



Data Integration on Semantic Web

\$Date: 2006/07/27 12:27:50 \$

Ivan Herman, W3C

Introduction

This is just a short example on how the Semantic Web technologies can be used for data integration

The example is, of course, artificial and simplified

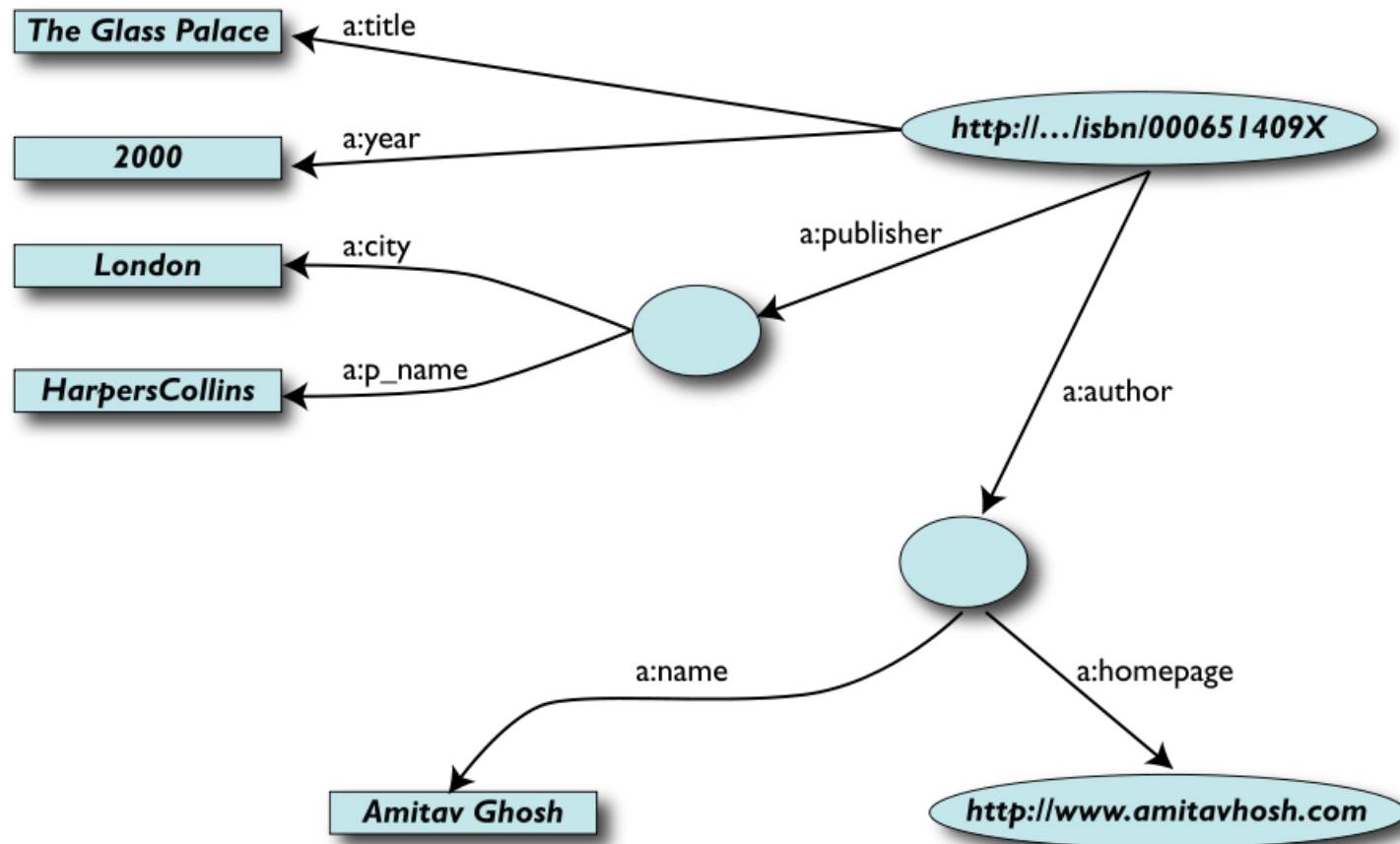
A simplified bookstore data (dataset “A”)

ID	Author	Title	Publisher	Year
ISBN 0-00-651409-X	id_xyz	The Glass Palace	id_qpr	2000

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

ID	Publisher Name	City
id_qpr	Harper Collins	London

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



Some notes on the data export

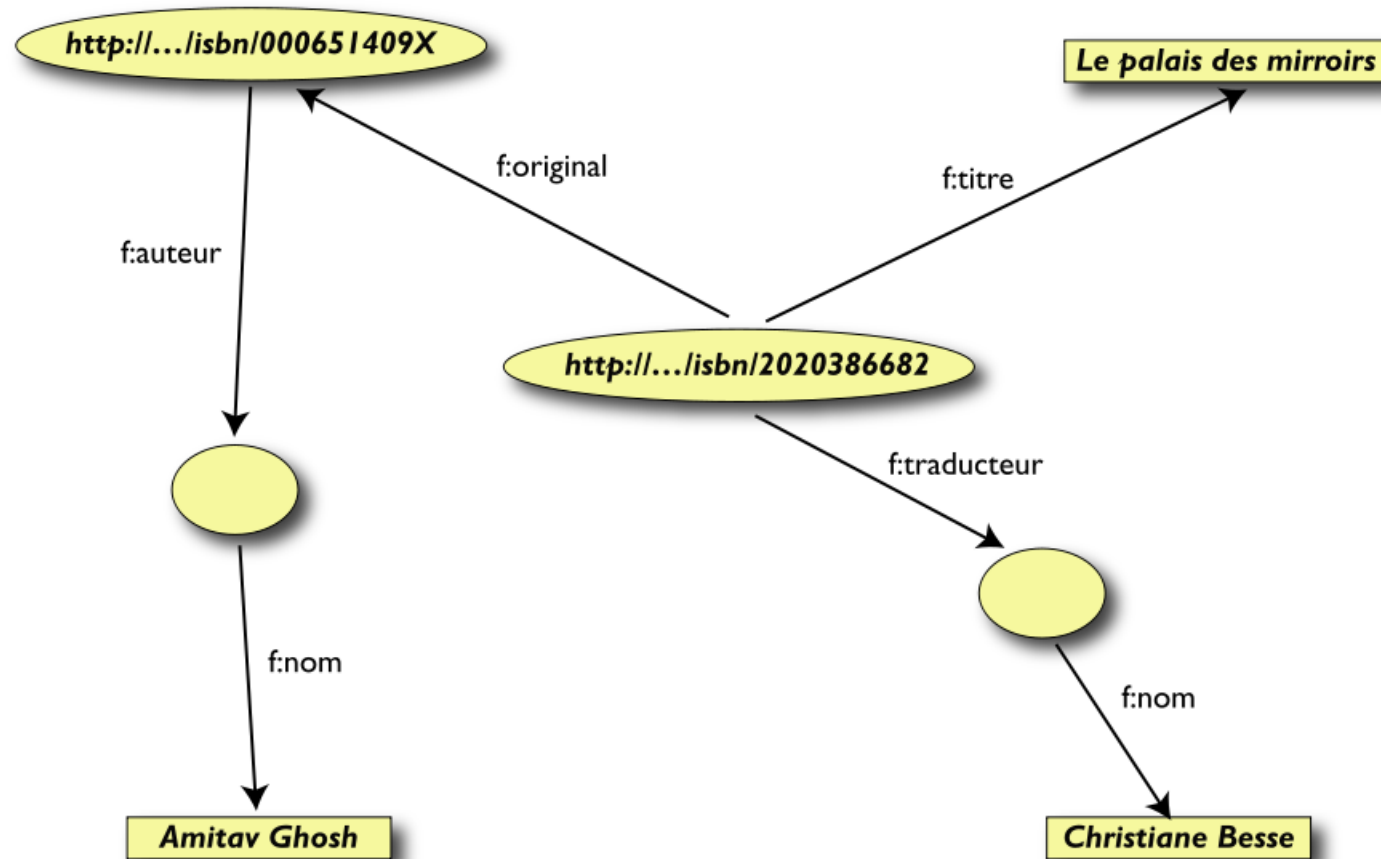
- Data export does not necessarily mean physical conversion of the data
 - *relations can be generated on-the-fly at query time*
 - via SQL “bridges”
 - scraping HTML pages
 - extracting data from Excel sheets
 - etc.
- One can export *part* of the data

Another bookstore data (dataset “F”)

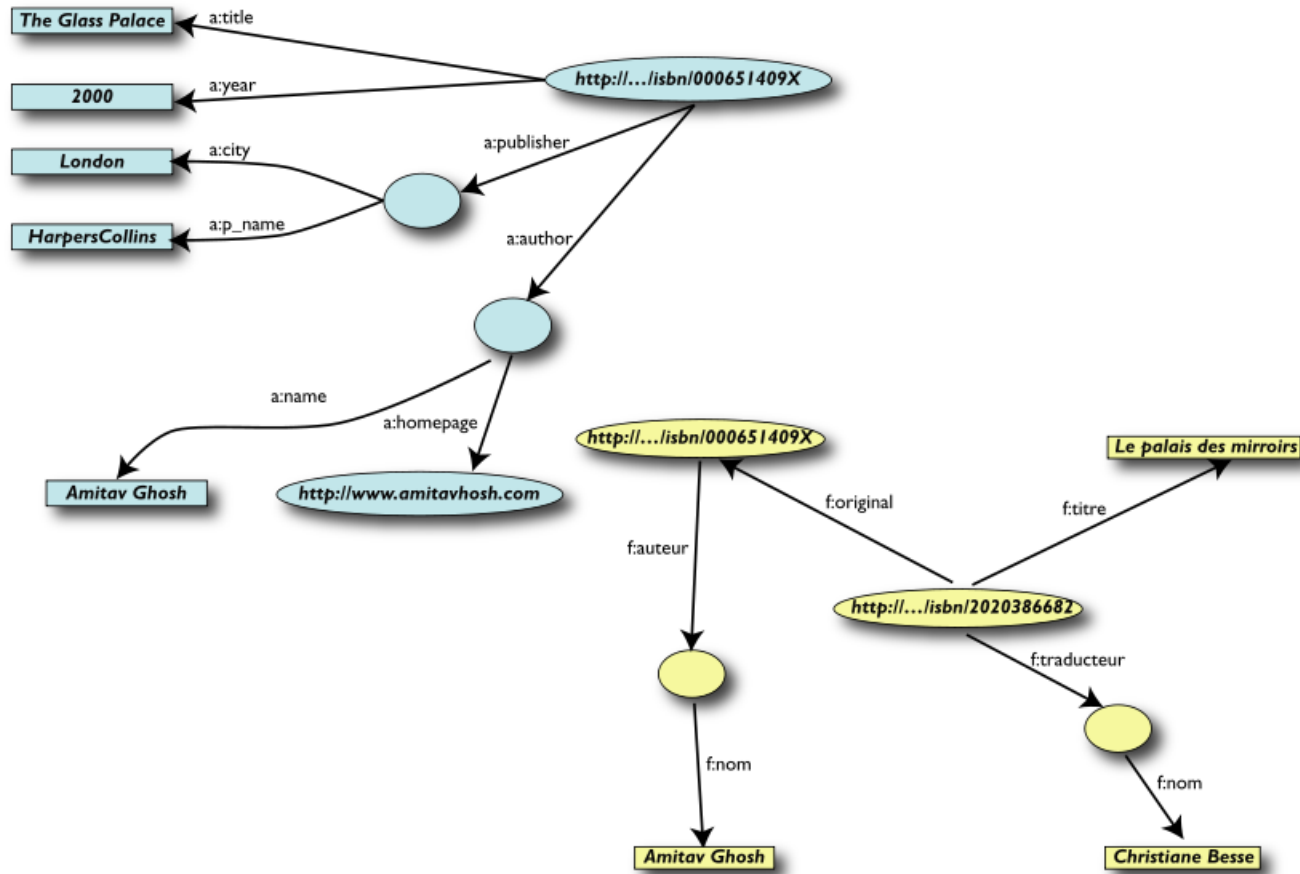
ID	Titre	Auteur	Traducteur	Original
ISBN 2020386682	Le Palais des miroirs	i_abc	i_qrs	ISBN 0-00-651409-X

ID	Nom
i_abc	Amitav Ghosh
i_qrs	Christiane Besse

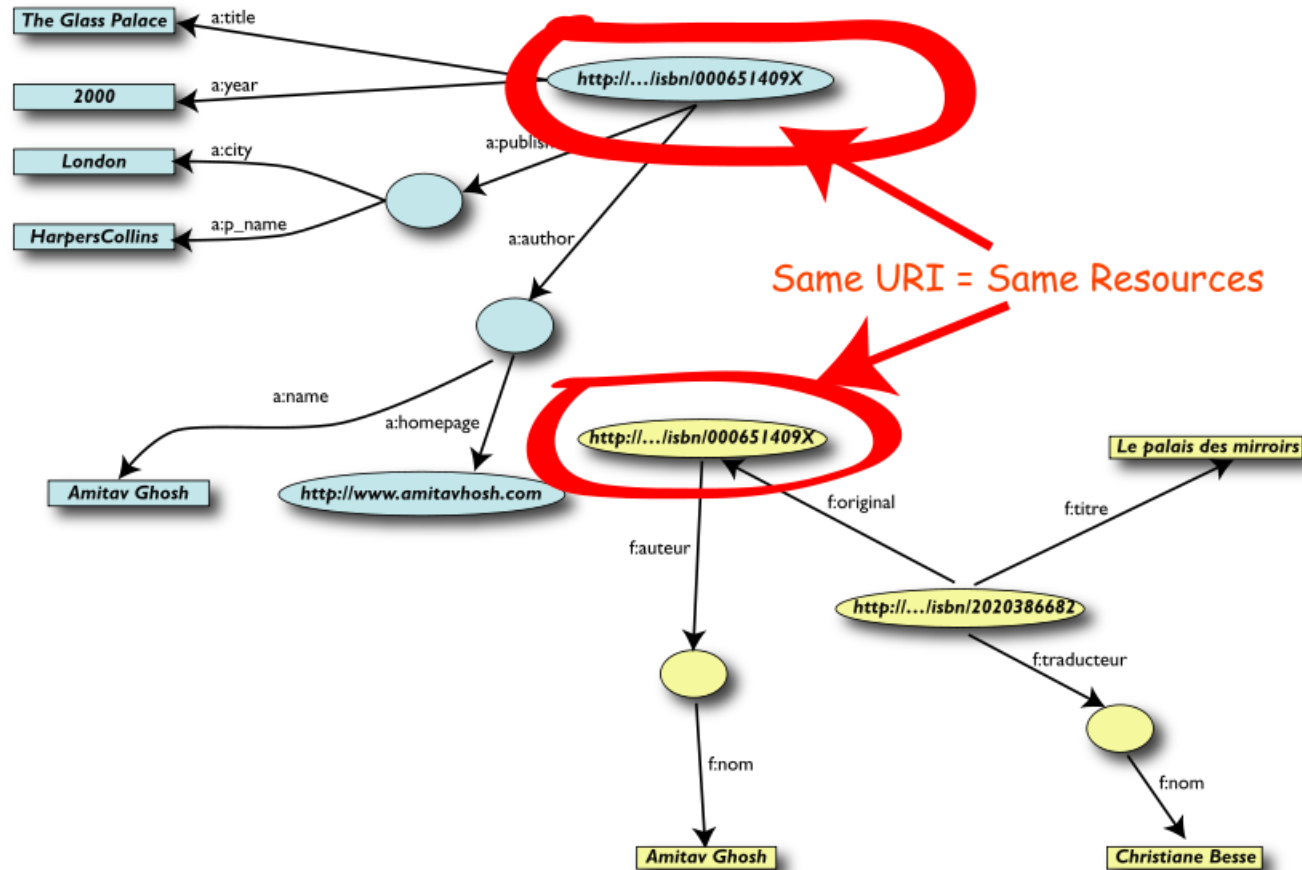
2nd step: export your second data into RDF



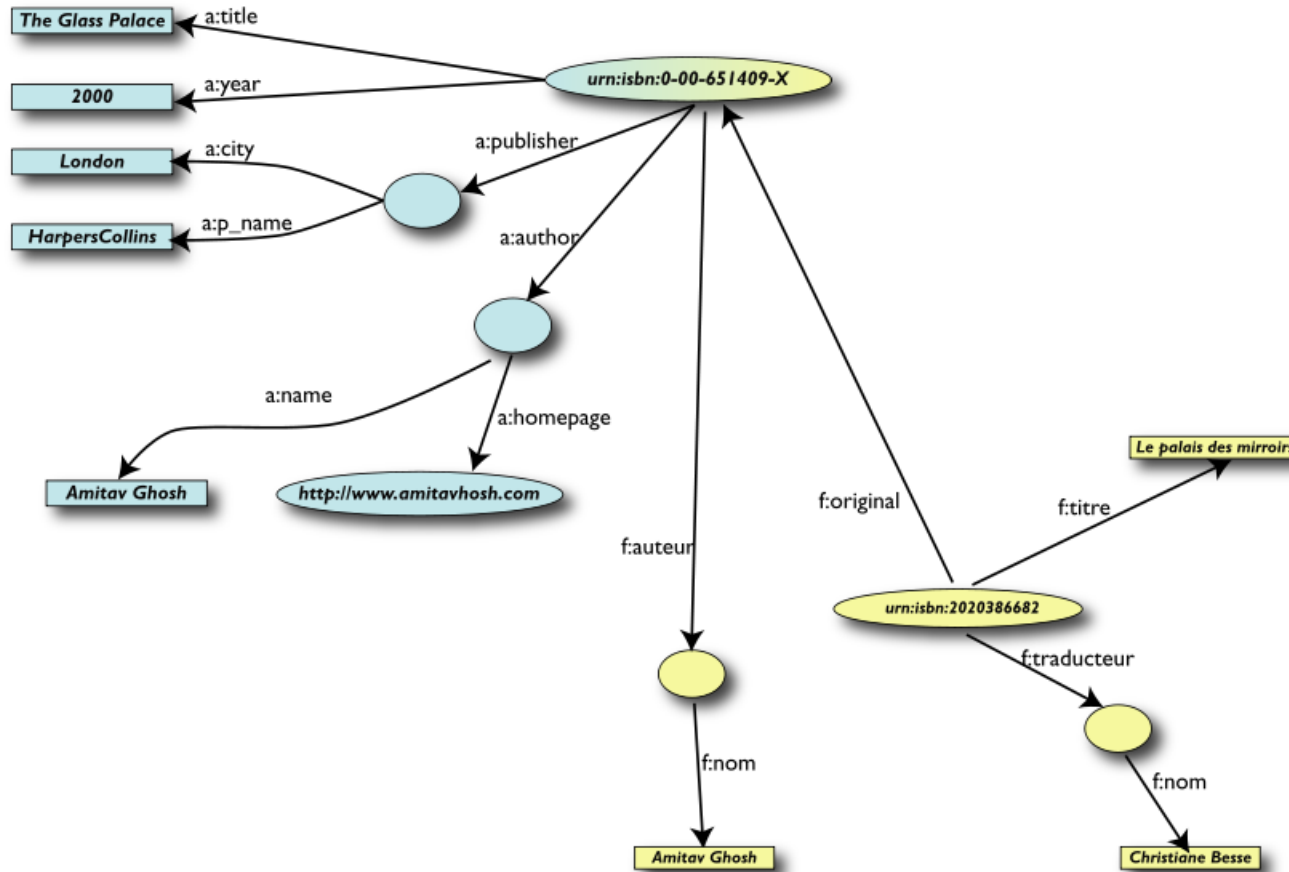
3rd step: start merging your data



3rd step: start merging your data (cont.)



3rd step: merge identical resources



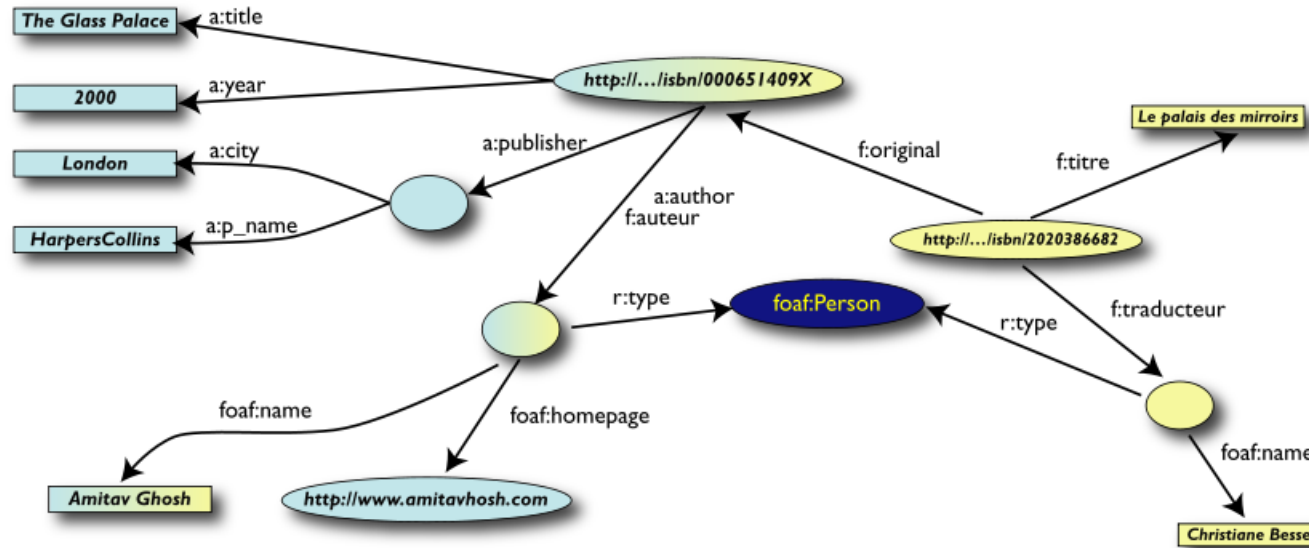
Start making queries...

- User of data “F” can now ask queries like:
 - « *donnes-moi le titre de l'original* »
 - (*ie: “give me the title of the original”*)
- This information is not in the dataset “F”...
- ...but can be automatically retrieved by merging with dataset “A”!

However, things are not complete yet...

- 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 has 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 step revisited: use the extra knowledge



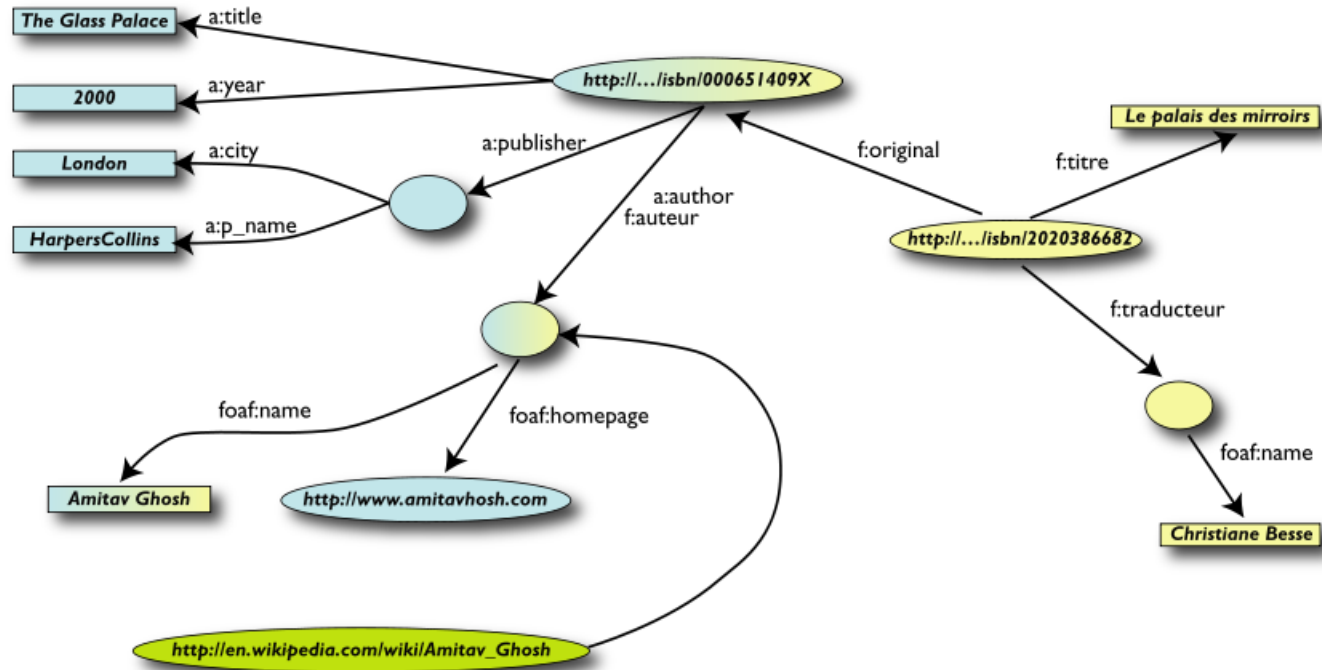
Start making richer queries!

- User of dataset “F” can now query:
 - *« donnes-moi la page d'accueil de l'auteur de l'original »*
 - *(ie, “give me the home page of the original’s author)*
- The data is not in dataset “F” ...
- ...but was made available by:
 - *merging datasets “A” and datasets “F”*
 - *adding three simple extra statements as an extra “glue”*
 - *using existing terminologies as part of the “glue”*

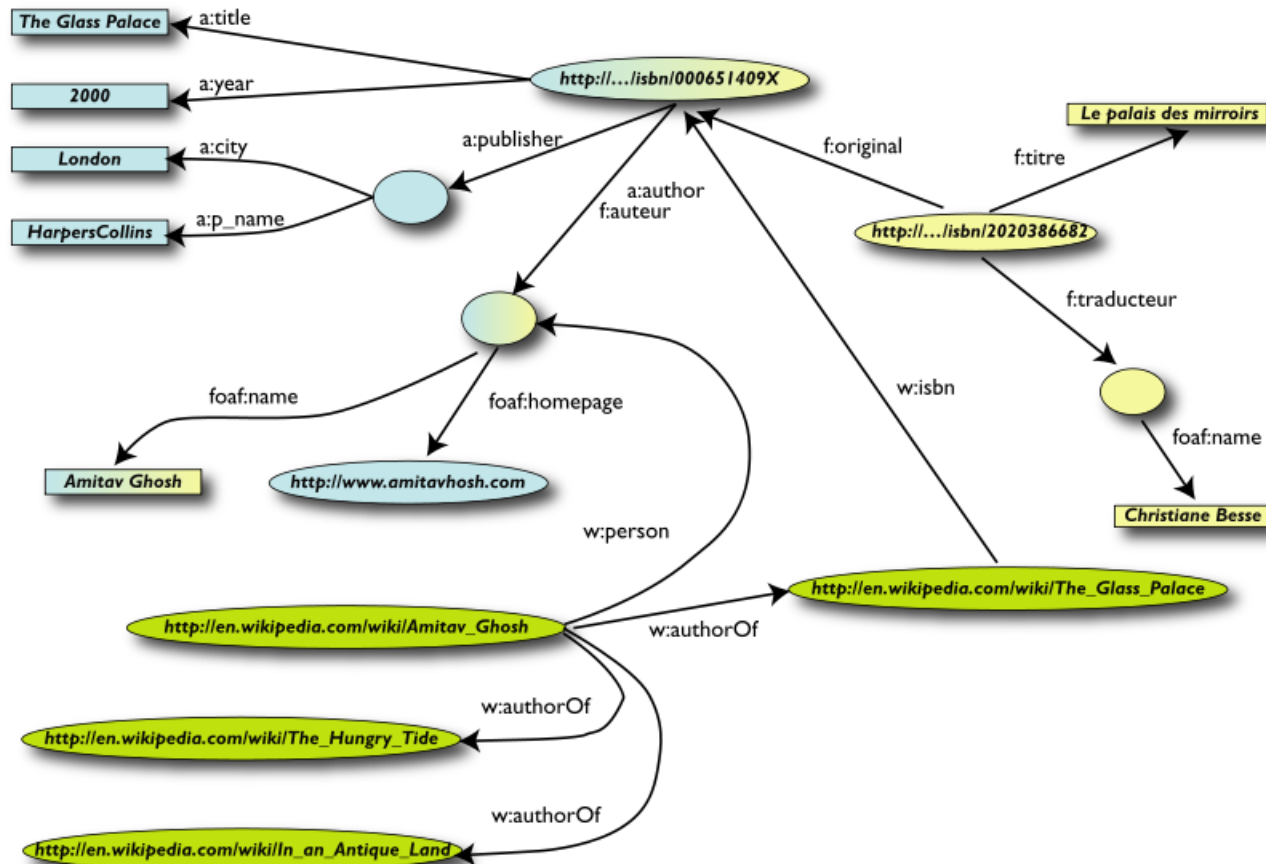
Combine with different datasets

- Using, e.g., the “Person”, the dataset can be combined with other sources
- For example, the data in Wikipedia can be extracted using simple (e.g., XSLT) tools
 - *there is an active development to add some simple semantic “tag” to wikipedia entries*
 - *we tacitly presuppose their existence in our example...*

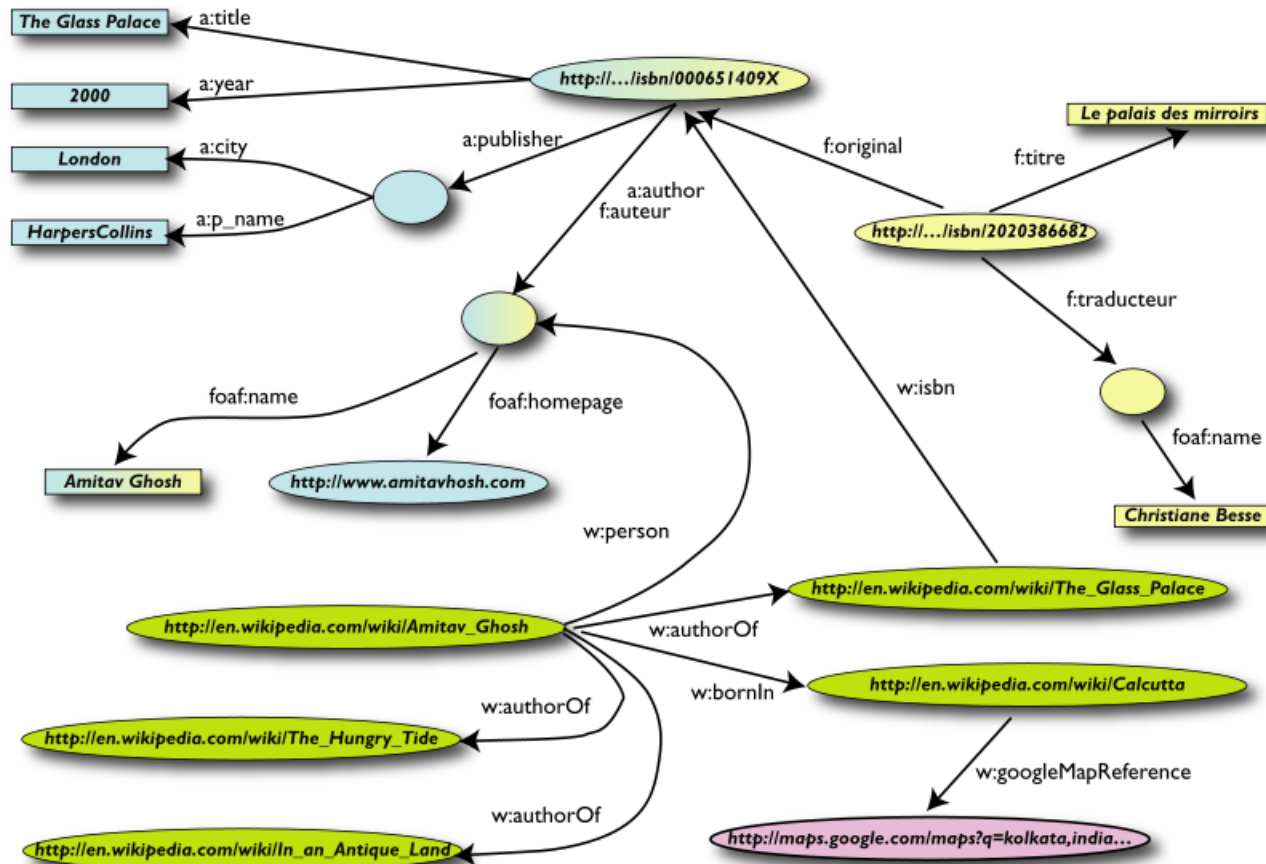
Merge with Wikipedia data



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 rigor (e.g., *naming* the relationships) is necessary so that machines could do this, too

What did we do?

- We combined different datasets
 - *all may be of different origin somewhere on the web*
 - *all may have different formats (mysql, excel sheet, XHTML, etc)*
 - *all may have different names for relations (e.g., multilingual)*
- 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”), also using common terminologies that a community has produced
- As a result, new relations could be found and retrieved

So where is the Semantic Web?

- The Semantic Web provides the technologies to make such integration possible!

For example:

- *an abstract model for the relational graphs: **RDF***
- *means to extract RDF information from, eg, XHTML pages: **GRDDL***
- *a query language adapted for the relational graphs: **SPARQL***
- *various technologies to characterize the relationships, categorize resources: **RDFS** (RDF Schemas), **OWL** (Web Ontology Language), **SKOS***
 - depending on the complexity required, applications may choose among the different technologies
- *reuse of existing “ontologies” that others have produced (FOAF in our case)*

A real life example: Antibodies Demo

- Scenario: find the known antibodies for a protein in a specific species
- Combine four different data sources
- Use SPARQL as an integration tool

Antibodies RDF Demo

The demo's purpose is to demonstrate the power of SPARQL against distributed life-sciences data sources on the web. This demo's scenario revolves around a researcher searching the NCBI's Entrez Protein database, identifying a protein of interest from the returned results, and then searching for antibodies against that target protein. This demo uses SPARQL to query over these data sources:

- Entrez Protein
- Alzheimer Research Forum Antibody Database
- Wikispecies directory of species

bc10

Protein (NCBI)	Antibody (AlzForum)
<p>NP_069912 (NCBI) B-cell CLL/lymphoma 10 Homo sapiens</p>	<p>Bcl-10 (AlzForum) Distributor: BD Pharmingen (cat. no. 551340) Immunogen: Specificity: 31 kDa Bcl-10</p>
<p>NP_776216 (NCBI) mucosa associated lymphoid tissue lymphoma translocation protein 1 isoform b Homo sapiens</p>	<p>Bcl-10 (AlzForum) Distributor: exalpha Biologicals (cat. no. X1119P) Immunogen: synthetic peptide corr. to aa. 5-19 of human bcl-10, N-term Specificity: Bcl-10</p>
<p>NP_006776 (NCBI) mucosa associated lymphoid tissue lymphoma translocation protein 1 isoform a Homo sapiens</p>	<p>Bcl-10 (AlzForum) Distributor: Abcam (cat. no. AB1142) Immunogen: immunogen = synthetic peptide: EMFLPLRS RTVSRQC, human Specificity: Reacts with the C terminal sequence [EMFLPLRS RTVSRQC] of Bcl-10</p>