “viewed” as RDF
- Specialized RDF database systems also exist to store large amount of data

Example: Linking Open Data Project

- Goal: “expose” open datasets in RDF via various means
- *Set RDF links among the data items* from different datasets
- Billions triples, millions of “links”



DBpedia: Extracting structured data from Wikipedia

```
http://en.wikipedia.org/wiki/Kolkata
```

```
<http://dbpedia.org/resource/Kolkata>
  dbpedia:native_name "Kolkata (Calcutta)"@en;
  dbpedia:altitude "9";
  dbpedia:populationTotal  "4580544";
  dbpedia:population_metro "14681589";
  geo:lat "22.56970024108887"^^xsd:float;
 ...
```

Kolkata (Chutiyon ka Shahar)

West Bengal • India

Victoria Memorial

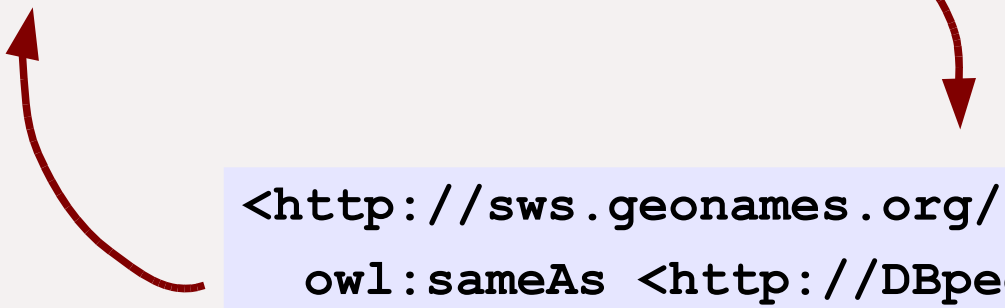
Coordinates:	22.5697, 88.3697
Time zone	IST (UTC+5:30)
Area • Elevation	1,480 km² (571 sq mi) • 9 m (30 ft)
District(s)	Calcutta †
Population • Density • Metro	14,681,589 (2001) • 9,920/km² (25,693/sq mi) • 4,580,544
Mayor	Bikash Ranjan Bhattacharya
Codes • Pincode • Telephone	 • 700 xxx • +91 (33)
Website: www.kolkatamycity.com	

† The Kolkata urban agglomeration also includes portions of North 24 Parganas and South 24 Parganas districts.

Automatic links among open datasets

```
<http://dbpedia.org/resource/Kolkata>
  owl:sameAs <http://sws.geonames.org/1275004/>;
 ...
```

DBpedia



Geonames

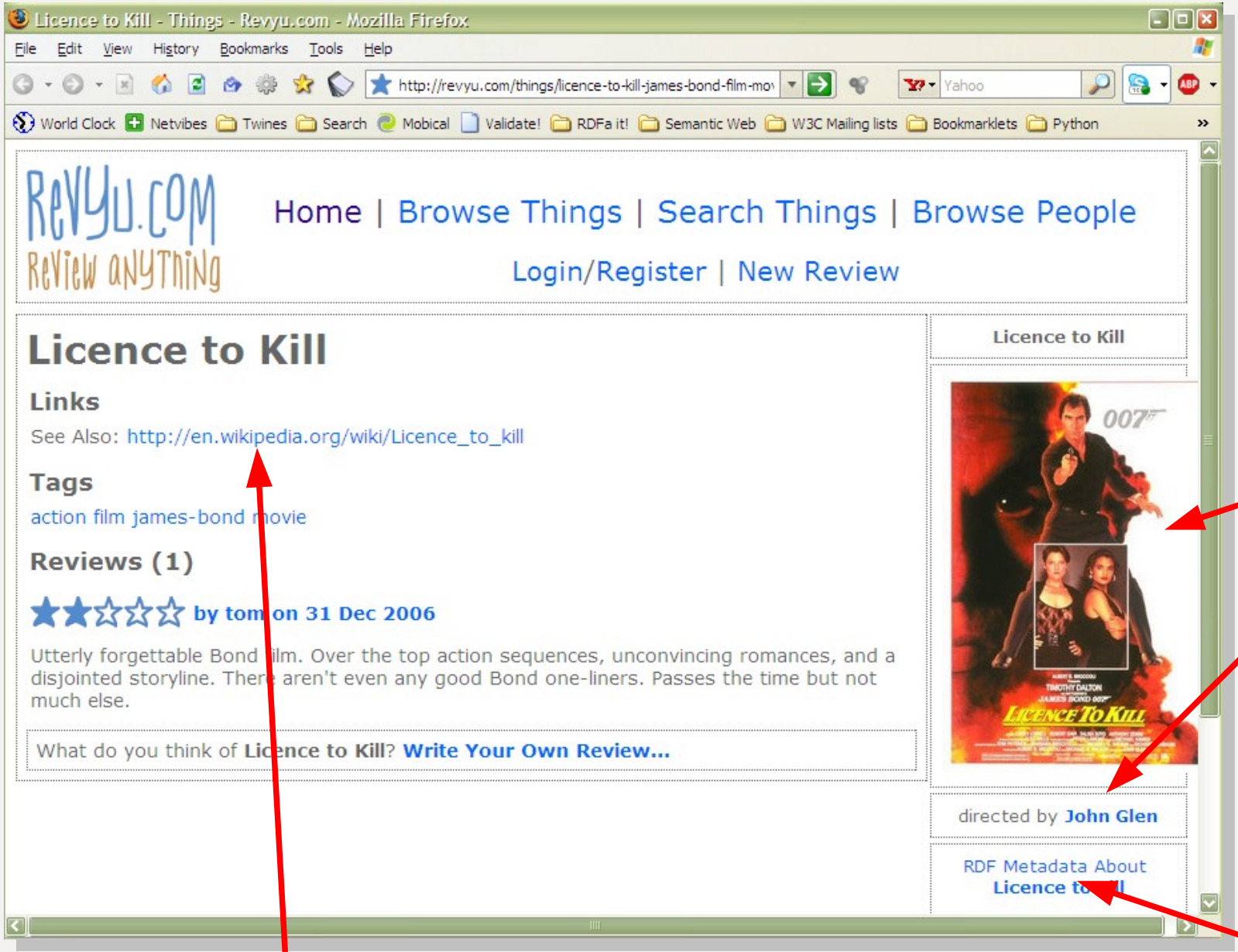
```
<http://sws.geonames.org/1275004/>
  owl:sameAs <http://DBpedia.org/resource/Kolkata>
  wgs84_pos:lat "22.5697222";
  wgs84_pos:long "88.3697222";
  sws:population "4631392"
 ...
```

Processors can switch automatically from one to the other…

The importance of the LOD project

- It creates a basis for large scale, Web-based data integration
- Applications begin to appear that use this dataset
- Other large datasets are continuously added. Some possible candidates:
  - data on movies, published music
  - various library catalogues
  - BBC programmes
  - etc.

“Review Anything”



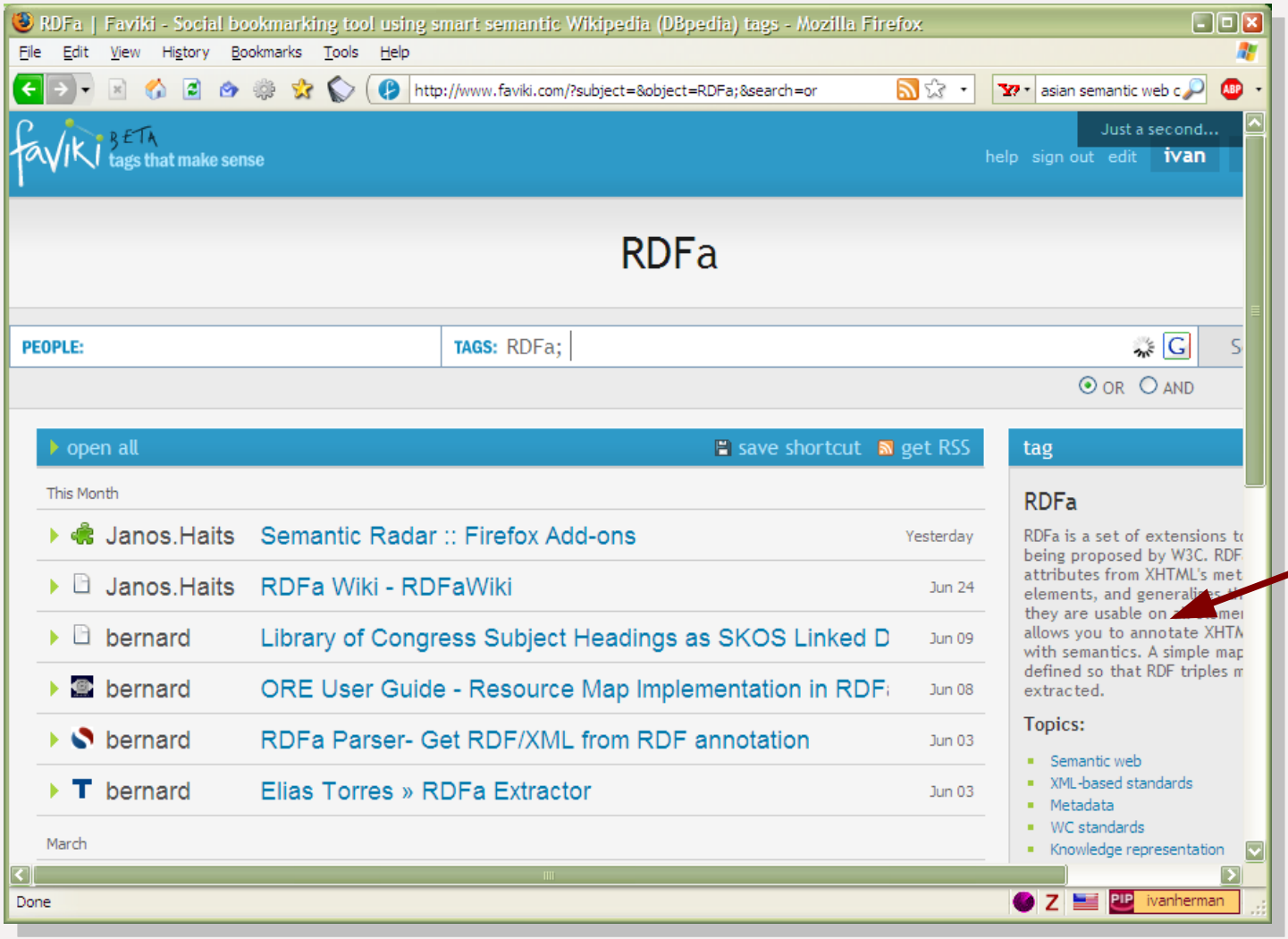
enhance output with linked data

data in RDF

links to, eg, (DB/Wiki)Pedia

Faviki: social bookmarking with Wiki tagging

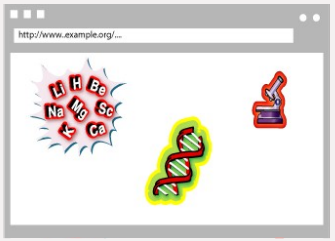
- Tag bookmarks via Wikipedia terms
- Use DBpedia URI-s and extra information

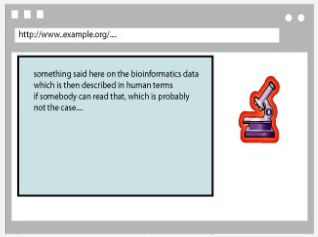


Extra info from DBpedia

What have we achieved?

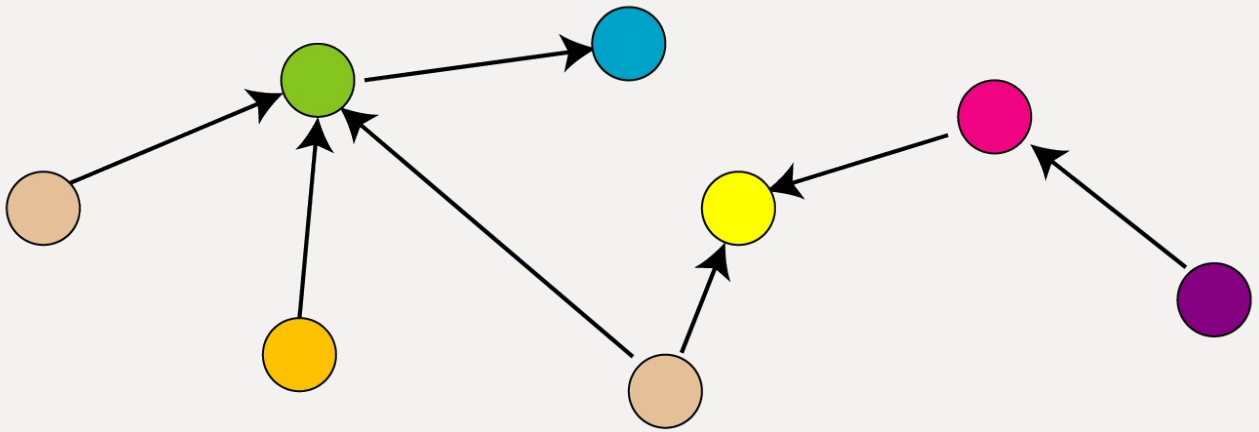
Remember the integration example?





Applications

Query, Manipulate, etc.

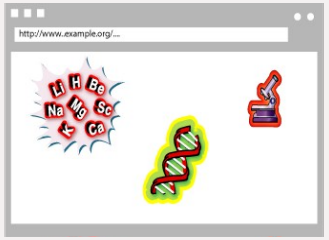


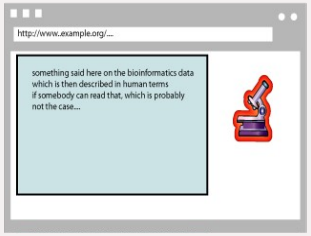
Data represented in abstract format

Map, Expose, etc.

Data in various formats

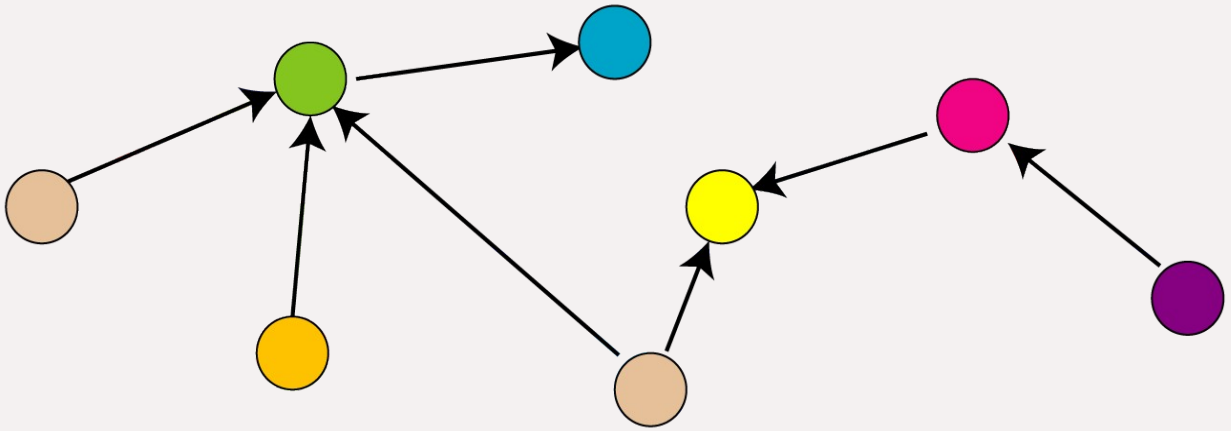
Same with what we learned





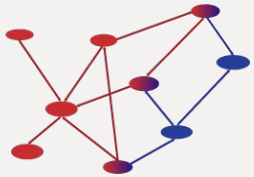
Applications

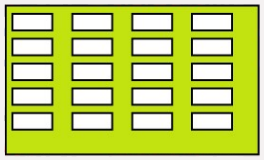
SPARQL,
OWL inferences,
etc.

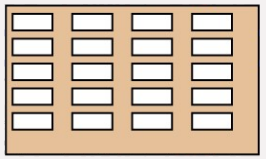


Data represented in RDF, possibly with extra knowledge (RDFS, OWL, SKOS, Rules, …)

SQL <=> RDF,
GRDDL, RDFa
etc.







Data in various formats

Available documents, tools

“Core” vocabularies

- There are also a number “core vocabularies” (not necessarily OWL based)
  - Dublin Core: about information resources, digital libraries, with extensions for rights, permissions, digital right management
  - FOAF: about people and their organizations
  - DOAP: on the descriptions of software projects
  - SIOC: Semantically-Interlinked Online Communities
  - vCard in RDF
  - …
- One should never forget: ontologies/vocabularies must be shared and reused!

Some books

- G. Antoniu and F. van Harmelen: Semantic Web Primer, 2nd edition in 2008
- D. Allemang and J. Hendler: Semantic Web for the Working Ontologist, 2008
- …

See the separate Wiki page collecting book references

Further information

- Dave Beckett's Resources at Bristol University
  - *huge* list of documents, publications, tools, …
- Planet RDF aggregates a number of SW blogs
- Semantic Web Interest Group
  - a forum developers with archived (and public) mailing list, and a constant IRC presence on freenode.net#swig
  - anybody can sign up on the list

Lots of Tools (*not* an exhaustive list!)

- Categories:
  - Triple Stores
  - Inference engines
  - Converters
  - Search engines
  - Middleware
  - CMS
  - Semantic Web browsers
  - Development environments
  - Semantic Wikis
  - …
- Some names:
  - Jena, AllegroGraph, Mulgara, Sesame, flickurl, …
  - TopBraid Suite, Virtuoso environment, Falcon, Drupal 7, Redland, Pellet, …
  - Disco, Oracle 11g, RacerPro, IODT, Ontobroker, OWLIM, Tallis Platform, …
  - RDF Gateway, RDFLib, Open Anzo, DartGrid, Zitgist, Ontotext, Protégé, …
  - Thetus publisher, SemanticWorks, SWI-Prolog, RDFStore…
  - …

Tools

- Worth noting: major companies offer (or will offer) Semantic Web tools or systems using Semantic Web: Adobe, Oracle, IBM, HP, Software AG, webMethods, Northrop Gruman, Altova, Dow Jones, BBN, …
- See also the W3C Wiki page on tools

Applications

- We have no time to look at more applications…
- There is a collection of Use Cases and Case Studies on the W3C site:
  - examples coming from: BT, Vodafone, Agfa, Sun, UK Ordnance Survey, NASA, Eli Lilly, Tata, Elsevier, UN FAO, Oracle, …
  - see http://www.w3.org/2001/sw/sweo/public/UseCases/

Thank you for your attention!

- These slides are publicly available on:

http://www.w3.org/2008/Talks/0822-Ghent-IH/