

Introduction to Semantic Web Technologies

Ivan Herman, W3C
June 22nd, 2010



The Music site of the BBC

BBC - Music - Eric Clapton

http://www.bbc.co.uk/music/artists/618b6900-0618-4f1e-b835-bccb17f84294

Netvibes Feedly Social Private Mailing lists SW Python RDFa it! Bookmarklets Add Zemanta bit.ly To Mendeley TinyURL To Faviki Dokuwiki


MUSIC BETA GENRES ARTISTS REVIEWS NEWS BLOG QUICK FIND Enter an artist name ...

BBC Music > Artists > Eric Clapton

Eric Clapton

Born 30 March 1945.

MOST PLAYED ON BBC RADIO 2




David Redfern/Redferns

Latest Tracks Played On The BBC

Promises BBC Radio 2 Ken Bruce 22/02/2010
Bad Love BBC Radio 2 Alex Lester 22/02/2010
Lay Down Sally BBC Radio 2 Chris Evans Breakfast 18/02/2010
I Ain't Gonna Stand For It BBC Radio 2 Alex Lester 15/02/2010
Wonderful Tonight BBC Radio 2 Ken Bruce 10/02/2010

Audio Previews From Latest Album Review

	Me And Mr Johnson
8	Milkcow's Calf Blues
10	Come on in My Kitchen

Biography

Eric Patrick Clapton, CBE (born 30 March 1945) is an English blues-rock guitarist, singer, songwriter and composer. Clapton has been inducted into the Rock and Roll Hall of Fame as a solo performer, as a member of rock bands; the Yardbirds and Cream. Clapton is the only person ever to be inducted three times. Often viewed by critics and fans alike as one of the most important and influential guitarists of all time, Clapton was ranked fourth in Rolling Stone magazine's list of the "100 Greatest Guitarists of All Time" and #53 on their list of the Immortals: 100 Greatest Artists of All Time.

The Music site of the BBC

BBC - Music - Eric Clapton

http://www.bbc.co.uk/music/artists/618b6900-0618-4f1e-b835-bccb17f84294 BBC Music

Netvibes Feedly Social Private Mailing lists SW Python RDFa it! Bookmarklets Add Zemanta bit.ly To Mendeley TinyURL To Faviki Dokuwiki

this focus, he is credited as an innovator in a wide variety of genres. These include blues-rock (with John Mayall & the Bluesbreakers and The Yardbirds) and psychedelic rock (with Cream). Clapton's chart success was not limited to the blues, with chart-toppers in Delta Blues (Me and Mr. Johnson), pop ("Change the World") and reggae (Bob Marley's "I Shot the Sheriff") (He is often credited for bringing reggae and Bob Marley to the mainstream.) Two of his most successful recordings were the hit love song "Layla", which he played with the band Derek and the Dominos, and Robert Johnson's "Crossroads", which has been his staple song since his days with Cream.

[Read more at Wikipedia...](#)

WIKIPEDIA This entry is from Wikipedia, the user-contributed encyclopedia. It may not have been reviewed by professional editors and is licensed under the GNU Free Documentation License. If you find the biography content factually incorrect, defamatory or highly offensive you can [edit this article at Wikipedia](#). Find out more about our use of this data.

Links & Information

LINKS

- Official homepage at ericclapton.com
- Fanpage at whereseric.com
- Wikipedia article on [Eric Clapton](#)
- MySpace at myspace.com/ericclapton
- Last.fm page on [Eric Clapton](#)
- MusicBrainz entry on [Eric Clapton](#)

MEMBER OF [Derek and the Dominos](#), [Blind Faith](#) (1968-1969), [Cream](#) (1966-1968), [John Mayall & The Bluesbreakers](#) (1965-1966), [The Yardbirds](#) (1963-1965)








COLLABORATED ON [J.J. Cale & Eric Clapton](#), [Eric Clapton & The Immediate All Stars](#), [Eric Clapton & The Impressions](#), [Eric Clapton & Jimmy Page](#), [Eric Clapton & David Sanborn](#), [Eric Clapton & Stan Webb's Chicken Shack](#), [Eric Clapton & The Powerhouse](#), [Eric Clapton & Stevie Ray Vaughan](#), [Eric Clapton & Marc Shaiman](#), [The Dirty Mac](#), [Bob Dylan, Roger McGuinn, Tom Petty, Neil Young, Eric Clapton & George Harrison](#), [Jimmie Vaughan](#), [Eric Clapton, Bonnie Raitt, Robert Cray, B.B. King, Buddy Guy, Dr. John & Art Neville](#), [Elton John & Eric Clapton](#), [Michael Kamen](#), [Eric Clapton and David Sanborn](#), [B.B. King & Eric Clapton](#), [Mark Knopfler & Eric Clapton](#), [Paul McCartney & Eric Clapton](#), [Sting with Eric Clapton](#), [Steve Winwood & Eric Clapton](#)

Links & information come from [MusicBrainz](#). You can add or edit information about [Eric Clapton at musicbrainz.org](#). Find out more about our use of this data. The BBC is not responsible for the content of external sites

Latest News Stories

Played By

Since December 2008

-  **Alex Lester**
2 BBC Radio 2
-  **Steve Wright in the Afternoon**
2 BBC Radio 2
-  **Sarah Kennedy**
2 BBC Radio 2
-  **Wake Up to Wogan**
2 BBC Radio 2
-  **Ken Bruce**
2 BBC Radio 2
-  **Steve Wright's Sunday Love Songs**
2 BBC Radio 2
-  **Jeremy Vine**
2 BBC Radio 2

Information displayed about artists played on BBC programmes is incomplete
[out more about this artist play count information.](#)

How to build such a site 1.

- ▶ Site editors roam the Web for new facts
 - ▶ may discover further links while roaming
- ▶ They update the site manually
- ▶ And the site gets soon out-of-date 😞

How to build such a site 2.

- ▶ Editors roam the Web for new data published on Web sites
- ▶ “Scrape” the sites with a program to extract the information
 - ▶ ie, write some code to incorporate the new data
- ▶ Easily get out of date again... 😞

How to build such a site 3.

- ▶ Editors roam the Web for new data via API-s
- ▶ Understand those...
 - ▶ input, output arguments, datatypes used, etc
- ▶ Write some code to incorporate the new data
- ▶ Easily get out of date again... 😞

The choice of the BBC

- ▶ Use external, public datasets
 - ▶ Wikipedia, MusicBrainz, ...
- ▶ They are available as data
 - ▶ not API-s or hidden on a Web site
 - ▶ data can be extracted using, eg, HTTP requests or standard queries

In short...

- ▶ Use the Web of Data as a Content Management System
- ▶ Use the community at large as content editors

And this is no secret...

BBC - Music - Eric Clapton

http://www.bbc.co.uk/music/artists/618b6900-0618-4f1e-b835-bccb17f84294

Netvibes Feedly Social Private Mailing lists SW Python RDFa it! Bookmarklets Add Zemanta bit.ly To Mendeley TinyURL To Faviki Dokuwiki

MusicBrainz entry on [Eric Clapton](#)

MEMBER OF [Derek and the Dominos](#), [Blind Faith](#) (1968-1969), [Cream](#) (1966-1968), [John Mayall & The Bluesbreakers](#) (1966), [The Yardbirds](#) (1963-1965)

COLLABORATED ON [J.J. Cale & Eric Clapton](#), [Eric Clapton & The Immediate All Stars](#), [Eric Clapton & The Impressions](#), [Clapton & Jimmy Page](#), [Eric Clapton & David Sanborn](#), [Eric Clapton & Stan Webb's Chicken Shack](#), [Eric Clapton & The Powerhouse](#), [Eric Clapton & Stevie Ray Vaughan](#), [Eric Clapton & Marc Shaiman](#), [The Dirty Mac](#), [Bob Dylan, Roger Daltrey & Eric Clapton](#), [Tom Petty, Neil Young, Eric Clapton & George Harrison](#), [Jimmie Vaughan](#), [Eric Clapton, Bonnie Raitt, Robert Cray, B.B. King, Buddy Guy, Dr. John & Art Neville](#), [Elton John & Eric Clapton](#), [Michael Kamen, Eric Clapton and David Sanborn](#), [B.B. King, Eric Clapton, Mark Knopfler & Eric Clapton](#), [Paul McCartney & Eric Clapton](#), [Sting with Eric Clapton](#), [Steve Winwood & Eric Clapton](#)

Links & information come from [MusicBrainz](#). You can add or edit information about [Eric Clapton at musicbrainz.org](#). [Find out more about this data](#). The BBC is not responsible for the content of external sites

Latest News Stories

NEWS FROM THE BBC

Clapton recovering after surgery
Tue 27 Oct 2009 17:46 Musician Eric Clapton is recuperating after having an operation to remove gallstones,...

Clapton to use a slowhand to take Wyman's wicket

Data on the Web

- ▶ There are more and more data on the Web
 - ▶ government data, health related data, general knowledge, company information, flight information, restaurants,...
- ▶ More and more applications rely on the availability of that data

But... data are often in isolation, “silos”



Photo credit Alex (ajagendorf25), Flickr

Imagine...

- ▶ A “Web” where
 - ▶ documents are available for download on the Internet
 - ▶ but there would be no hyperlinks among them

And the problem is real...

The image displays three overlapping web browser windows, illustrating the complexity of integrating data from different sources.

- CoCoDat - Collation of Cortical Data - Mozilla Firefox**: The top-left window shows the CoCoDat homepage. It describes the database as a microcircuitry database that collates published experimental reports. It lists available data types: Morphology, Firing properties, Ionic currents, Ionic conductances, Synaptic currents, and Connectivity. It also mentions that the database is available for download as data tables or through a Search Board.
- Cell Centered Database - Mozilla Firefox**: The top-right window shows the Cell Centered Database Gallery. It is part of the National Center for Microscopy and Imaging Research. The page features a navigation bar with links to Data, Search, Gallery, Dictionary, Publications, MyCCDB, Data Download, Contact us, and Help. Below the navigation bar are tabs for 2D image, Reconstruction, Segmentation, and Animation.
- NeuronDB = Thalamic relay neuron - Overview (A) () - Mozilla Firefox**: The bottom window shows the NeuronDB overview page for a Thalamic relay neuron. It includes a navigation bar with links to Overview, Data/Search, plus Connectivity, plus Classical References/Notes, and Models. The page lists various properties and regions, such as Distal equivalent dendrite, Middle equivalent dendrite, Proximal equivalent dendrite, Soma, Axon hillock, Axon fiber, Axon terminal, and All Compartments. It also lists properties like Receptors, Channels, Transmitters, and All Properties. The page includes a diagram of the neuron and a list of regions with corresponding 'Show' and 'other' links.

Data on the Web is not enough...

- ▶ We need a proper infrastructure for a real Web of Data
 - ▶ data is available on the Web
 - ▶ accessible via standard Web technologies
 - ▶ data are interlinked over the Web
 - ▶ ie, data can be integrated over the Web
- ▶ This is where Semantic Web technologies come in

A Web of Data unleashes now applications

The screenshot shows the data.gov.uk website in a web browser. The browser's address bar displays 'http://data.gov.uk/'. The website header includes the HM Government logo and the text 'data.gov.uk'. A navigation bar contains links for Home, Blog, Data, SPARQL, Apps, Ideas, Forum, Wiki, Resources, and About. A search bar is located on the right side of the navigation bar. The main content area features a large banner with the text 'Unlocking innovation' and 'Working with UK Public Sector information and data', accompanied by a blue molecular structure graphic. Below the banner, there is a paragraph of text explaining the initiative, followed by a 'Search Data' section with a keyword input field and a 'Search' button. To the right of the search section is a 'Browse for Data' section with links for 'List all datasets' and 'Common tags'. At the bottom left, there is a 'Most Recent Apps' section with a 'View all apps' link. On the right side of the page, there is a 'User login' section with fields for 'Username' and 'Password', a 'Log In' button, and a link for 'Request new password'. Below the login section is a 'What is the Semantic Web?' section with a brief explanation and a 'Read more' button. The browser's toolbar shows various icons for navigation and search, and the page is titled 'Unlocking innovation | data.gov.uk'.

Unlocking innovation | data.gov.uk

http://data.gov.uk/

Netvibes Feedly Social Private Mailing lists SW Python RDFa it! Bookmarklets Add Zemanta bit.ly To Mendeley TinyURL To Faviki

HM Government data.gov.uk

Home Blog Data SPARQL Apps Ideas Forum Wiki Resources About

Unlocking innovation

Working with UK Public Sector information and data

Advised by Sir Tim Berners-Lee and Professor Nigel Shadbolt and others, government are opening up data for reuse. This site seeks to give a way into the wealth of government data and is under constant development. We want to work with you to make it better.

We're very aware that there are more people like you outside of government who have the skills and abilities to make wonderful things out of public data. These are our first steps in building a collaborative relationship with you.

Search Data

Enter keyword(s) **Search**

e.g. education, NHS, crime, transport, environment

Powered by: CKAN

Browse for Data

[List all datasets](#)

[Common tags](#)

User login

Username:

Password:

Log In

[Request new password](#)

What is the Semantic Web?

Combining different data sources has never been easy but the Semantic Web will enable data to be joined easily across boundaries.

Read more

Most Recent Apps [View all apps +](#)

A nice usage of UK government data

Windows browser window showing the Newspaper Club blog post.

Address bar: <http://blog.newspaperclub.co.uk/2009/10/16/data-gov-uk-newspaper/>

Navigation: Netvibes, Feedly, Social, Private, Mailing lists, SW, Python, RDFa it, Bookmarklets, Add Zemanta, bit.ly, To Mendeley, TinyURL, To Faviki

Newspaper Club

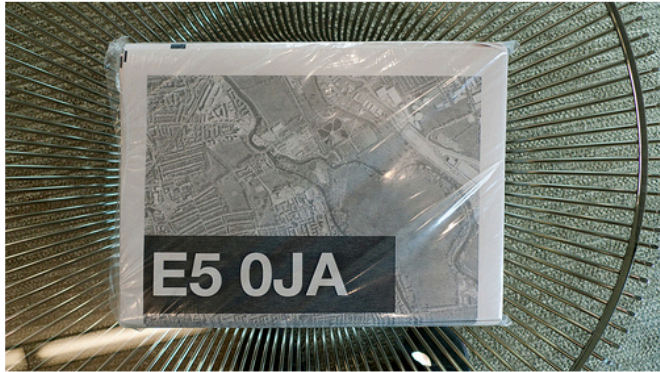
Data.gov.uk Newspaper

Friday, October 16th 2009

Over the last three days we've been working on a side project. A design exercise if you like.

We've been thinking about the beta [Data.gov.uk](#) repository, and wanted to explore putting some of the information contained within into people's hands in a form that is accessible, timely, and relevant.

And perhaps unsurprisingly, we thought a good way to do that was with a newspaper. So here it is, the Postcode Paper:



It's a prototype of a service for people moving into a new area. In our exercise we imagined you might receive it after paying your council tax for the first time.

It gathers information about your area, such as local services, environmental information and crime statistics.

This is a post by Tom from the **Newspaper Club** Blog. File under [case studies](#).

We're building a service to help people make their own newspapers. This is the blog where we're alarmingly honest about where it's all going wrong. And occasionally smug about where it's going right.

WE'RE IN ALPHA

You can stick your name on the beta invite list [here](#).

Search

[Subscribe via RSS](#)

Archives

- [January 2010](#)
- [December 2009](#)
- [November 2009](#)
- [October 2009](#)
- [September 2009](#)
- [August 2009](#)
- [July 2009](#)
- [June 2009](#)

Filed under

- [4ip](#) (3)
- [art](#) (8)
- [case studies](#) (9)
- [engineering](#) (5)
- [fulfillment etc](#) (2)
- [investors](#) (2)
- [media](#) (1)
- [printers](#) (6)
- [Uncategorized](#) (31)

Right sidebar images: A stack of papers, a person at a desk, a group of people, a whiteboard, a stack of papers, a stack of papers, a travel times sign.

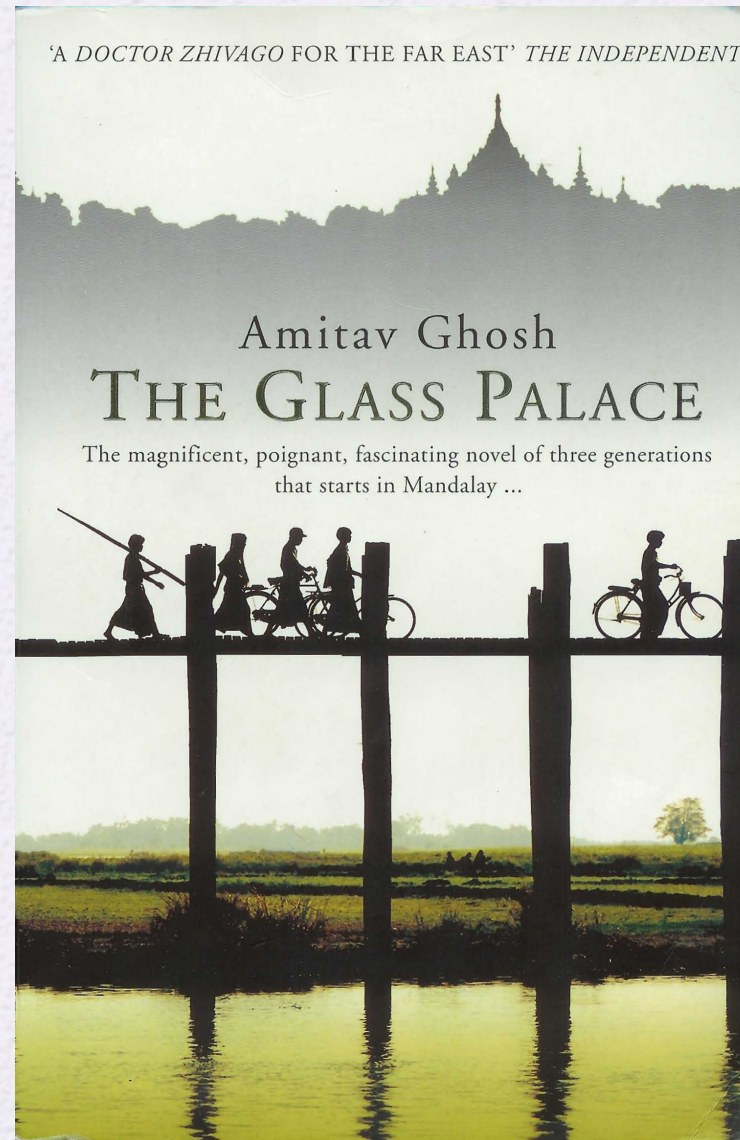
In what follows...

- ▶ We will use a simplistic example to introduce the main Semantic Web concepts

The rough structure of data integration

- ▶ Map the various data onto an abstract data representation
 - ▶ make the data independent of its internal representation...
- ▶ Merge the resulting representations
- ▶ Start making queries on the whole!
 - ▶ queries not possible on the individual data sets

We start with a book...



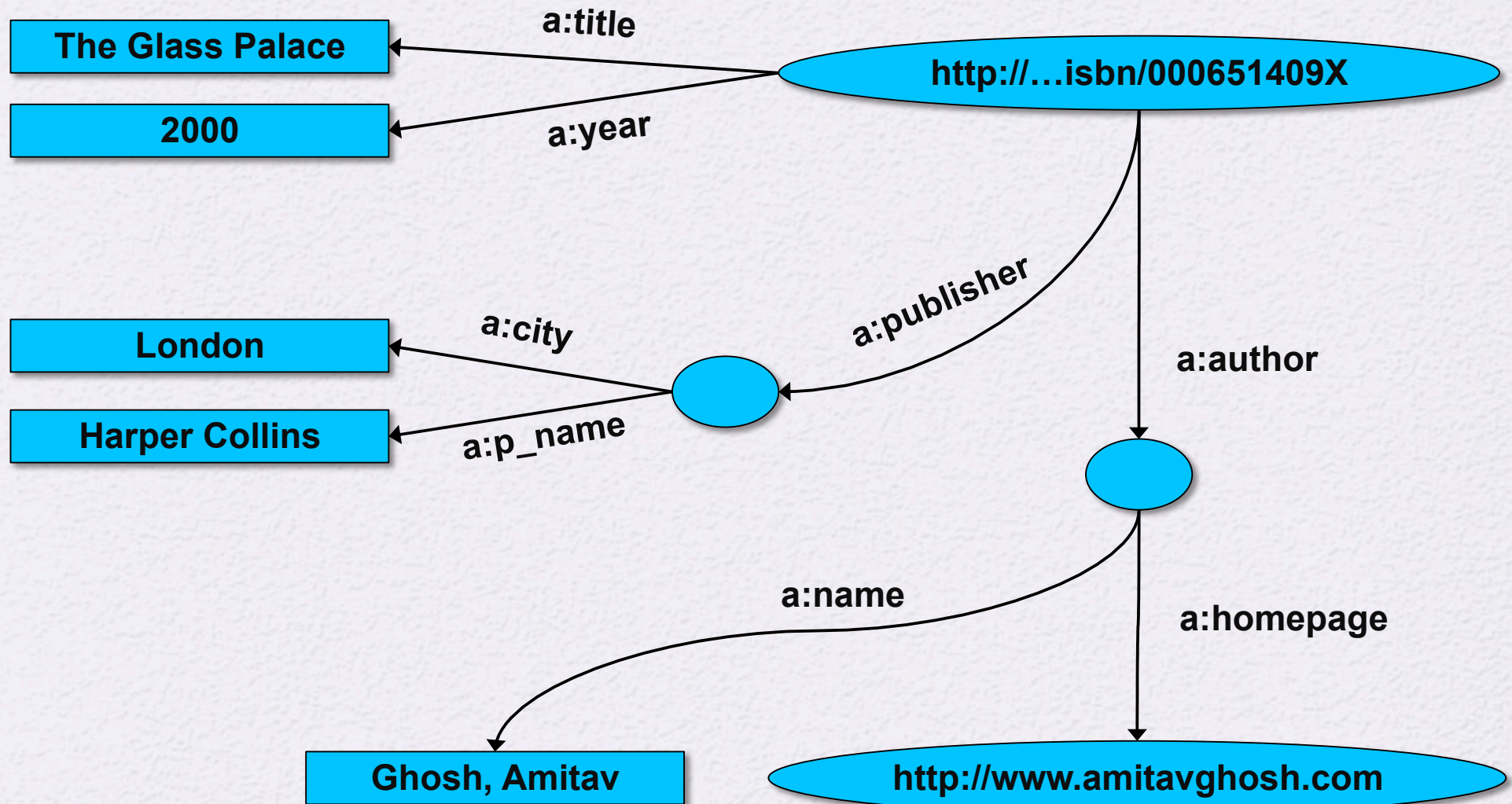
A simplified bookstore data (dataset “A”)

ID	Author	Title	Publisher	Year
ISBN 0-00-6511409-X	id_xyz	The Glass Palace	id_qpr	2000

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

ID	Publisher's name	City
id_qpr	Harper Collins	London

1st: export your data as a set of relations



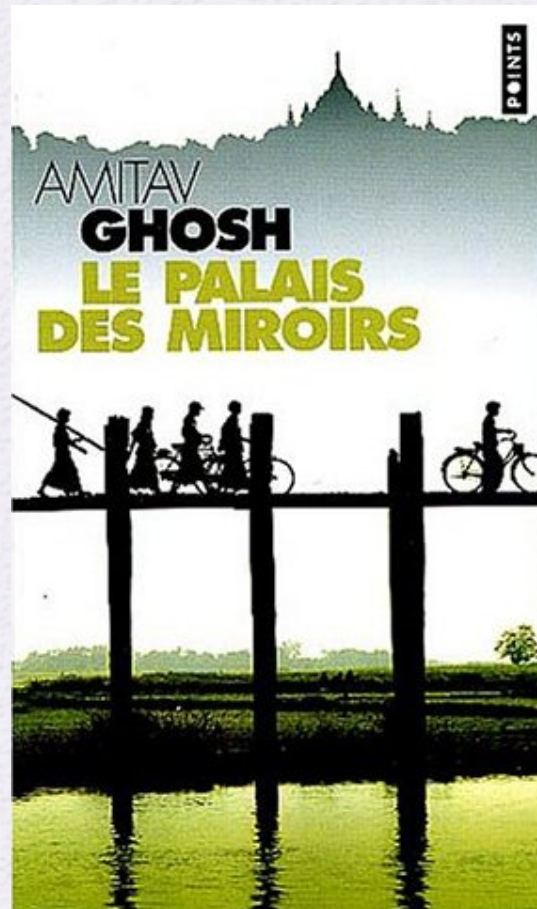
Some notes on the exporting the data

- ▶ 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

Some notes on the exporting the data

- ▶ 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

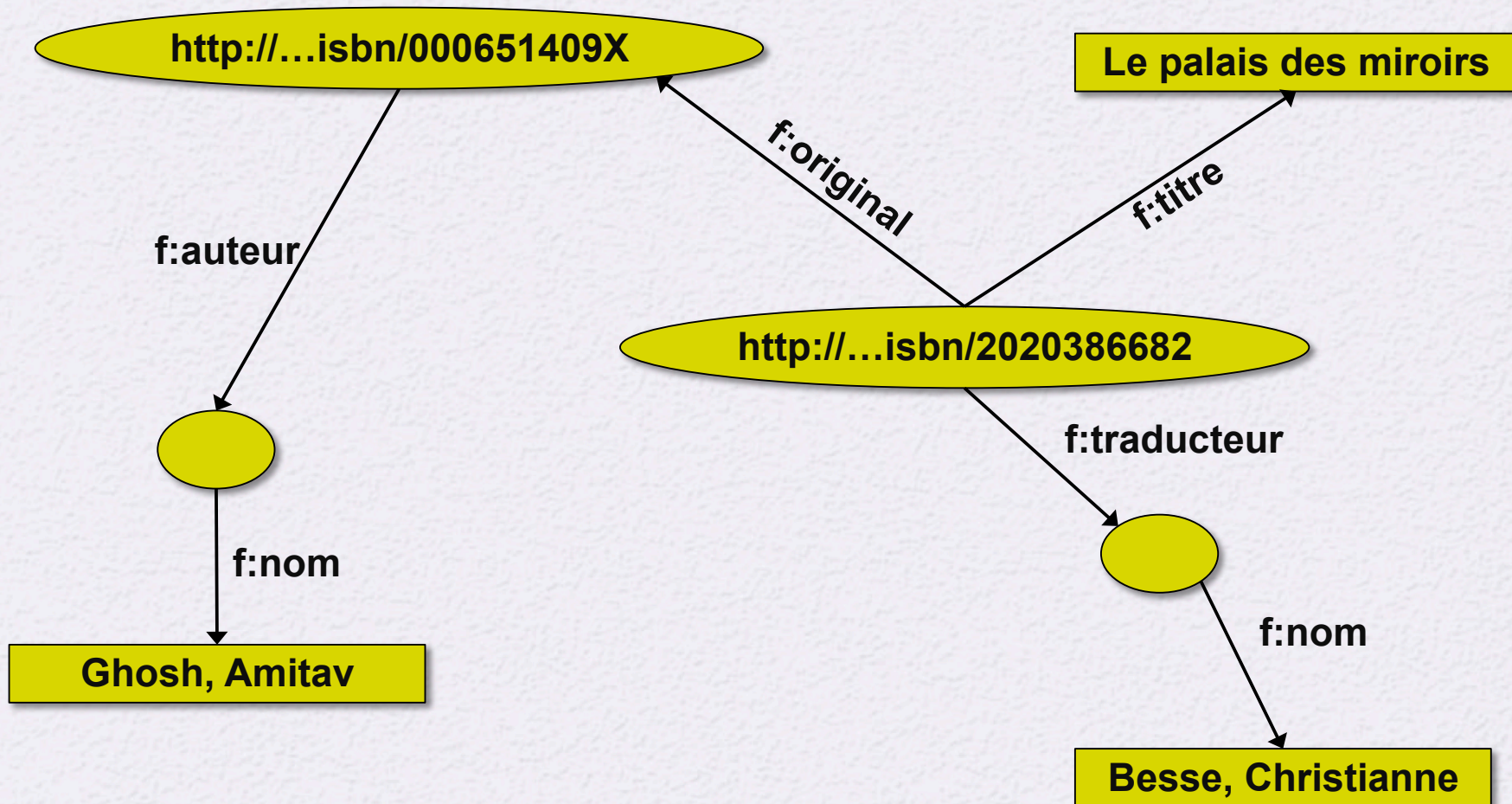
Same book in French...



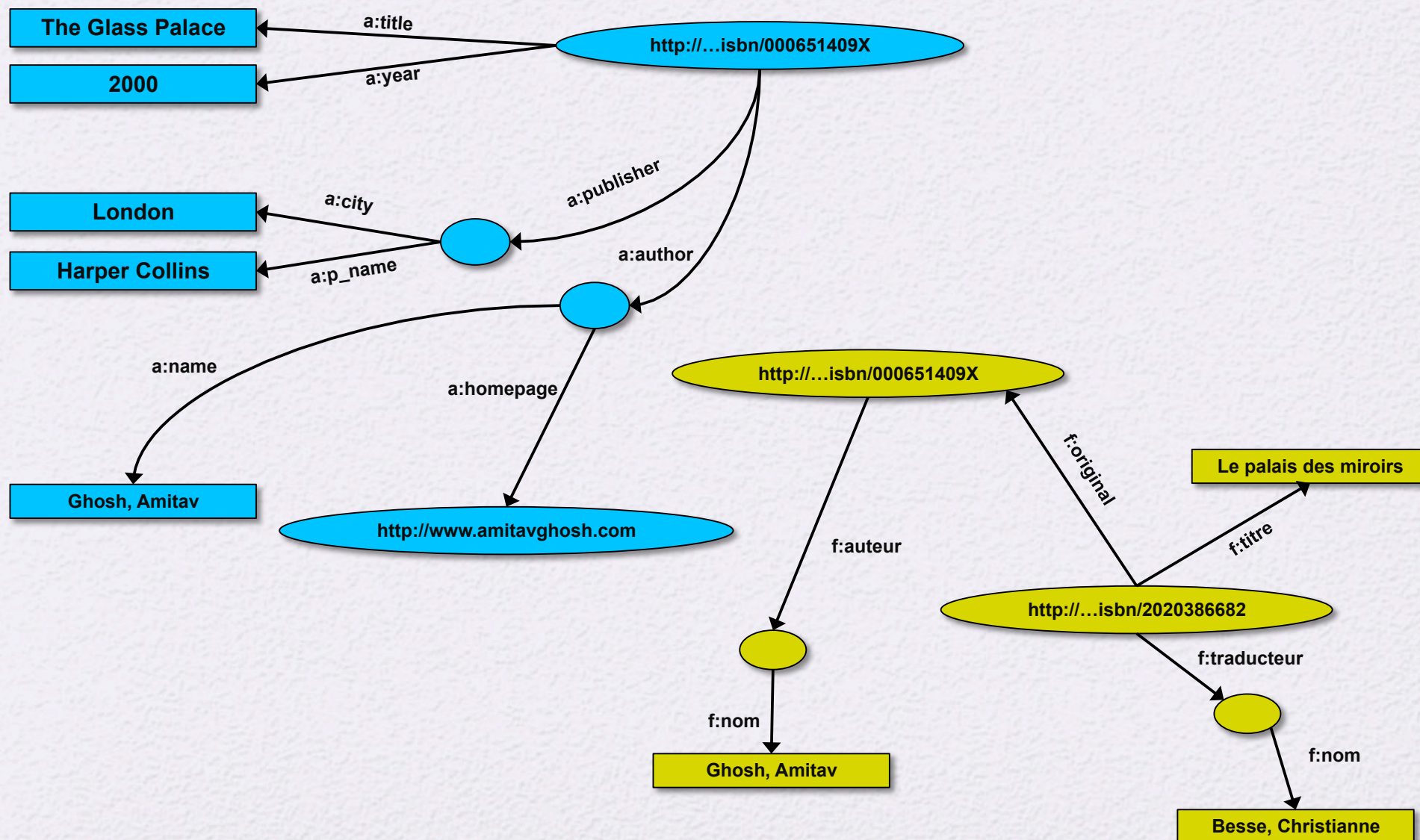
Another bookstore data (dataset “F”)

A	B	C	D	
1	ID	Titre	Traducteur	Original
2	ISBN 2020286682	Le Palais des Miroirs	\$A12\$	ISBN 0-00-6511409-X
3				
4				
5				
6	ID	Auteur		
7	ISBN 0-00-6511409-X	\$A11\$		
8				
9				
10	Nom			
11	Ghosh, Amitav			
12	Besse, Christianne			

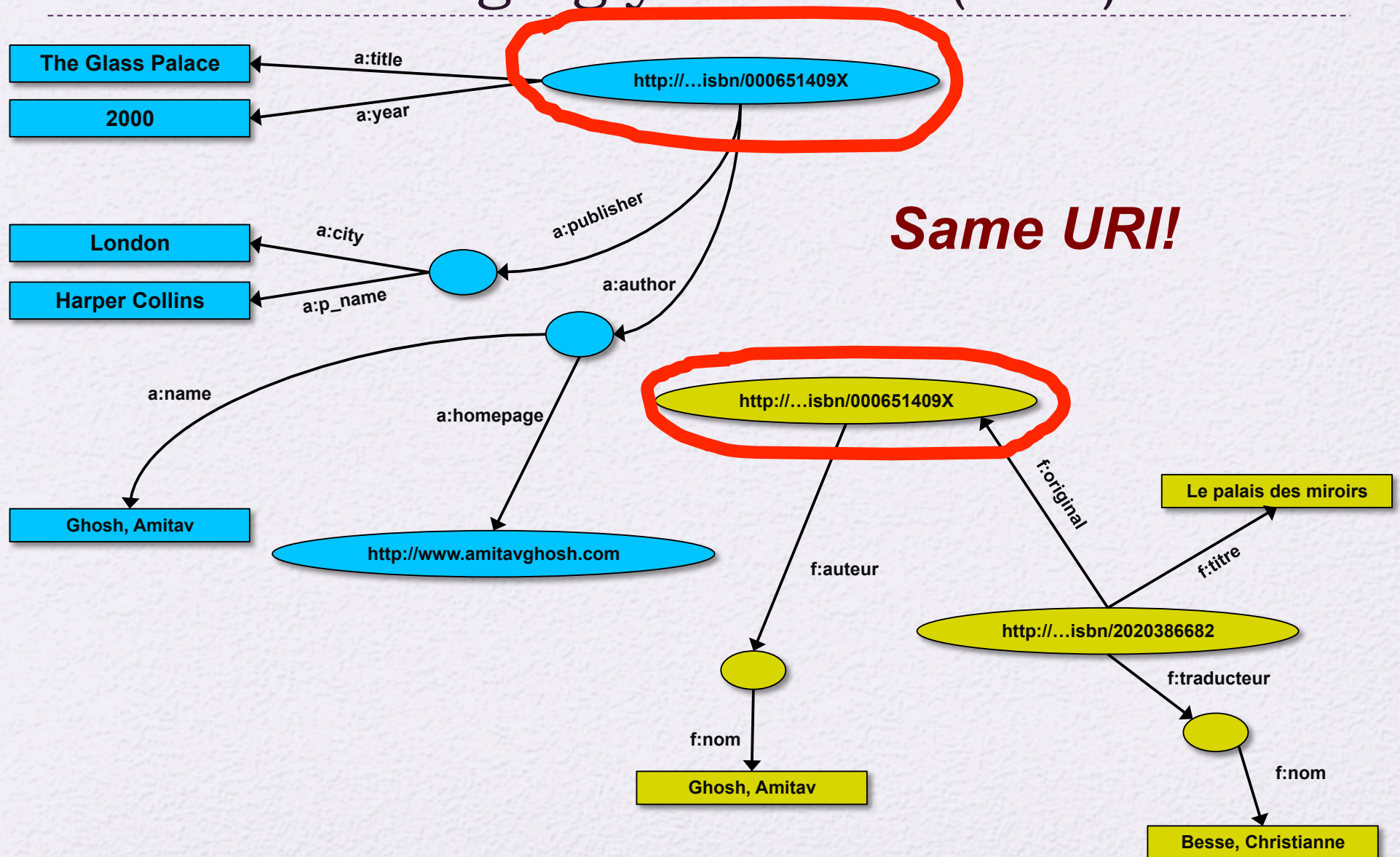
2nd: export your second set of data



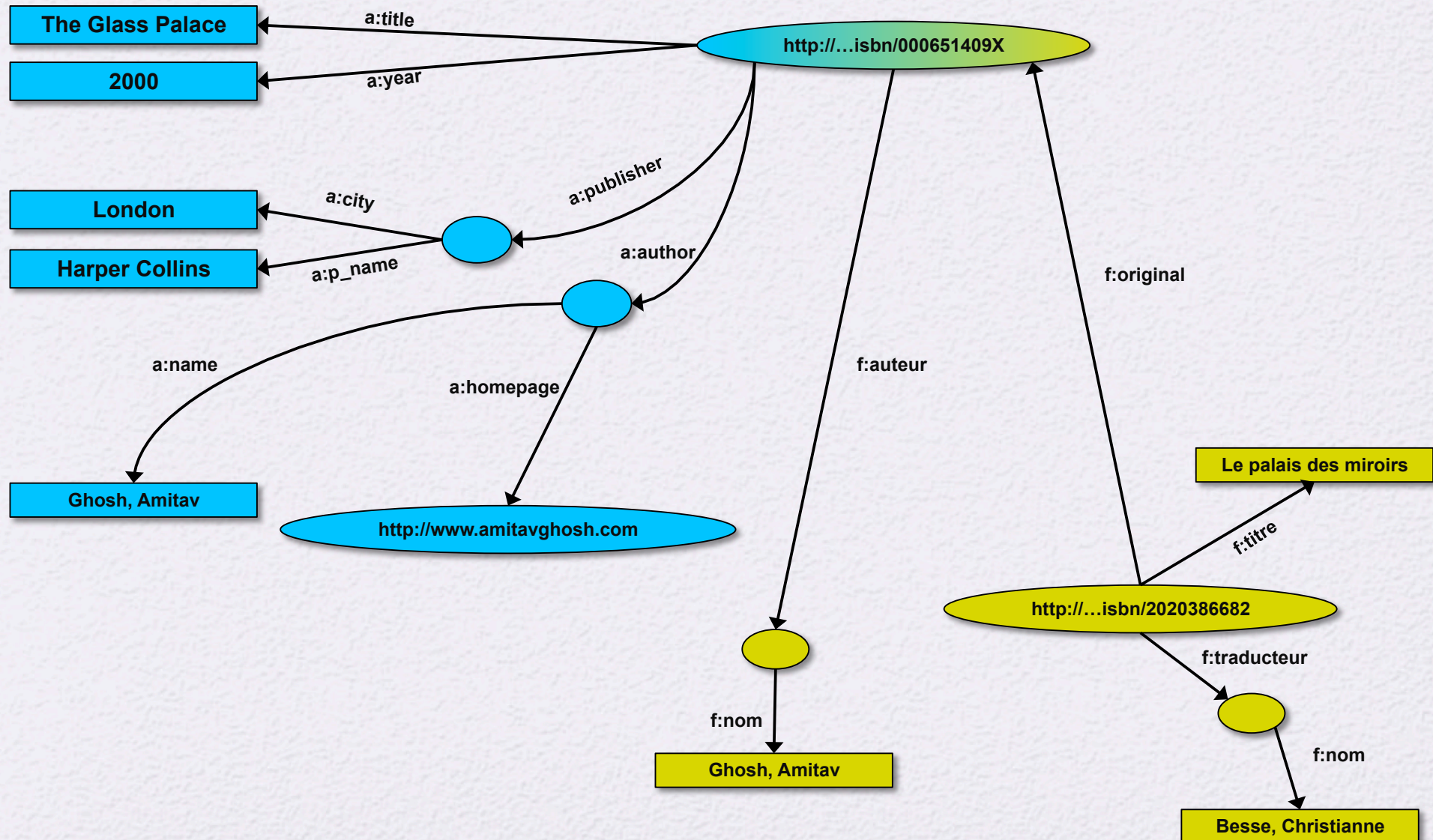
3rd: start merging your data



3rd: start merging your data (cont)

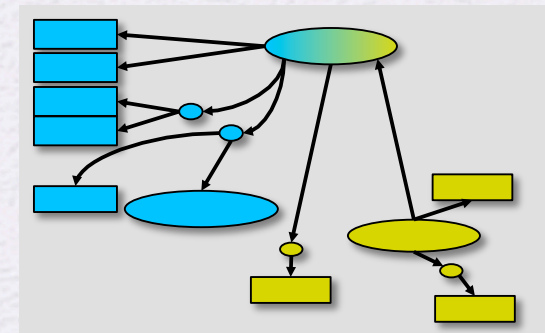


3rd: start merging your data



Start making queries...

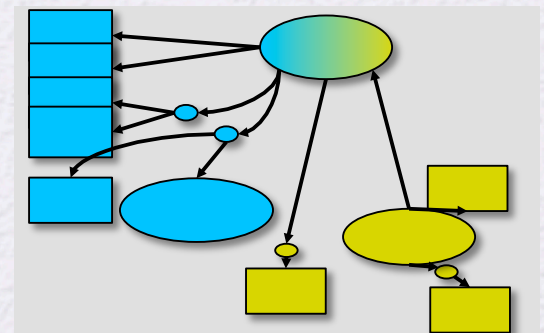
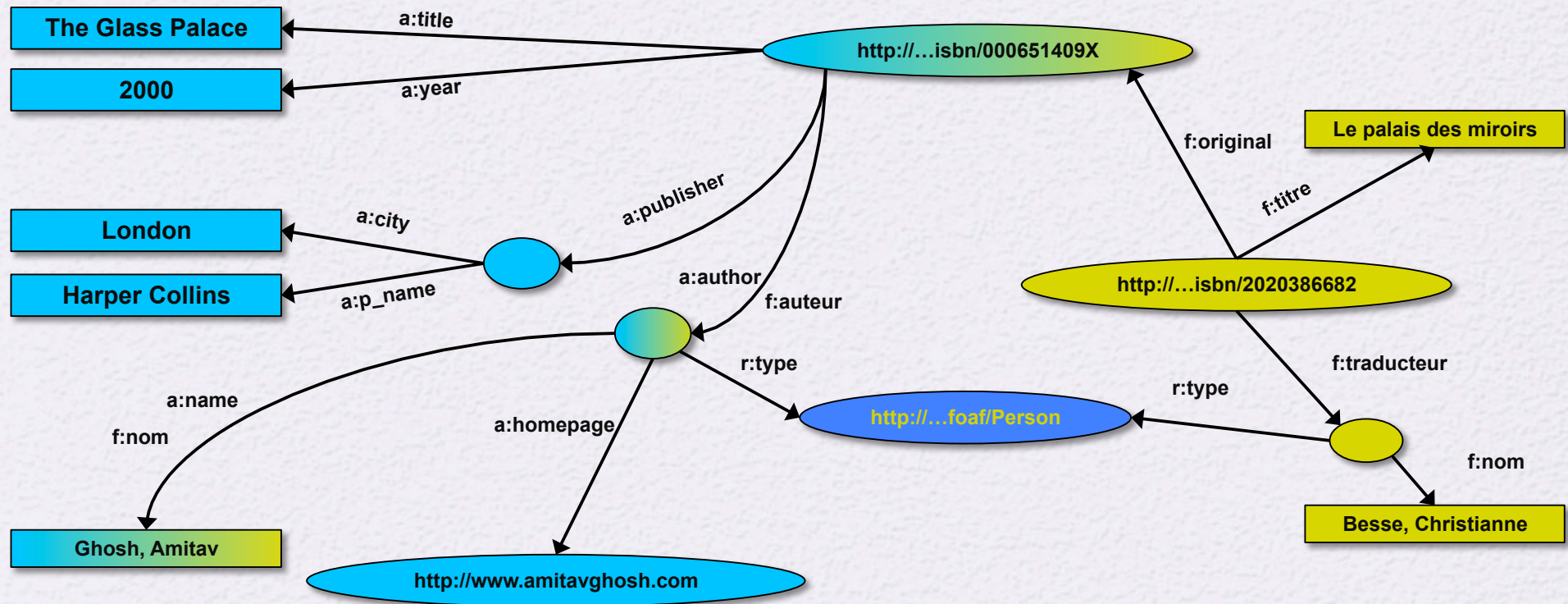
- ▶ User of data “F” can now ask queries like:
 - ▶ “give me the title of the original”
 - ▶ well, ... « donne-moi le titre de l’original »
- ▶ This information is not in the dataset “F”...
- ▶ ...but can be retrieved by merging with dataset “A”!



However, more can be achieved...

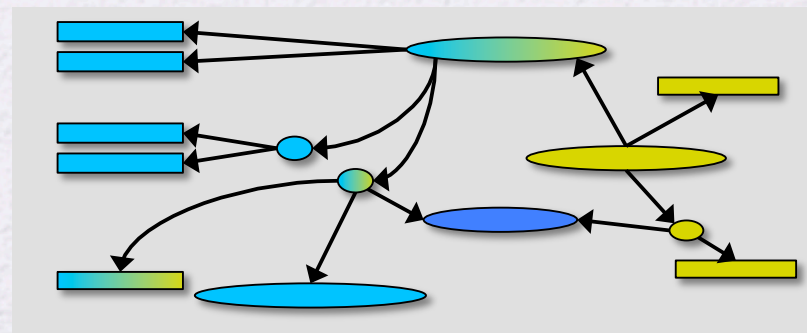
- ▶ We “feel” that a:author and f:auteur should be the same
- ▶ But an automatic merge does not know that!
- ▶ Let us add some extra information to the merged data:
 - ▶ a:author same as f:auteur
 - ▶ both identify a “Person”
 - ▶ a term that a community may have already defined:
 - ▶ a “Person” is uniquely 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!

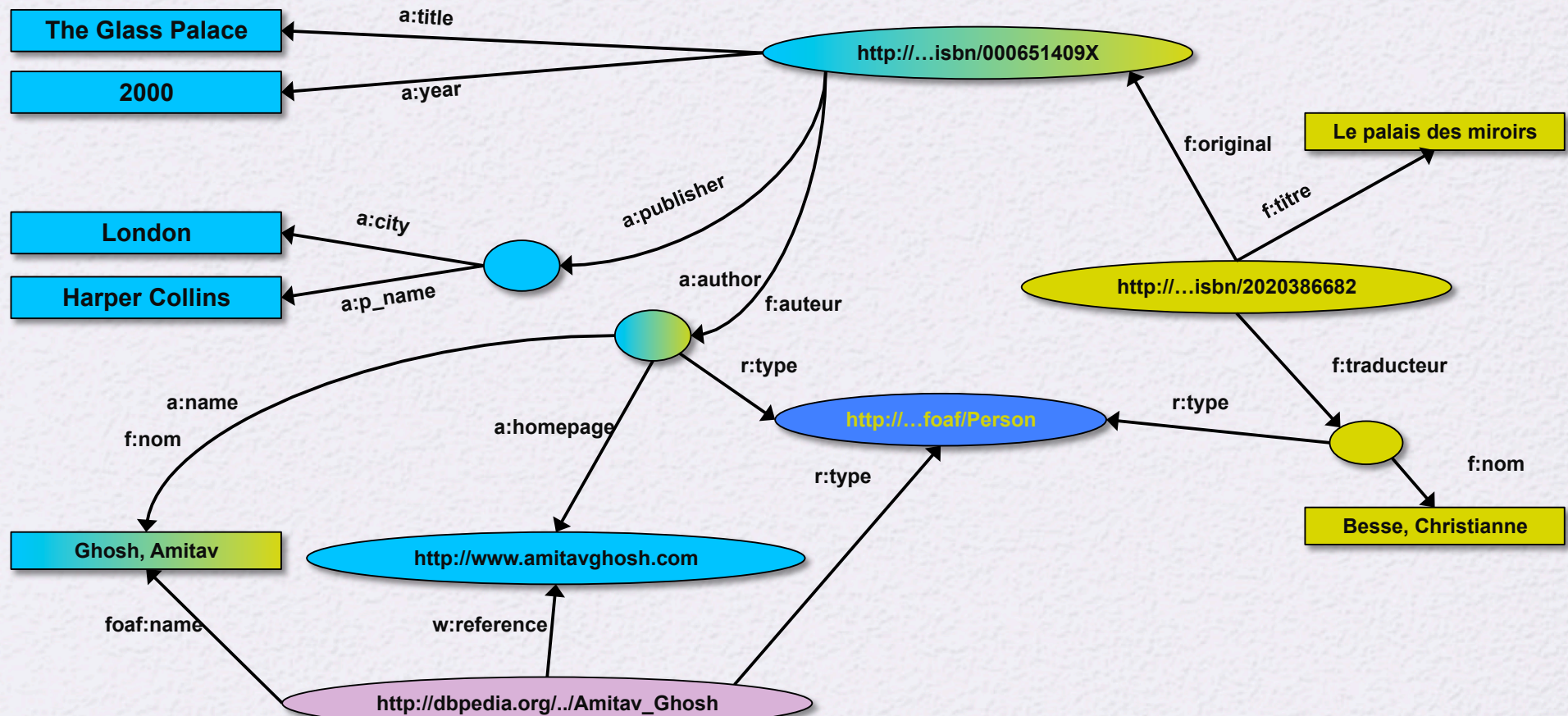
- ▶ User of dataset “F” can now query:
 - ▶ “donnes-moi la page d’accueil de l’auteur de l’original”
 - ▶ well... “give me the home page of the original’s ‘auteur’”
- ▶ The information is not in datasets “F” or “A”...
- ▶ ...but was made available by:
 - ▶ merging datasets “A” and datasets “F”
 - ▶ adding three simple extra statements as an extra “glue”



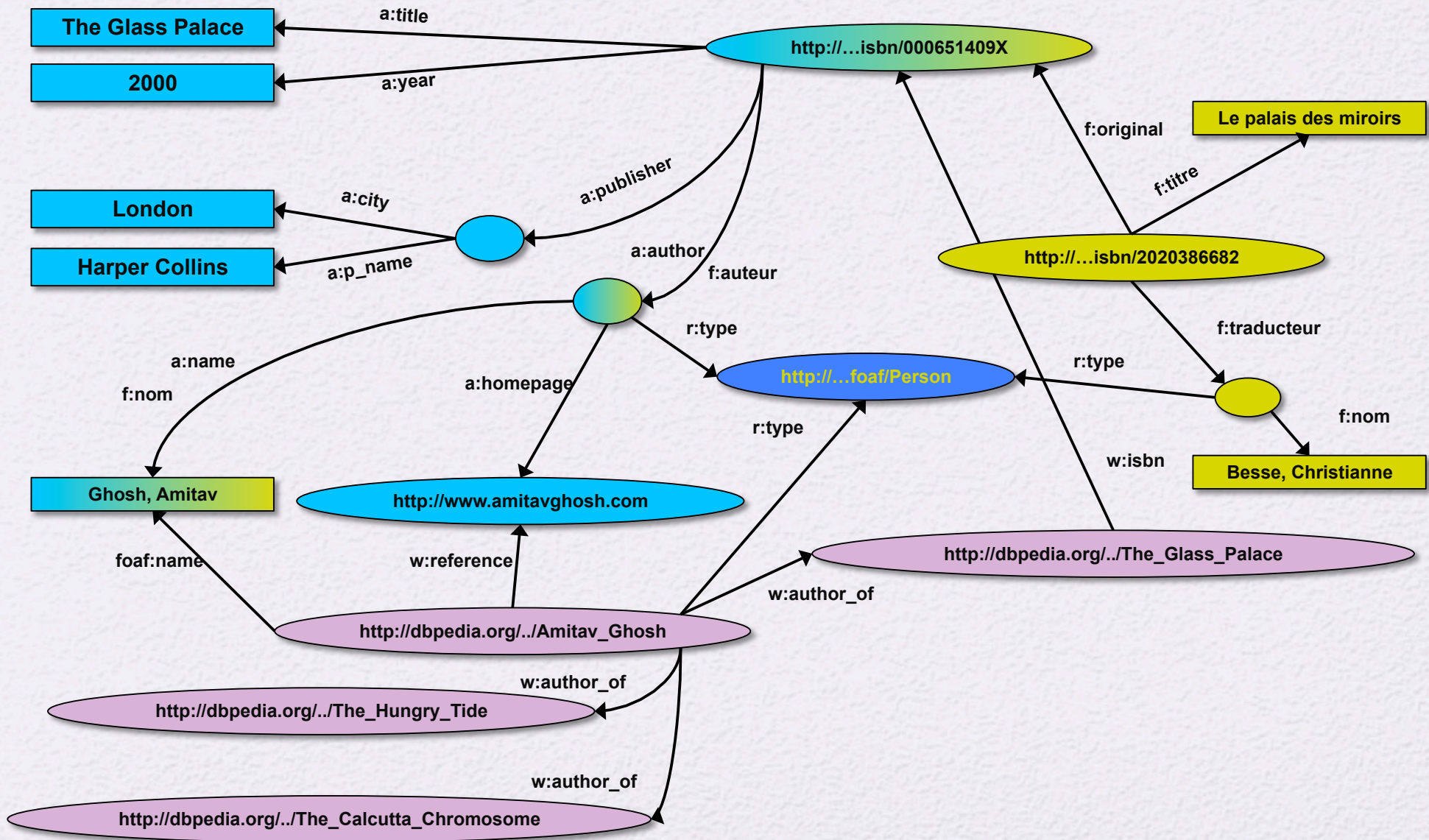
Combine with different datasets

- ▶ Using, e.g., the “Person”, the dataset can be combined with other sources
- ▶ For example, data in Wikipedia can be extracted using dedicated tools
 - ▶ e.g., the “[dbpedia](#)” project can extract the “infobox” information from Wikipedia already...

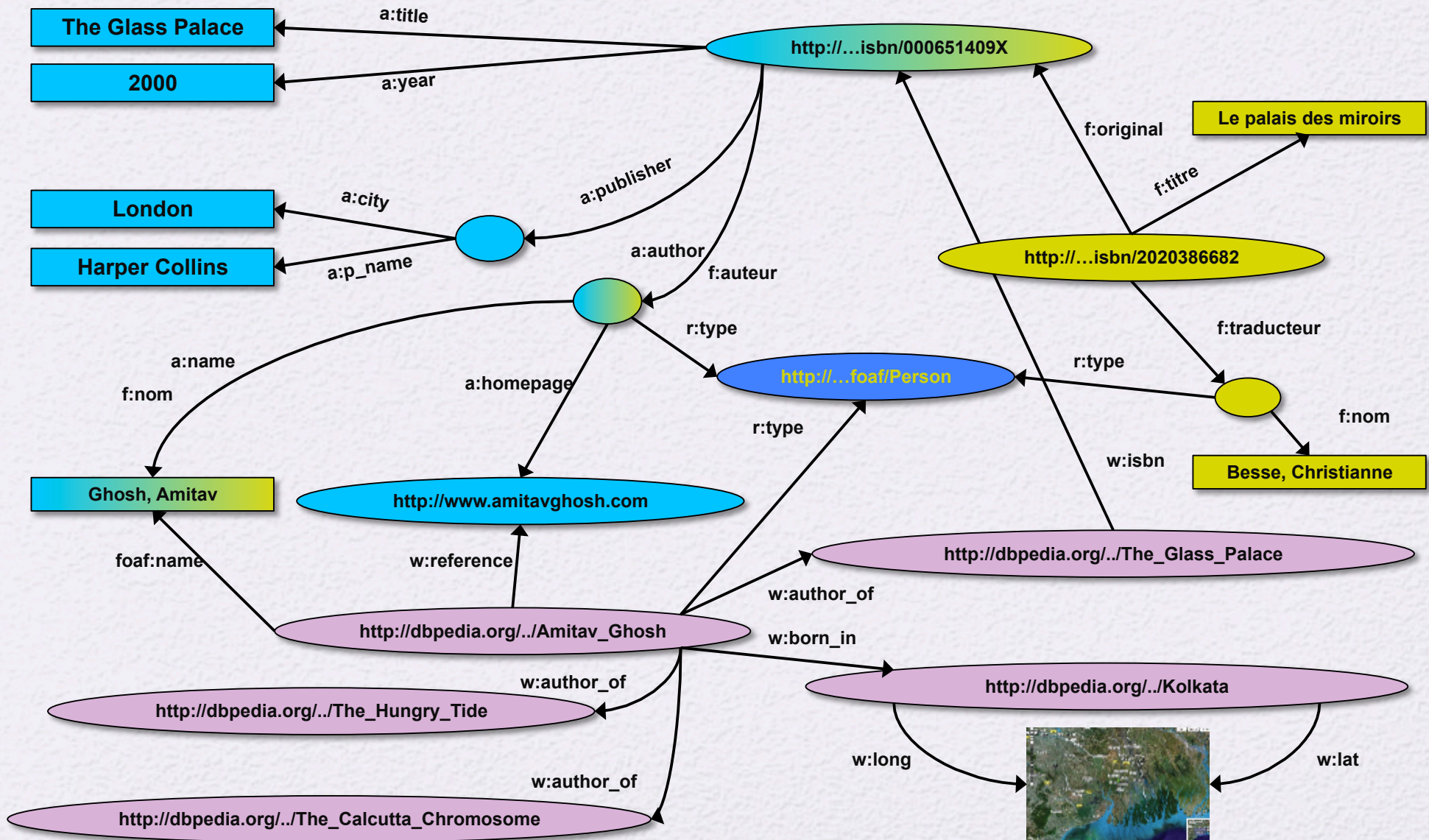
Merge with Wikipedia data



Merge with Wikipedia data



Merge with Wikipedia data



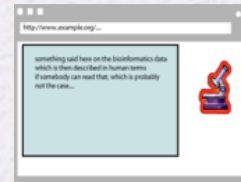
Is that surprising?

- ▶ It may look like it but, in fact, it should not be...
- ▶ What happened via automatic means is done every day by Web users!
- ▶ The difference: a bit of extra rigour so that machines could do this, too

It could become even more powerful

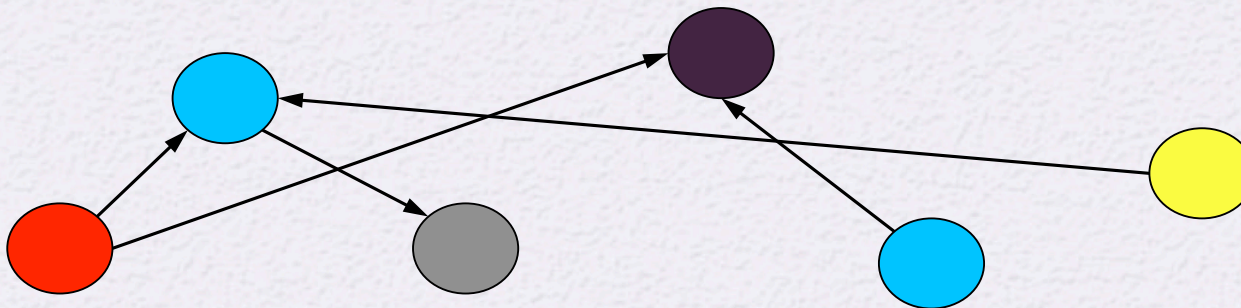
- ▶ We could add extra knowledge to the merged datasets
 - ▶ e.g., a full classification of various types of library data
 - ▶ geographical information
 - ▶ etc.
- ▶ This is where ontologies, extra rules, etc, come in
 - ▶ ontologies/rule sets can be relatively simple and small, or huge, or anything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

What did we do?



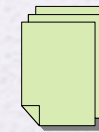
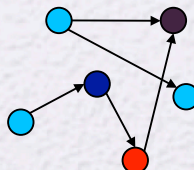
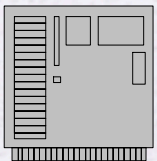
Applications

Manipulate
Query
...



Data represented in abstract format

Map,
Expose,
...



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...

The Basis: RDF

RDF triples

- ▶ Let us begin to formalize what we did!
 - ▶ we “connected” the data...
 - ▶ but a simple connection is not enough... data 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”, “property”, and “object”
 - ▶ here is the complete triple:

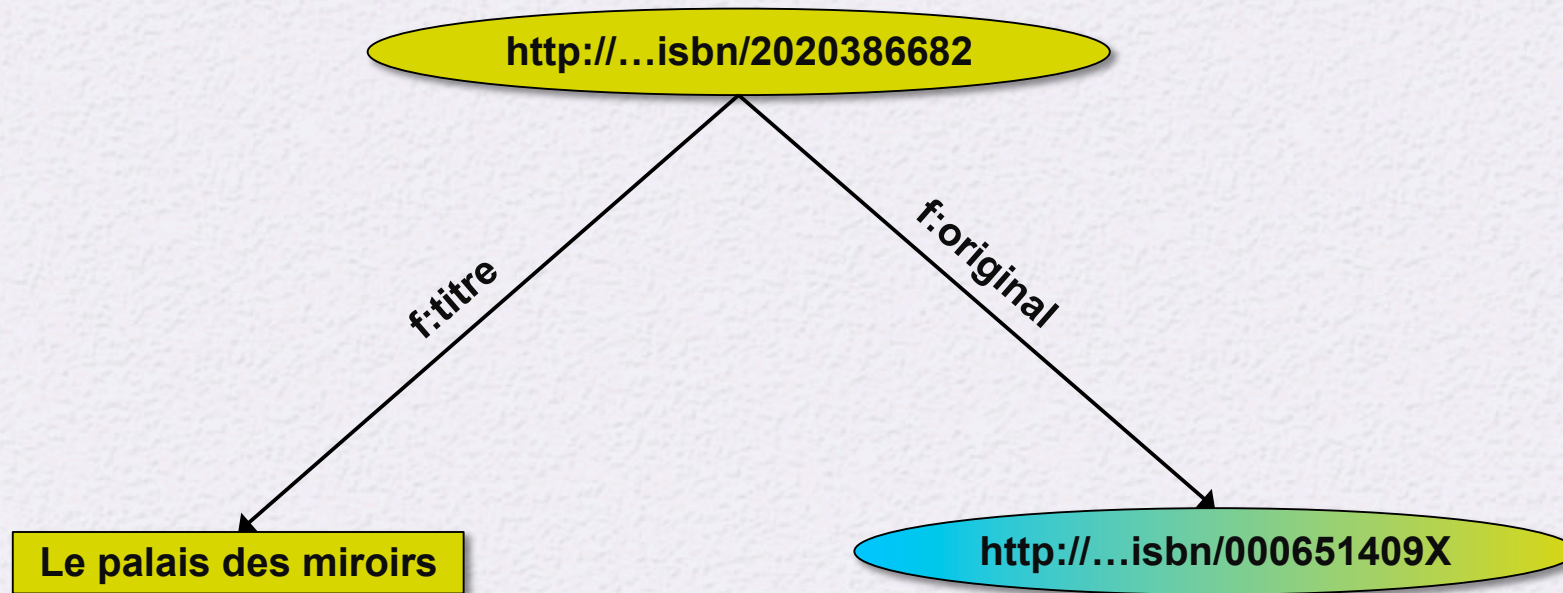
```
(<http://...isbn...6682>, <http://.../original>, <http://...isbn...409X>)
```

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

RDF triples (cont.)

- ▶ Resources can use *any* URI
 - ▶ `http://www.example.org/file.html#home`
 - ▶ `http://www.example.org/file2.xml#xpath(//q[@a=b])`
 - ▶ `http://www.example.org/form?a=b&c=d`
- ▶ RDF triples form a directed, labeled graph (the best way to think about them!)

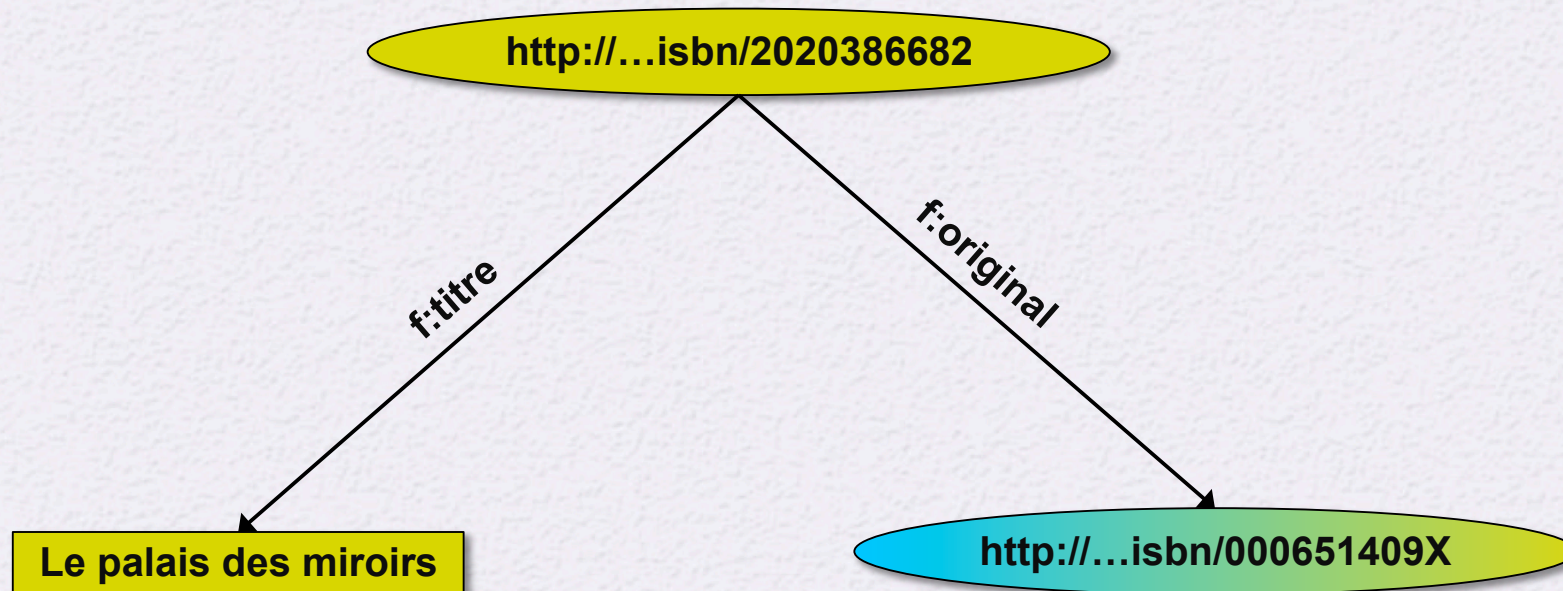
A simple RDF example (in RDF/XML)



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

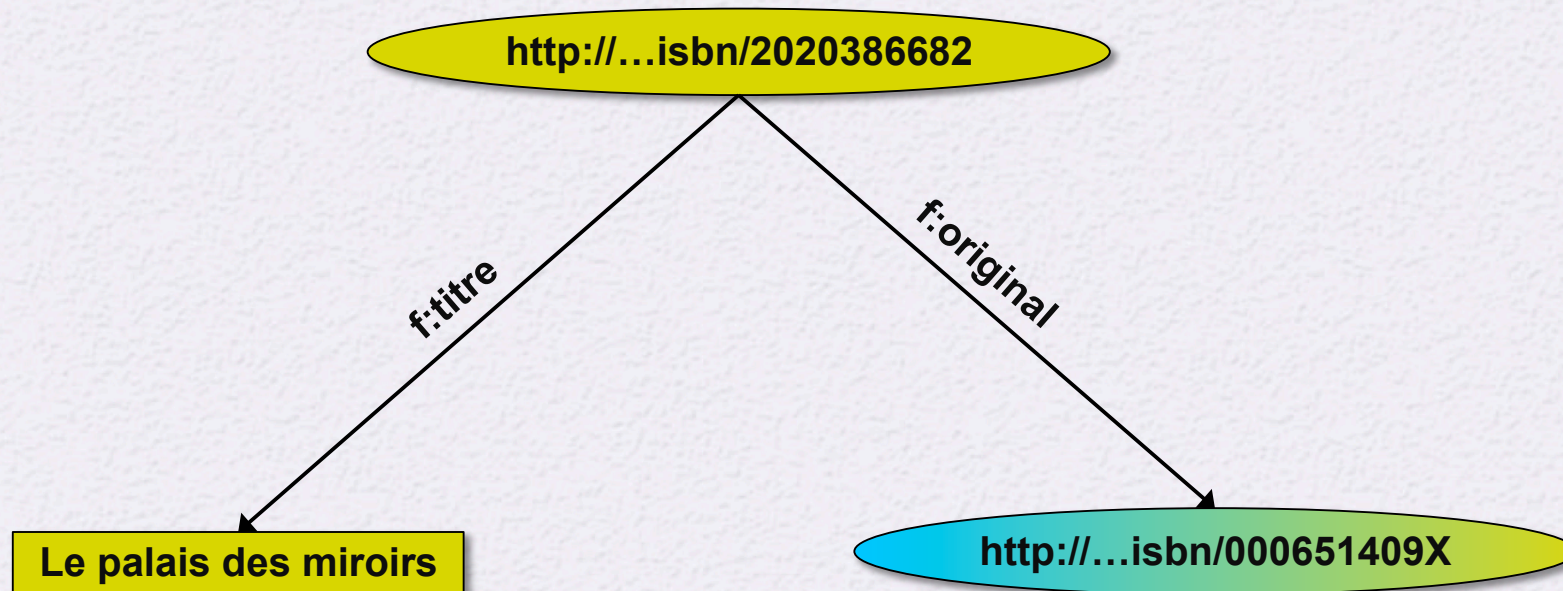
(Note: namespaces are used to simplify the URI-s)

A simple RDF example (in Turtle)



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


A simple RDF example (in RDFa)



```
<p about="http://.../isbn/2020386682">The book entitled  
"<span property="f:title" lang="fr">Le palais des miroirs</span>"  
is the French translation of the  
"<span rel="f:original" resource="http://.../isbn/000651409X">Glass  
Palace</span>"</p> .
```


“Internal” nodes

- ▶ Consider the following statement:
 - ▶ “the publisher is a «thing» that has a name and an address”
- ▶ Until now, nodes were identified with a URI. But...
- ▶ ...what is the URI of «thing»?



One solution: create an extra URI

- ▶ The resource will be “visible” on the Web
 - ▶ care should be taken to define unique URI-s

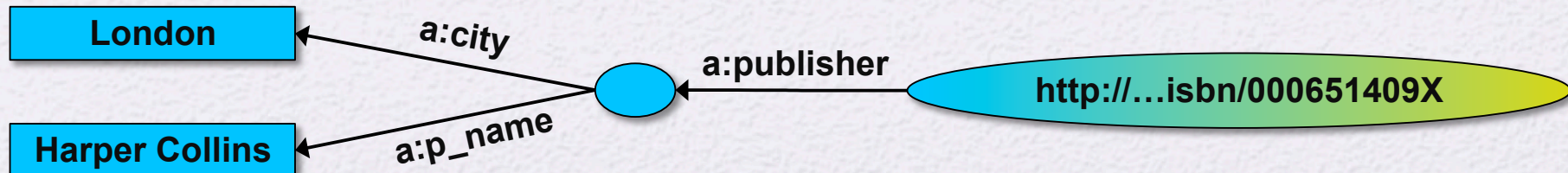
```
<rdf:Description rdf:about="http://.../isbn/000651409X">
  <a:publisher rdf:resource="urn:uuid:f60ffb40-307d-..." />
</rdf:Description>
<rdf:Description rdf:about="urn:uuid:f60ffb40-307d-...">
  <a:p_name>HarpersCollins</a:p_name>
  <a:city>HarpersCollins</a:city>
</rdf:Description>
```


Internal identifier (“blank nodes”)

```
<rdf:Description rdf:about="http://.../isbn/000651409X">
  <a:publisher rdf:nodeID="A234"/>
</rdf:Description>
<rdf:Description rdf:nodeID="A234">
  <a:p_name>HarpersCollins</a:p_name>
  <a:city>HarpersCollins</a:city>
</rdf:Description>
```

```
<http://.../isbn/2020386682> a:publisher _:A234.
_:A234 a:p_name "HarpersCollins".
```

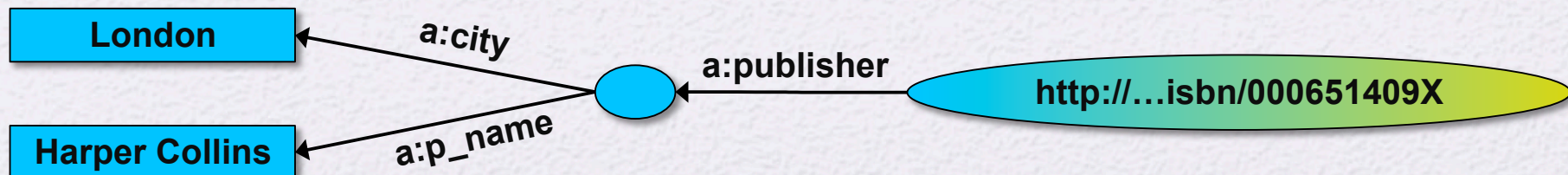
Internal = these resources are *not* visible outside



Blank nodes: the system can do it

- ▶ Let the system create a “nodeID” internally (you do not really care about the name...)

```
<http://.../isbn/000651409X> a:publisher [  
  a:p_name "HarpersCollins";  
  ...  
].
```



Blank nodes when merging

- ▶ Blank nodes require attention when merging
 - ▶ blank nodes with identical nodeID-s in different graphs are different
 - ▶ implementations must be careful...

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.

Jena example

```
// create a model
Model model=new ModelMem();
Resource subject=model.createResource("URI_of_Subject")
// 'in' refers to the input file
model.read(new InputStreamReader(in));
StmtIterator iter=model.listStatements(subject,null,null);
while(iter.hasNext()) {
    st = iter.next();
    p = st.getProperty();
    o = st.getObject();
    do_something(p,o);
}
```


Merge in practice

- ▶ Environments merge graphs automatically
 - ▶ e.g., in Jena, the Model can load several files
 - ▶ the load merges the new statements automatically
 - ▶ merge takes care of blank node issues, too

Another relatively simple application

- Goal: reuse of older experimental data
- Keep data in databases or XML, just export key “fact” as RDF
- Use a faceted browser to visualize and interact with the result

Internal Compound Repurposing Example

Welcome, Allergy & Respiratory Team Member

This tool allows you to identify opportunities for additional uses of compounds from other teams within your project. It combines internal data, public data and the results of data mining experiments to provide testable hypotheses.

Control Panel & Item Filtering

Internal Compound Repurposing Example

Welcome, Allergy & Respiratory Team Member

This tool allows you to identify opportunities for additional uses of compounds from other teams within your project. It combines internal data, public data and the results of data mining experiments to provide testable hypotheses.

Control Panel & Item Filtering

Area	Approach	Term+Reason	Max_Stage_Reached	Literature Links
20 Pain	<input checked="" type="checkbox"/> 7 Antibody	<input type="checkbox"/> 37 ACTIVE	<input type="checkbox"/> 51 Candidate	<input checked="" type="checkbox"/> 0 - 50
16 Metabolic Disease	<input checked="" type="checkbox"/> 1 Recombinant	<input type="checkbox"/> 12 BIOMARKER	<input type="checkbox"/> 10 Discovery	<input type="checkbox"/>
8 Cancer	<input type="checkbox"/> 18 SM_Agonist	<input checked="" type="checkbox"/> 51 EFFICACY	<input checked="" type="checkbox"/> 41 Exploratory	<input type="checkbox"/>
3 Sexual Health	<input checked="" type="checkbox"/> 12 SM_Antagonist	<input checked="" type="checkbox"/> 11 MARKET	<input type="checkbox"/> 19 HTS	<input type="checkbox"/>
2 Infectives	<input checked="" type="checkbox"/> 21 SM_Inhibitor	<input checked="" type="checkbox"/> 11 REORG	<input type="checkbox"/> 11 Phase I	<input type="checkbox"/>
1 Urogenitals	<input checked="" type="checkbox"/>	<input type="checkbox"/> 10 TOXIC	<input type="checkbox"/> 13 Phase III	<input type="checkbox"/>
			<input type="checkbox"/> 41 Screening	<input type="checkbox"/>

51 items filtered from 710 originally (Reset All Filters)

Area	Original+Indication	Target_Name	Approach	Start	Term+Reason	Max_Stage_Reached	Owner	OMIM	UR	Alt_UR	2007	UR_Mech	IMA	GEO	Pathway	Compounds
Metabolic Disease	Diabetes	Liver glycogen phosphorylase	SM_Inhibitor	2007-Q2	EFFICACY	Candidate	P. Person									SW-030072
Sexual Health	Erectile Dysfunction	Integrin alpha-3 (Galactose protein 83)(VLA-3) (CD49C)	SM_Antagonist	2006-Q3	EFFICACY	Candidate	P. Person						1			SW-029782
Sexual Health	Erectile Dysfunction	Leukotriene C4 synthase	SM_Agonist	2006-Q3	EFFICACY	Candidate	M. Manager					1	1			SW-029638
Sexual Health	Erectile Dysfunction	transcription elongation factor A (SII)-like 4	SM_Inhibitor	2005-Q2	EFFICACY	Candidate	P. Person									SW-029626
Infectives	HIV	Putative four repeat ion channel (Iv)	SM_Inhibitor	2006-Q2	EFFICACY	Candidate	L. Leader									SW-029994
Infectives	HIV	Voltage-gated potassium channel protein V3.2 (Iv)	SM_Agonist	2007-Q1	EFFICACY	Candidate	A. Scientist						1			SW-029653
Urogenitals	Incontinence	Human RNA binding motif (RBM) gene, partial cds	SM_Agonist	2007-Q3	EFFICACY	Candidate	L. Leader							1		SW-029684
Pain	Migraine	Monocarboxylate transporter homologue 2294064CD1 (Iv)	SM_Inhibitor	2007-Q3	EFFICACY	Candidate	L. Leader	18								SW-030085

One level higher up

(RDFS, Datatypes)

Need for RDF schemas

- ▶ First step towards the “extra knowledge”:
 - ▶ define the terms we can use
 - ▶ what restrictions apply
 - ▶ what extra relationships are there?
- ▶ Officially: “RDF Vocabulary Description Language”
 - ▶ the term “Schema” is retained for historical reasons...

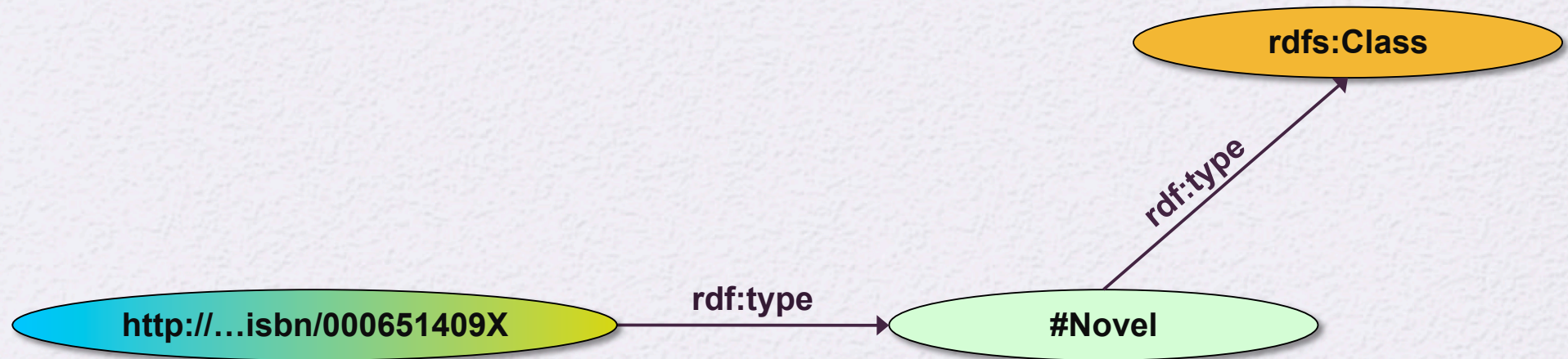
Classes, resources, ...

- ▶ Think of well known traditional vocabularies:
 - ▶ 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.)

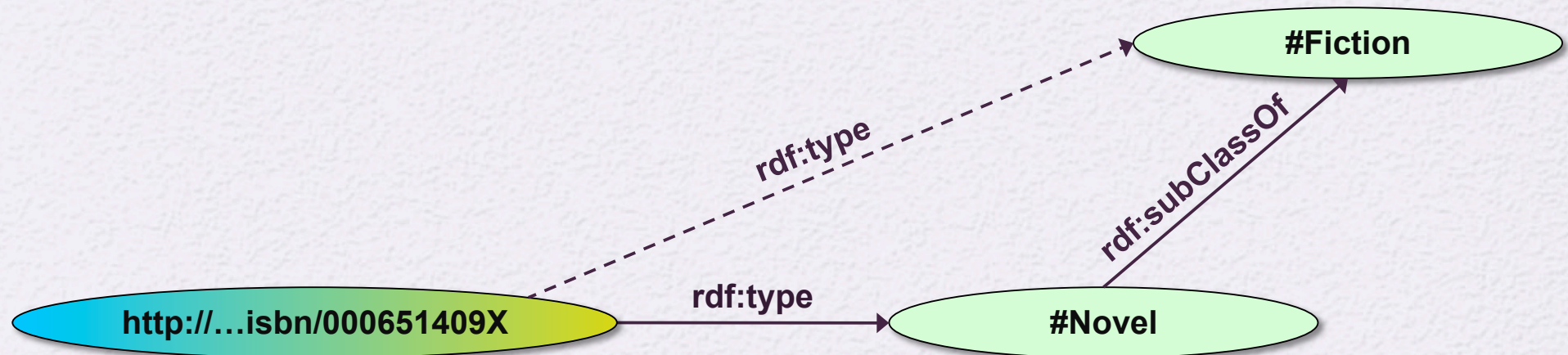
- ▶ Relationships are defined among resources:
 - ▶ “typing”: an individual belongs to a specific class
 - ▶ “«The Glass Palace» is a novel”
 - ▶ to be more precise: “«<http://.../000651409X>» is a novel”
 - ▶ “subclassing”: all instances of one are also the instances of the other (“every novel is a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

Classes, resources in RDF(S)



- ▶ RDFS defines the meaning of these terms
 - ▶ (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 environments return that triple, too

Inference: let us be formal...

- ▶ The RDF Semantics document has a list of (33) entailment rules:
 - ▶ “if such and such triples are in the graph, add this and this”
 - ▶ 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

- ▶ Property is a special class (rdf:Property)
 - ▶ properties are also resources identified by URI-s
- ▶ There is also a possibility for a “sub-property”
 - ▶ all resources bound by the “sub” are also bound by the other
- ▶ Range and domain of properties can be specified
 - ▶ i.e., what type of resources serve as object and subject

Example for property characterization

```
: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 .
```


Literals

- ▶ Literals may have a data type
 - ▶ floats, integers, booleans, etc, defined in XML Schemas
 - ▶ full XML fragments
- ▶ (Natural) language can also be specified

Examples for datatypes

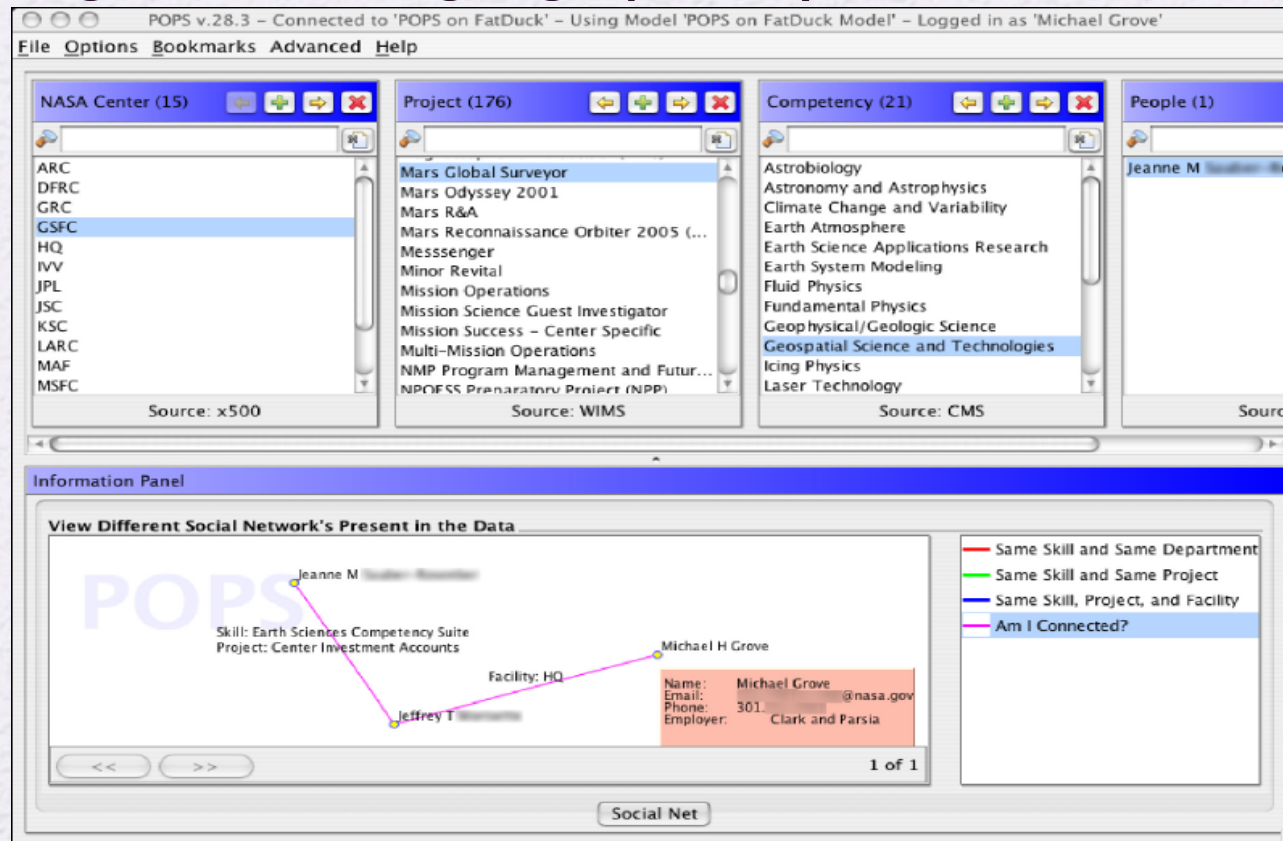
```
<http://.../isbn/000651409X>  
  :page_number "543"^^xsd:integer ;  
  :publ_date    "2000"^^xsd:gYear ;  
  :price        "6.99"^^xsd:float .
```


A bit of RDFS can take you far...

- ▶ Remember the power of merge?
- ▶ We could have used, in our example:
 - ▶ f:auteur is a subproperty 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 (see later...)

Find the right experts at NASA

- Expertise locator for nearly 70,000 NASA civil servants,
- integrate 6 or 7 geographically distributed databases, ...



How to get and create RDF Data?

Simple approach

- ▶ Write RDF/XML, RDFa, or Turtle “manually”
- ▶ In some cases that is necessary, but it really does not scale...

RDF with XHTML

- ▶ Obviously, a huge source of information
- ▶ By adding some “meta” information, the same source can be reused for, eg, data integration, better mashups, etc
- ▶ typical example: your personal information, like address, should be readable for humans and processable by machines

RDF with XML/ (X)HTML (cont)

- ▶ Two solutions have emerged:
 - ▶ use microformats and convert the content into RDF
 - ▶ XSLT is the favorite approach
 - ▶ add RDF-like statements directly into XHTML via RDFa

Bridge to relational databases

- ▶ Data on the Web are mostly stored in databases
- ▶ “Bridges” are being defined:
 - ▶ a layer between RDF and the relational data
 - ▶ RDB tables are “mapped” to RDF graphs, possibly on the fly
 - ▶ different mapping approaches are being used
 - ▶ a number RDB systems offer this facility already (eg, Oracle, OpenLink, ...)
- ▶ W3C is working on a standard in this area

Linked Open Data

Linked Open Data Project

- ▶ Goal: “expose” open datasets in RDF
- ▶ Set RDF links among the data items from different datasets
- ▶ Set up, if possible, query endpoints

Example data source: DBpedia

- ▶ DBpedia is a community effort to
 - ▶ extract structured (“infobox”) information from Wikipedia
 - ▶ provide a query endpoint to the dataset
 - ▶ interlink the DBpedia dataset with other datasets on the Web



UNIVERSITÄT LEIPZIG



Extracting structured data from Wikipedia

```
@prefix dbpedia <http://dbpedia.org/resource/>.
```

```
@prefix dbterm <http://dbpedia.org/property/>.
```

```
dbpedia:Amsterdam
```

```
dbterm:officialName "Amsterdam" ;
```

```
dbterm:longd "4" ;
```

```
dbterm:longm "53" ;
```

```
dbterm:longs "32" ;
```

```
dbterm:leaderName dbpedia:Lodewijk_Asscher ;
```

```
...
```

```
dbterm:areaTotalKm "219" ;
```

```
...
```

```
dbpedia:ABN_AMRO
```

```
dbterm:location dbpedia:Amsterdam ;
```

```
...
```

Amsterdam



The Keizersgracht at dusk

Location of Amsterdam

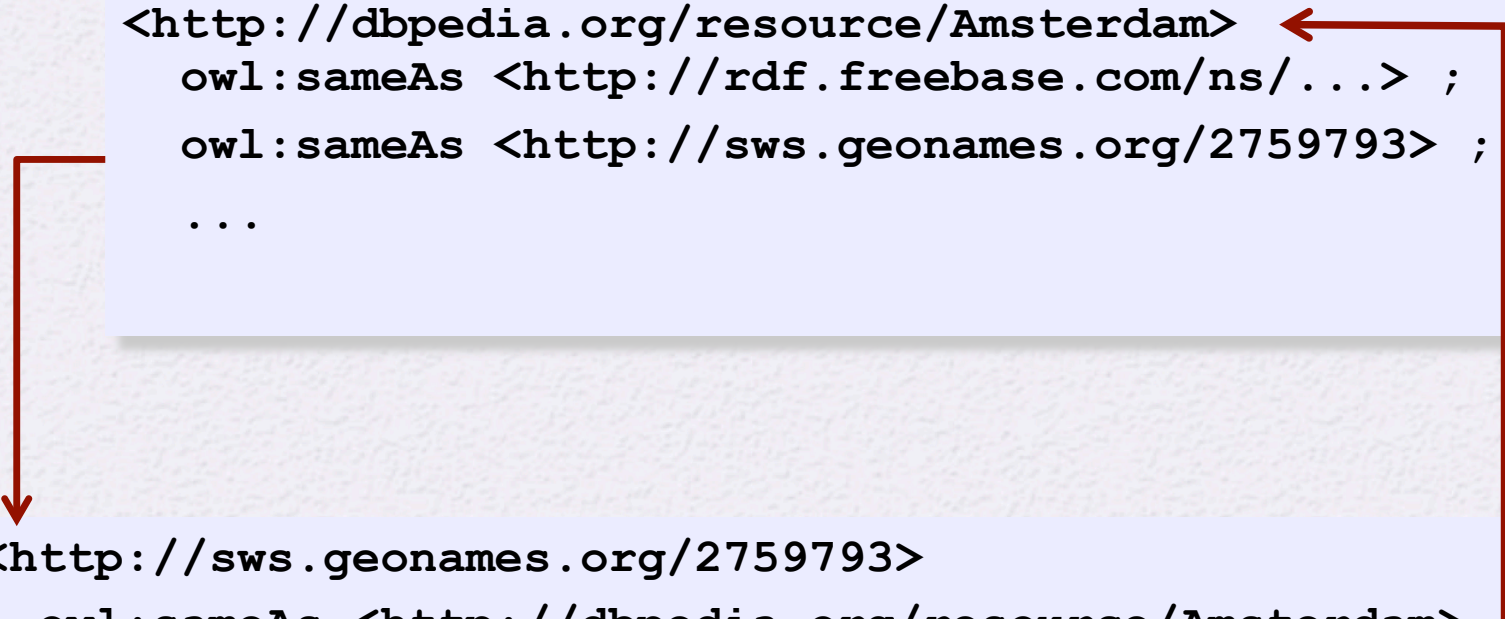
Coordinates:  52°22′23″N 4°53′32″E﻿ / ﻿

Country	 Netherlands
Province	 North Holland
Government	
 - Type	Municipality
 - Mayor	Job Cohen ^[1] (PvdA)
 - Aldermen	Lodewijk Asscher Carolien Gehrels Tjeerd Herrema Maarten van Poelgeest Marijke Vos
 - Secretary	Erik Gerritsen
Area ^{[2][3]}	
 - City	219 km² (84.6 sq mi)
 - Land	166 km² (64.1 sq mi)
 - Water	53 km² (20.5 sq mi)
 - Urban	1,003 km² (387.3 sq mi)
 - Metro	1,815 km² (700.8 sq mi)
Elevation ^[4]	2 m (7 ft)
Population (1 October 2008) ^{[5][6]}	
 - City	755,269
 - Density	4,459/km² (11,548.8/sq mi)
 - Urban	1,364,422
 - Metro	2,158,372
 - Demonym	Amsterdammer
Time zone	CET (UTC+1)
 - Summer (DST)	CEST (UTC+2)
Postcodes	1011 – 1109
Area code(s)	020

Website: www.amsterdam.nl 

Automatic links among open datasets

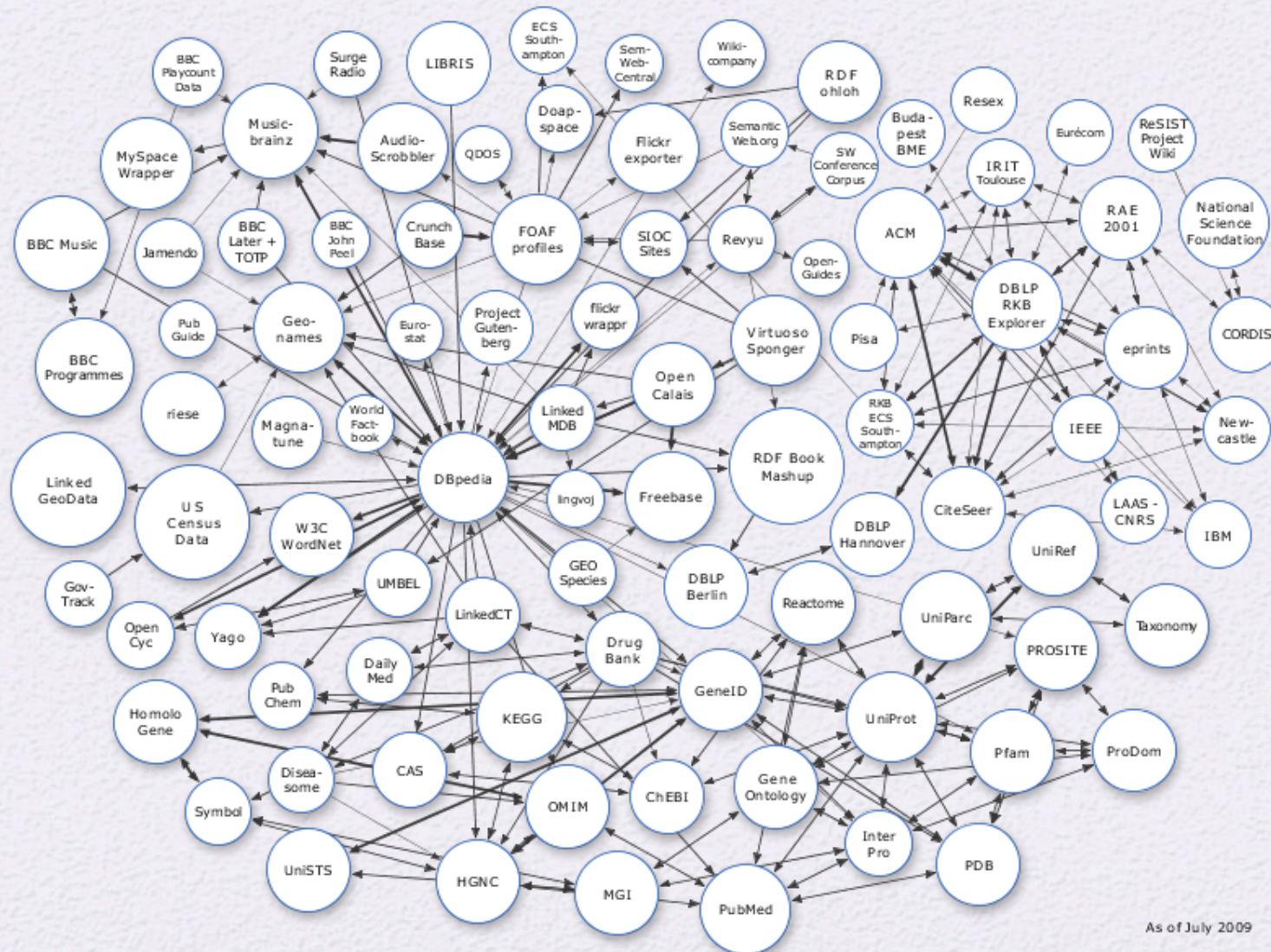
```
<http://dbpedia.org/resource/Amsterdam>
  owl:sameAs <http://rdf.freebase.com/ns/...> ;
  owl:sameAs <http://sws.geonames.org/2759793> ;
  ...
```



```
<http://sws.geonames.org/2759793>
  owl:sameAs <http://dbpedia.org/resource/Amsterdam>
  wgs84_pos:lat "52.3666667" ;
  wgs84_pos:long "4.8833333";
  geo:inCountry <http://www.geonames.org/countries/#NL> ;
  ...
```

Processors can switch automatically from one to the other...

The LOD “cloud”, June 2009



Remember the BBC example?

BBC - Music - Eric Clapton

http://www.bbc.co.uk/music/artists/618b6900-0618-4f1e-b835-bccb17f84294

Netvibes Feedly Social Private Mailing lists SW Python RDFa it! Bookmarklets Add Zemanta bit.ly To Mendeley TinyURL To Faviki Dokuwiki

MUSIC BETA GENRES ARTISTS REVIEWS NEWS BLOG


QUICK FIND Enter an artist name ...

BBC Music > Artists > Eric Clapton

Eric Clapton

Born 30 March 1945.

MOST PLAYED ON BBC RADIO 2



David Redfern/Redferns


Biography

Eric Patrick Clapton, CBE (born 30 March 1945) is an English blues-rock guitarist, singer, songwriter and composer. Clapton has been inducted into the Rock and Roll Hall of Fame as a solo performer, as a member of rock bands; the Yardbirds and Cream. Clapton is the only person ever to be inducted three times. Often viewed by critics and fans alike as one of the most important and influential guitarists of all time, Clapton was ranked fourth in Rolling Stone magazine's list of the "100 Greatest Guitarists of All Time" and #53 on their list of the Immortals: 100 Greatest Artists of All Time.

Latest Tracks Played On The BBC

- Promises**
BBC Radio 2 | [Ken Bruce 22/02/2010](#)
- Bad Love**
BBC Radio 2 | [Alex Lester 22/02/2010](#)
- Lay Down Sally**
BBC Radio 2 | [Chris Evans Breakfast 18/02/2010](#)
- I Ain't Gonna Stand For It**
BBC Radio 2 | [Alex Lester 15/02/2010](#)
- Wonderful Tonight**
BBC Radio 2 | [Ken Bruce 10/02/2010](#)

Audio Previews From Latest Album Review



Me And Mr Johnson

- 8 Milkcow's Calf Blues
- 10 Come on in My Kitchen

NYT articles on university alumni

Who Went Where

Hirek Social Private Mailing lists SW Python RDFa it! Bookmarklets bit.ly To Mendeley TinyURL To Faviki Dokuwiki MID

Who Went Where

The New York Times

Linked Open Data BETA


[View Application Source](#)

Alumni In The News


Enter a school name below and see our coverage of that school's alumni.

Harvard University

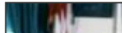
Harvard University

 **Henry Kissinger**
Born: May 27, 1923

Henry Kissinger Is Released From Hospital in South Korea - March 15, 2010
Got Your Back - July 19, 2009
SPORTS OF THE TIMES; Kissinger's Soccer Diplomacy - March 31, 2009
THE TV WATCH; A Question Reprised, but the Words Come None Too Easily for Palin - September 26, 2008
OP-ED COLUMNIST; Park Avenue Diplomacy - September 24, 2008
Palin Will Meet With Kissinger and Foreign Leaders - September 22, 2008
Using Star Power to Repair Nigeria's Image - July 10, 2008
OP-ED CONTRIBUTOR; Listening to Compromise - July 08, 2008
EDITORIAL OBSERVER; Thinking the Unthinkable: A World Without Nuclear Weapons - June 30, 2008
ON THE WHITE HOUSE; One Trip, Dual Purposes - April 27, 2008

 **Jacques Chirac**
Born: November 29, 1932

Chirac Faces New Inquiry On Charges Of Corruption - December 19, 2009
Ex-Leader Of France Faces Trial - October 31, 2009
French Power Elite Face a Fall From Grace - October 28, 2009
French President Raises Eyebrows With Remark on a Contentious Political Trial - September 25, 2009
INSIDE EUROPE; Order to Pay Back Farm Subsidies Comes at a Bad Time for France - August 11, 2009
WORLD BRIEFING | EUROPE; France: Chirac Creates Foundation - June 10, 2008
Putin Maintains Presidential Air in Paris Trip - May 31, 2008
A Statesman Without Borders - February 03, 2008
Chirac Under Investigation for Misuse of Funds as Paris Mayor - November 22, 2007
Correction: For the Record - November 05, 2007

 **Kennedy, John Fitzgerald**
Born: May 29, 1917

ESSAY; The Making of the President, Then and Now - March 21, 2010
Lost in Time: Letters Capture American Grief For a President - March 09, 2010

LIBRIS Sem-Web-Central Wiki-compa Se W Open Calais Linked MOD DBpedia ingvoj Freebase GEO Species DBLP Berlin Drug Bank GeneID KEGG OMIM ChEBI HGNC MGI Pub

Done

Query RDF Data (SPARQL)

Querying RDF graphs

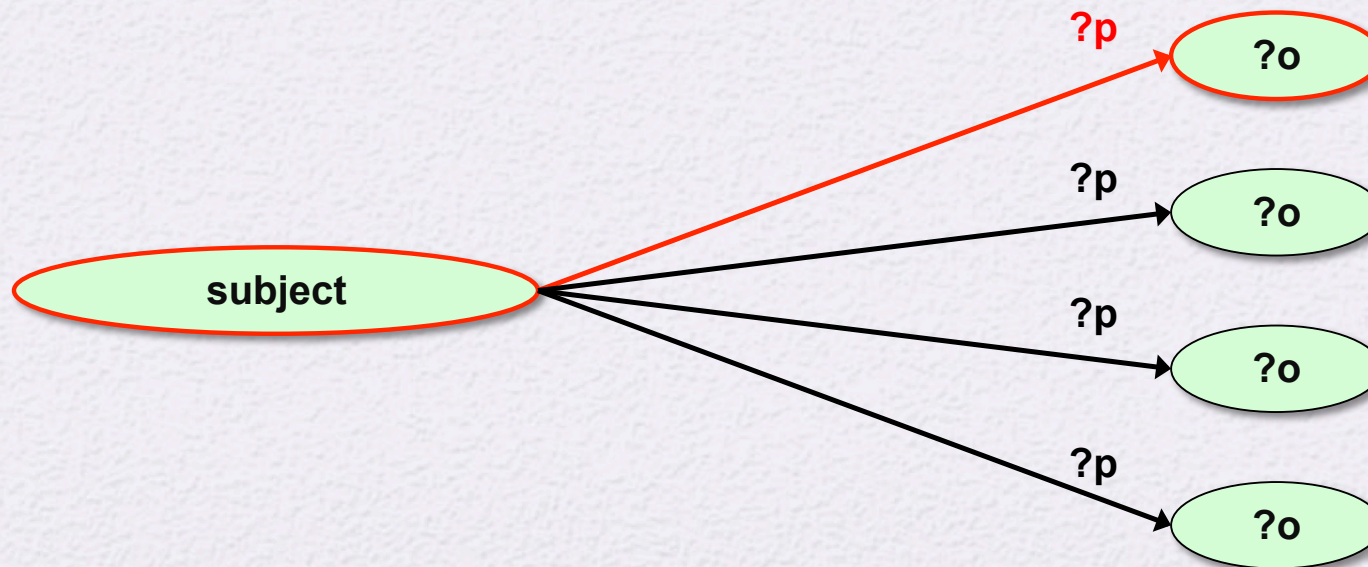
- ▶ Remember the Jena idiom:

```
StmtIterator iter=model.listStatements(subject,null,null);  
while(iter.hasNext()) {  
    st = iter.next();  
    p = st.getProperty(); o = st.getObject();  
    do_something(p,o);  
}
```

- ▶ In practice, more complex queries into the RDF data are necessary
 - ▶ something like “give me (a,b) pairs for which there is an x such that (x parent a) and (b brother x) holds” (ie, return the uncles)
- ▶ The goal of SPARQL (Query Language for RDF)

Analyze the Jena example

```
StmtIterator iter=model.listStatements(subject,null,null);  
while(iter.hasNext()) {  
    st = iter.next();  
    p = st.getProperty(); o = st.getObject();  
    do_something(p,o);  
}
```



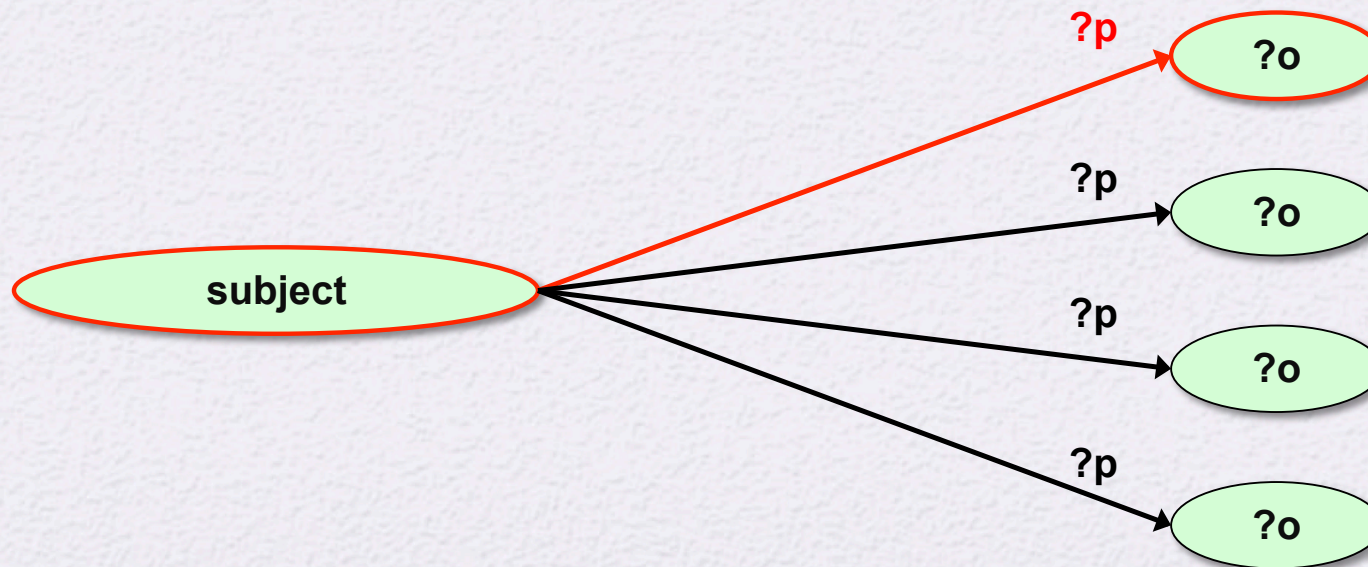
General: graph patterns

- ▶ The fundamental idea: use graph patterns
 - ▶ the pattern contains unbound symbols
 - ▶ by binding the symbols, subgraphs of the RDF graph are selected
 - ▶ if there is such a selection, the query returns bound resources

Our Jena example in SPARQL

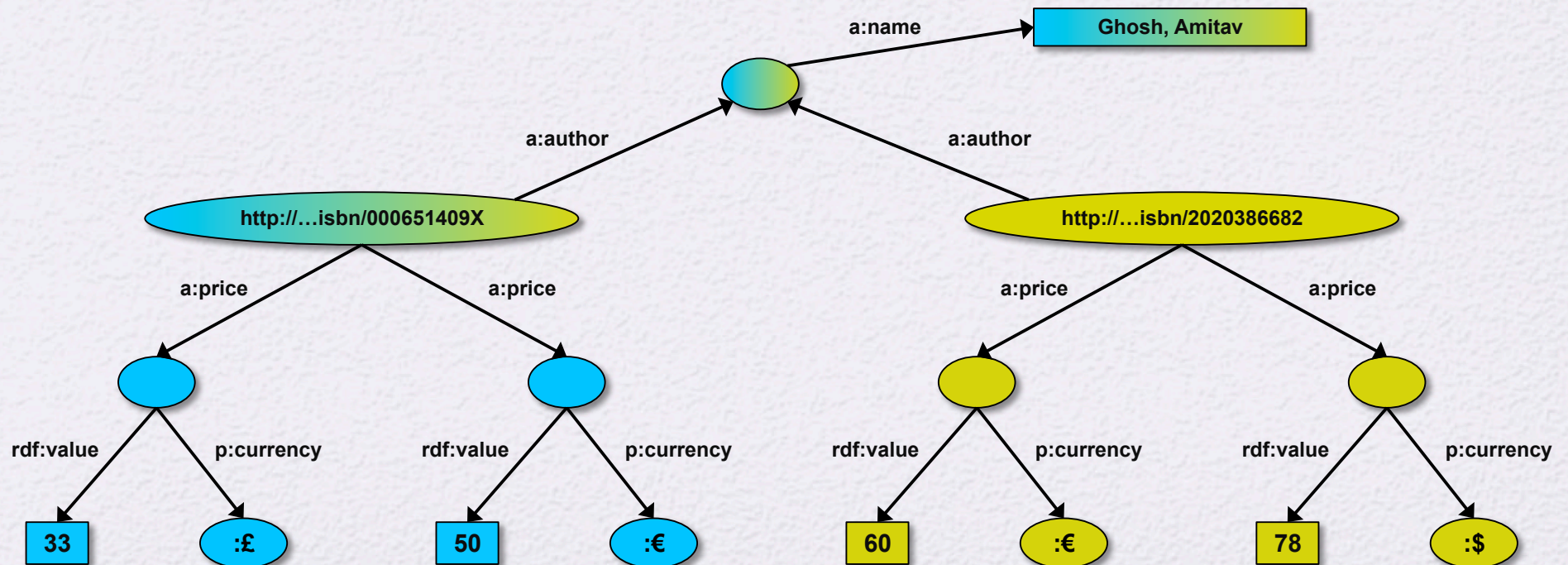
```
SELECT ?p ?o  
WHERE {subject ?p ?o}
```

- ▶ The triples in WHERE define the graph pattern, with ?p and ?o “unbound” symbols
- ▶ The query returns all p,o pairs



Simple SPARQL example

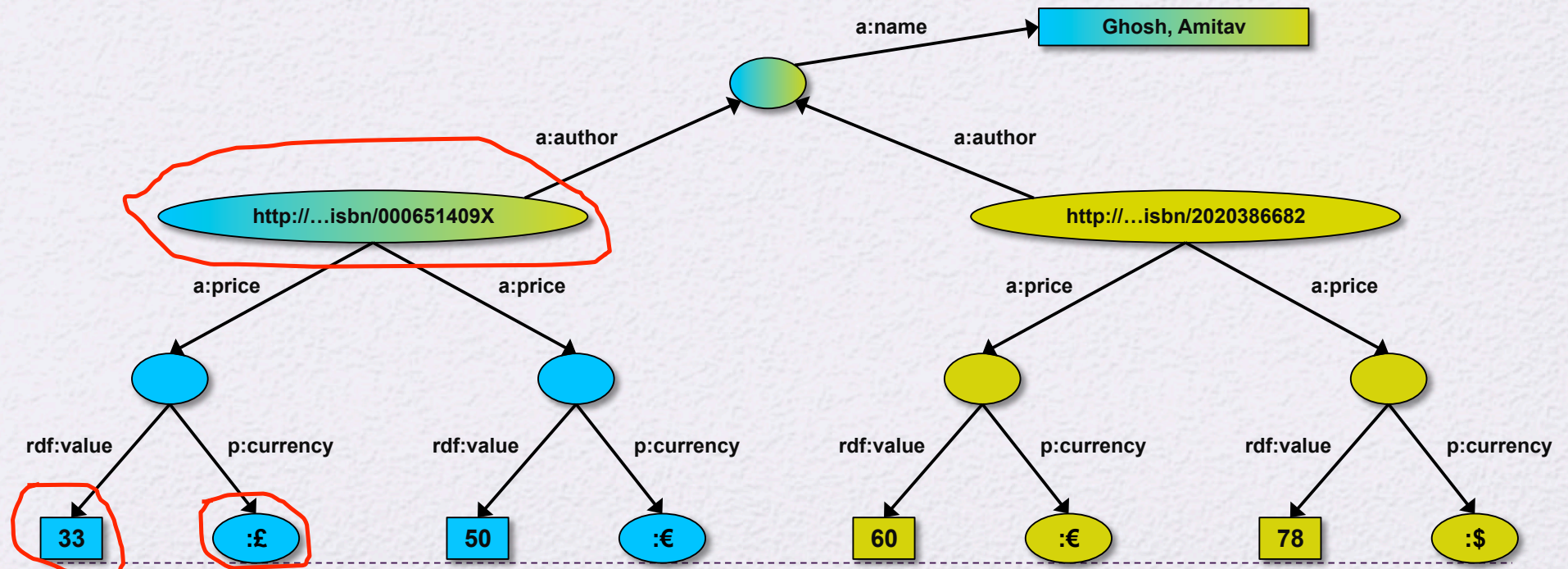
```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```



Simple SPARQL example

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```

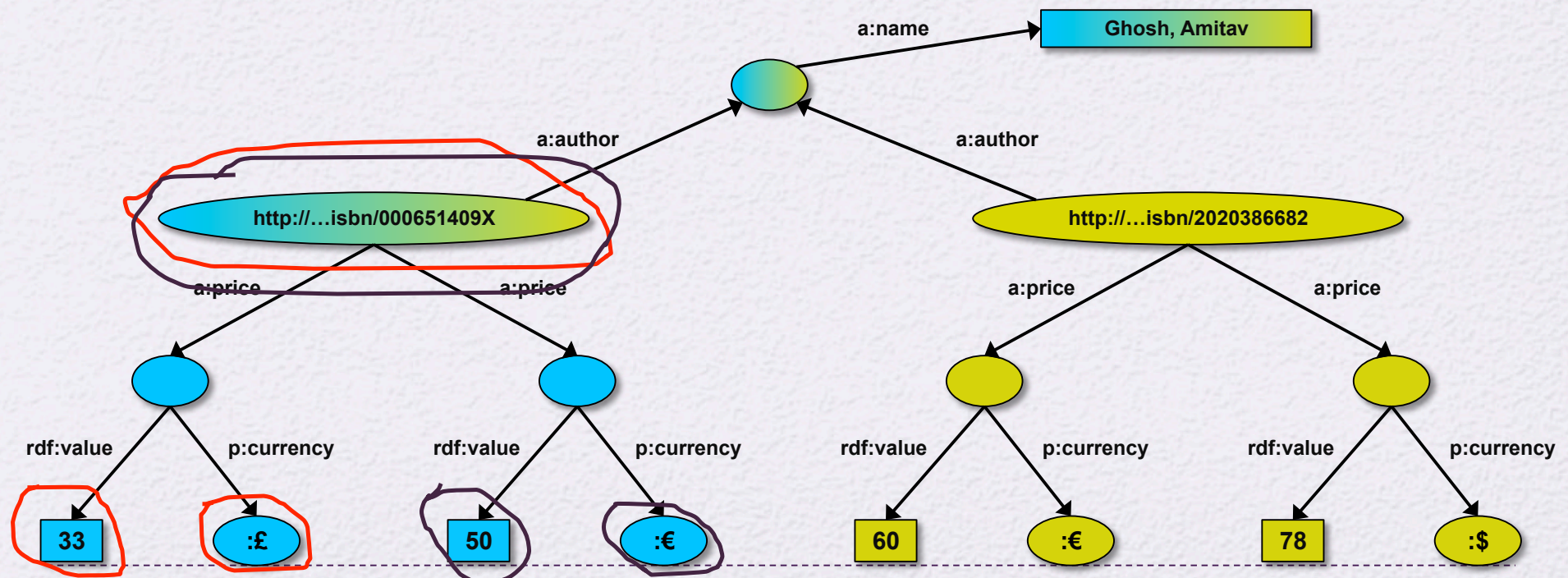
Returns: [**<...409X>**,33,:£]



Simple SPARQL example

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```

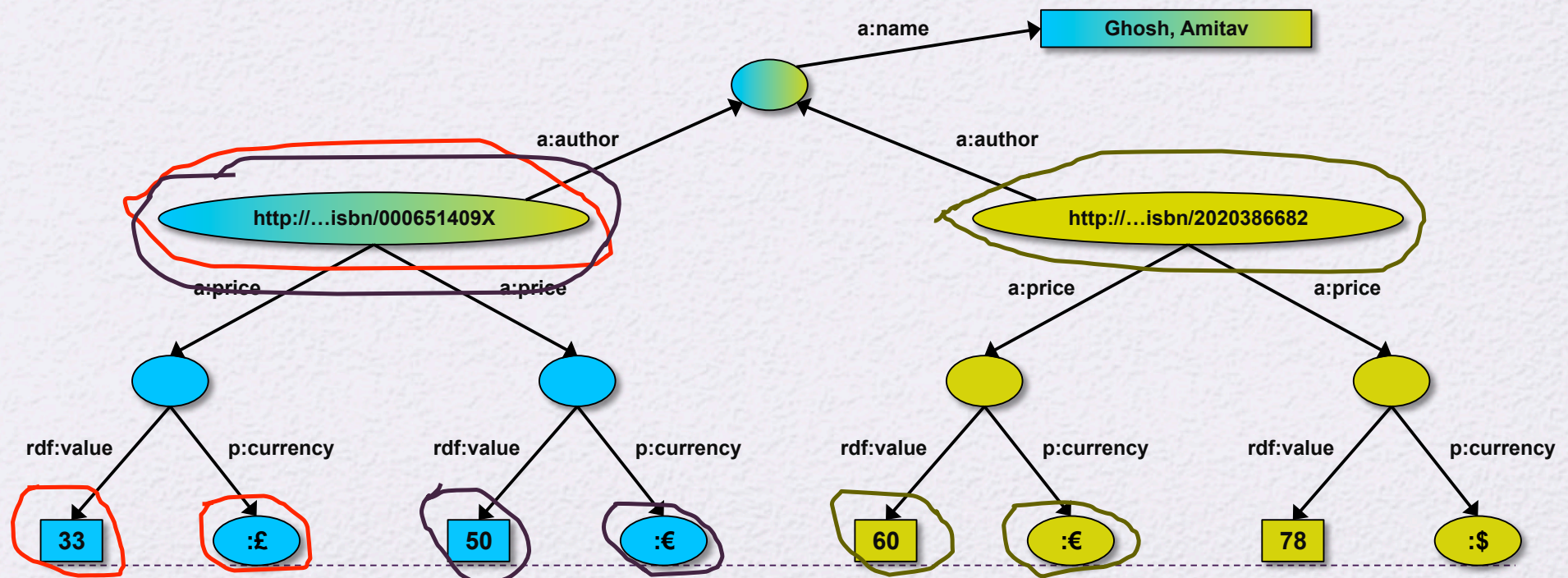
Returns: [**<...409X>,33,:£**], [**<...409X>,50,:€**]



Simple SPARQL example

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```

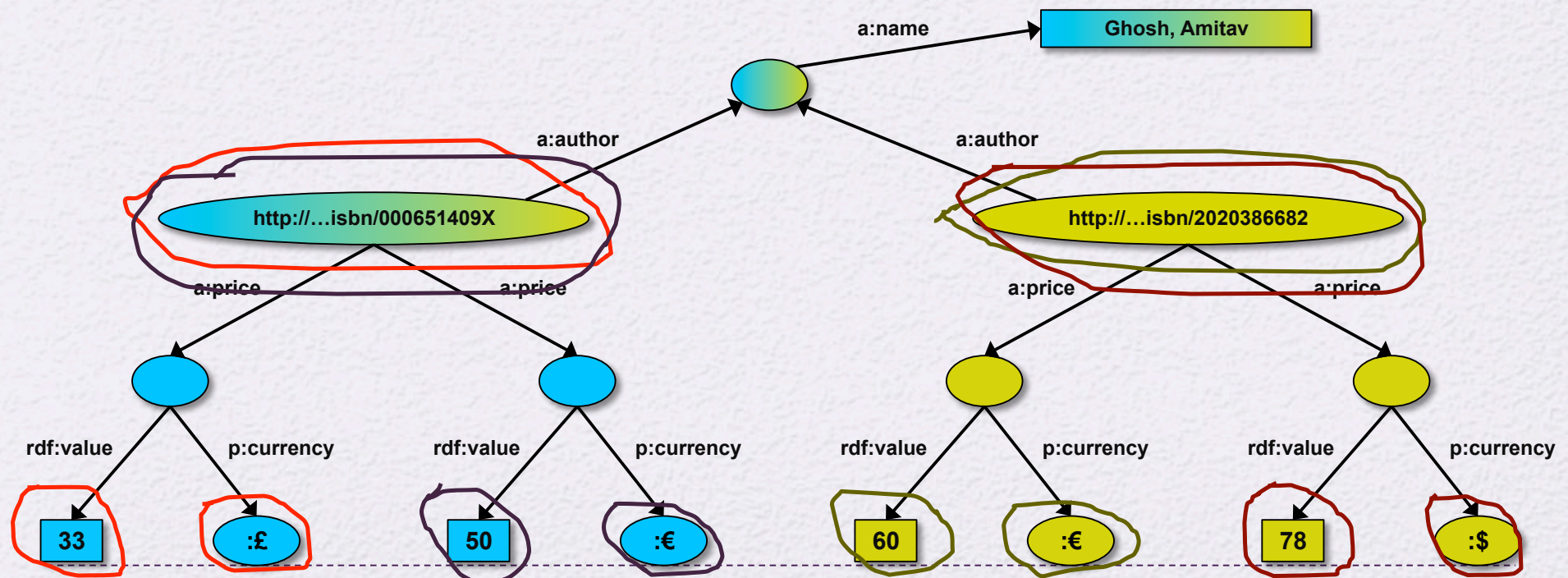
Returns: [**<...409X>,33,:£**], [**<...409X>,50,:€**],
[**<...6682>,60,:€**]



Simple SPARQL example

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```

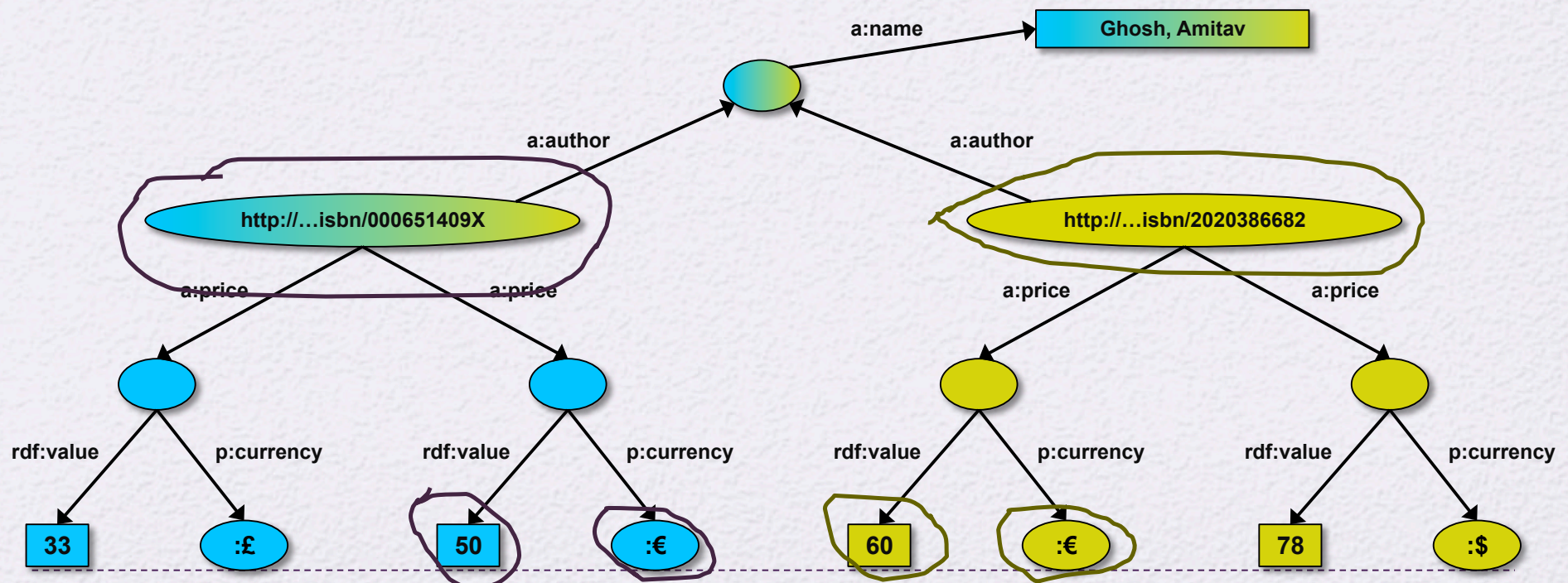
Returns: [**<...409X>,33,:£**], [**<...409X>,50,:€**],
[**<...6682>,60,:€**], [**<...6682>,78,:\$**]



Pattern constraints

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE { ?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.  
        FILTER(?currency == :€) }
```

Returns: [<...409X>,50,:€], [<...6682>,60,:€]



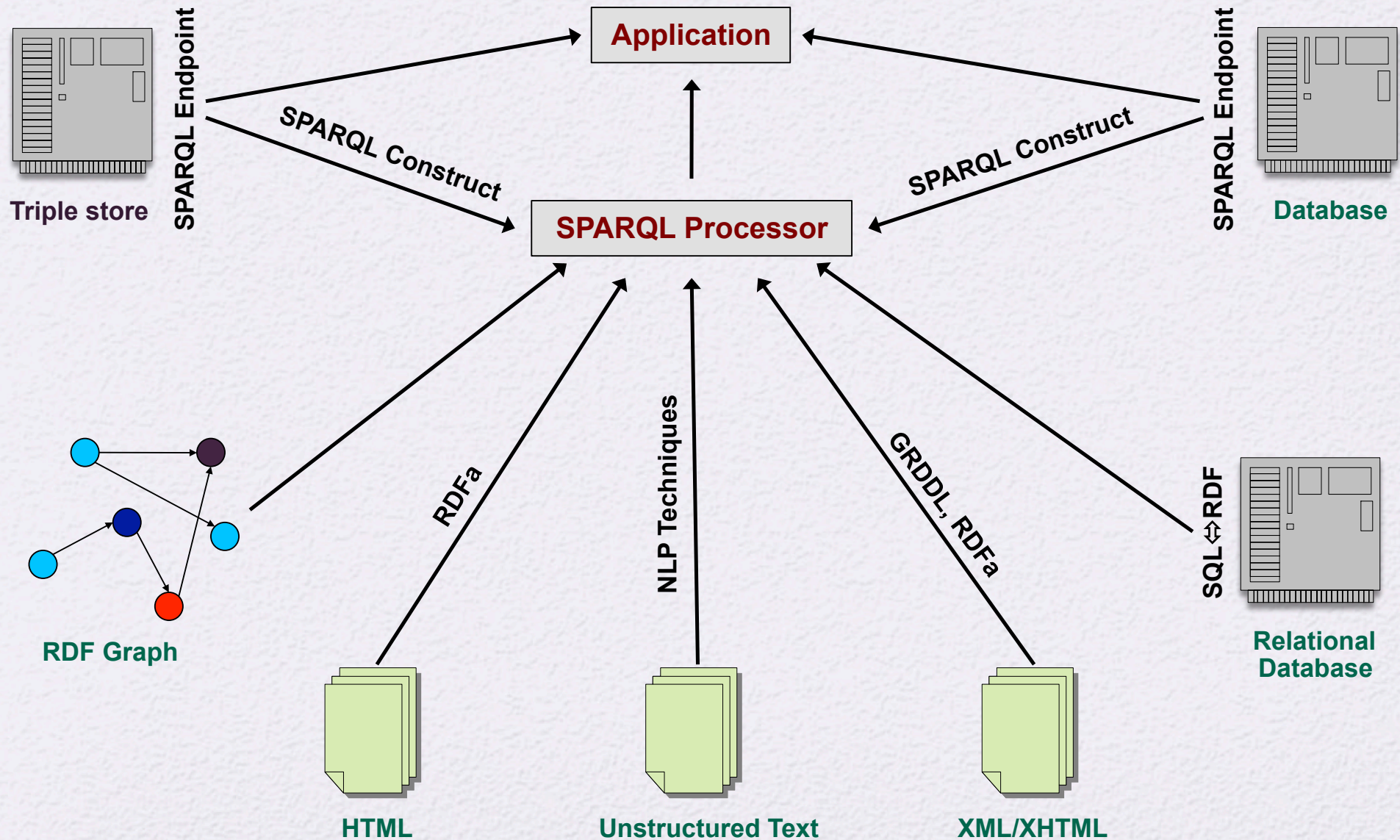
Many extra SPARQL features

- ▶ Limit the number of returned results; remove duplicates, sort them, ...
- ▶ Optional branches: if some part of the pattern does not match, ignore it
- ▶ Specify several data sources (via URI-s) within the query (essentially, a merge on-the-fly!)
- ▶ Construct a graph using a separate pattern on the query results
- ▶ In SPARQL 1.1: updating data, not only query

SPARQL usage in practice

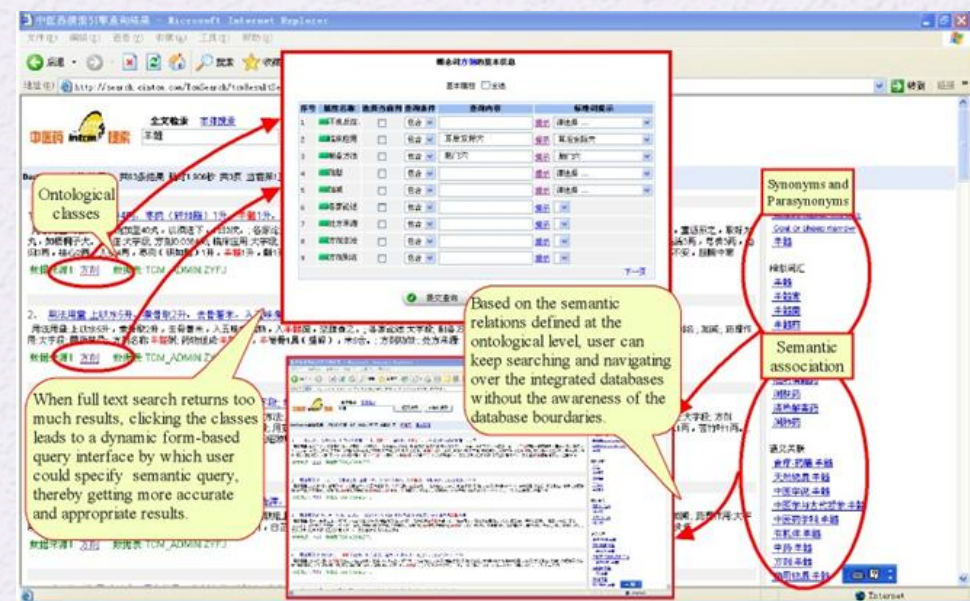
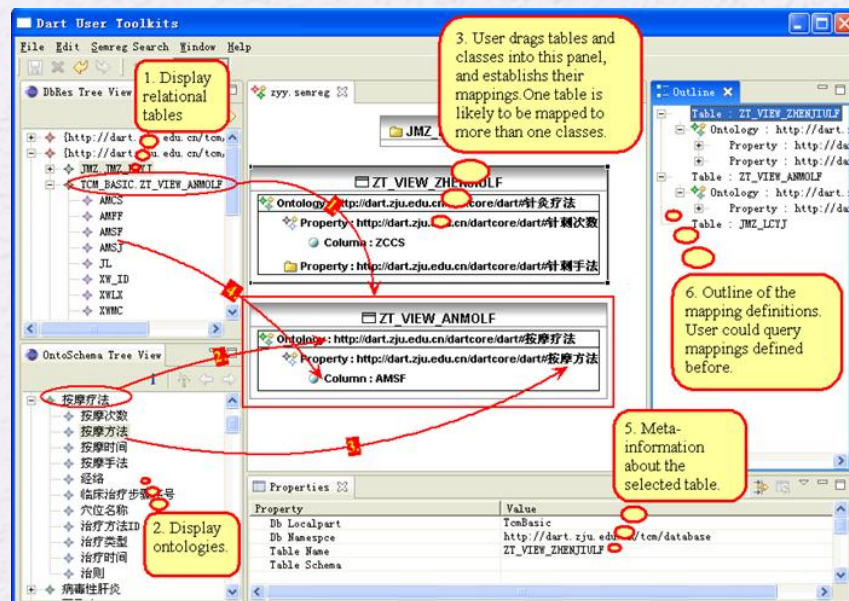
- ▶ SPARQL is usually used over the network
 - ▶ separate documents define the protocol and the result format
 - ▶ SPARQL Protocol for RDF with HTTP and SOAP bindings
 - ▶ SPARQL results in XML or JSON formats
- ▶ Big datasets often offer “SPARQL endpoints” using this protocol
 - ▶ typical example: SPARQL endpoint to DBpedia

SPARQL as a unifying point



Integrate knowledge for Chinese Medicine

- ▶ Integration of a large number of TCM databases
 - ▶ around 80 databases, around 200,000 records each



Vocabularies

Vocabularies

- ▶ Data integration needs agreements on
 - ▶ terms
 - ▶ “translator”, “author”
 - ▶ categories used
 - ▶ “Person”, “literature”
 - ▶ relationships among those
 - ▶ “an author is also a Person...”, “historical fiction is a narrower term than fiction”
 - ▶ ie, new relationships can be deduced

Vocabularies

- ▶ There is a need for “languages” to define such vocabularies
 - ▶ to define those vocabularies
 - ▶ to assign clear “semantics” on how new relationships can be deduced

But what about RDFS?

- ▶ Indeed RDFS *is* such framework:
 - ▶ there is typing, subtyping
 - ▶ properties can be put in a hierarchy
 - ▶ datatypes can be defined
- ▶ RDFS is enough for many vocabularies
- ▶ But not for all!

Three technologies have emerged

- ▶ To re-use thesauri, glossaries, etc: **SKOS**
- ▶ To define more complex vocabularies with a strong logical underpinning: **OWL**
- ▶ Generic framework to define rules on terms and data: **RIF**

Using thesauri, glossaries (SKOS)

SKOS

- ▶ Represent and share classifications, glossaries, thesauri, etc
 - ▶ for example:
 - ▶ Dewey Decimal Classification, Art and Architecture Thesaurus, ACM classification of keywords and terms...
 - ▶ classification/formalization of Web 2.0 type tags
- ▶ Define classes and properties to add those structures to an RDF universe
 - ▶ allow for a quick port of this traditional data, combine it with other data

Example: the term “Fiction”, as defined by the Library of Congress

The screenshot shows a web browser window displaying the Library of Congress Authorities & Vocabularies page for the term "Fiction". The browser's address bar shows the URL <http://id.loc.gov/authorities/sh85048050>. The page features the Library of Congress logo and navigation buttons for "ASK A LIBRARIAN", "DIGITAL COLLECTIONS", and "LIBRARY CATALOGS". A breadcrumb trail indicates the path: "The Library of Congress > Authorities & Vocabularies > Fiction". The main heading is "Authorities & Vocabularies". Below this, there is a "Return" link and a search section with a text input field labeled "Enter search terms..." and a "GO" button. Two tabs, "Details" and "Visualize", are visible, with "Visualize" currently selected. The content area displays the following information:

- Fiction**
- URI:** [<http://id.loc.gov/authorities/sh85048050#concept>](http://id.loc.gov/authorities/sh85048050#concept)
- Type:** Topical Term
- Alternate Labels:** Fiction--Philosophy; Metafiction; Novellas (Short novels); Novels; Stories
- Broader Terms:**
 - [Literature](#)
 - [Prose literature](#)
- Narrower Terms:**
 - [Adventure stories](#)
 - [Allegories](#)
 - [Alternative histories \(Fiction\)](#)
 - [Bildungsromans](#)
 - [Biographical fiction](#)

Example: the term “Fiction”, as defined by the Library of Congress

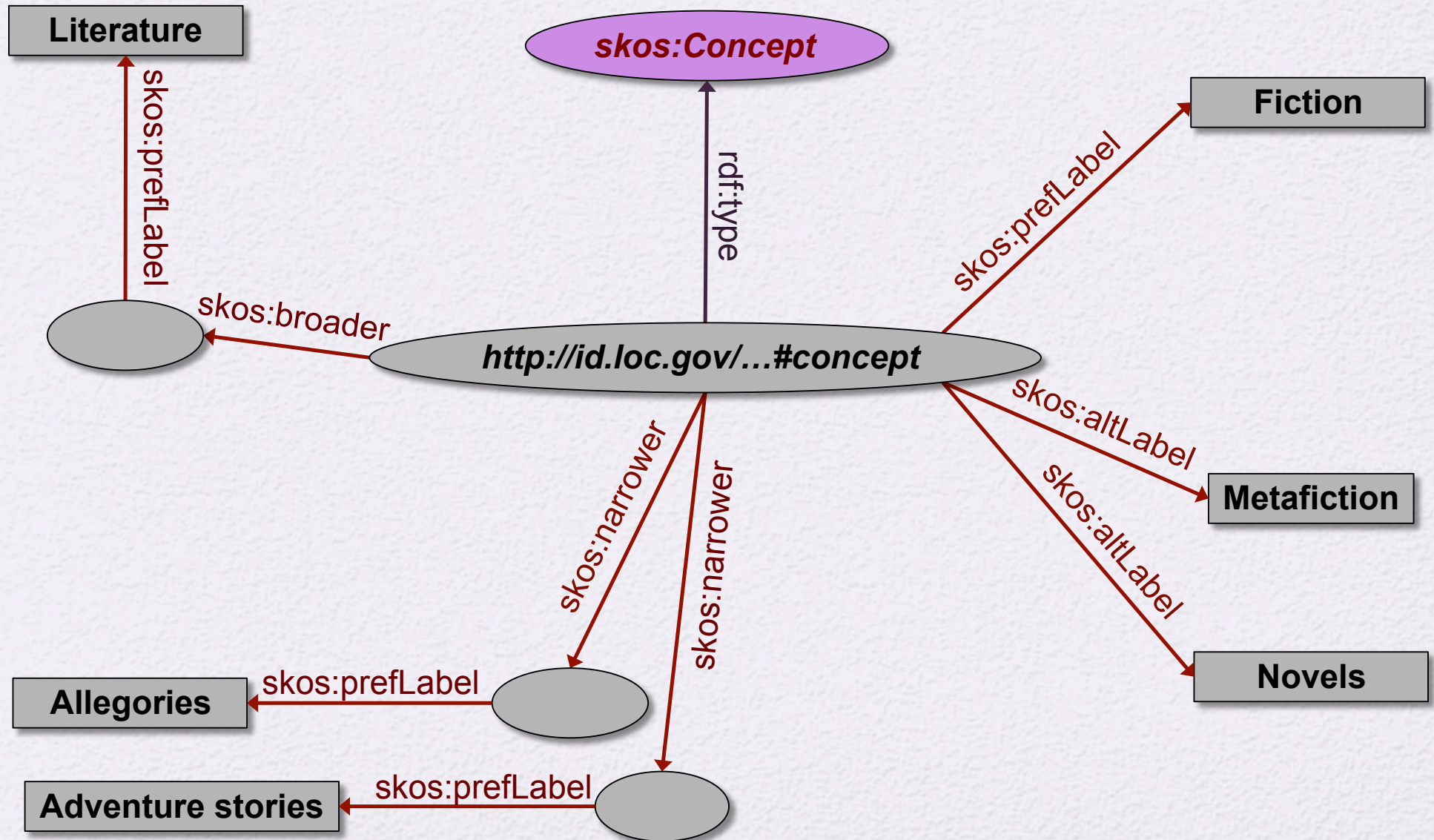
The screenshot shows a web browser window displaying the Library of Congress Authorities & Vocabularies page for the term 'Fiction'. The browser's address bar shows the URL <http://id.loc.gov/authorities/sh85048050>. The page features the Library of Congress logo and navigation buttons for 'ASK A LIBRARIAN', 'DIGITAL COLLECTIONS', and 'LIBRARY CATALOGS'. A breadcrumb trail indicates the path: 'The Library of Congress > Authorities & Vocabularies > Fiction'. The main heading is 'Authorities & Vocabularies'. Below this, there is a 'Return' link and a search box with the placeholder text 'Enter search terms...' and a 'GO' button. The page is divided into two tabs: 'Details' and 'Visualize'. The 'Details' tab is active, showing the following information:

- Fiction** (circled in red)
- URI:** <http://id.loc.gov/authorities/sh85048050#concept> (circled in red)
- Type:** Topical Term
- Alternate Labels:** Fiction--Philosophy; Metafiction; Novellas (Short novels); Novels; Stories (circled in red)
- Broader Terms:** (circled in red)
 - [Literature](#)
 - [Prose literature](#)
- Narrower Terms:** (circled in red)
 - [Adventure stories](#)
 - [Allegories](#)
 - [Alternative histories \(Fiction\)](#)
 - [Bildungsromans](#)
 - [Biographical fiction](#)

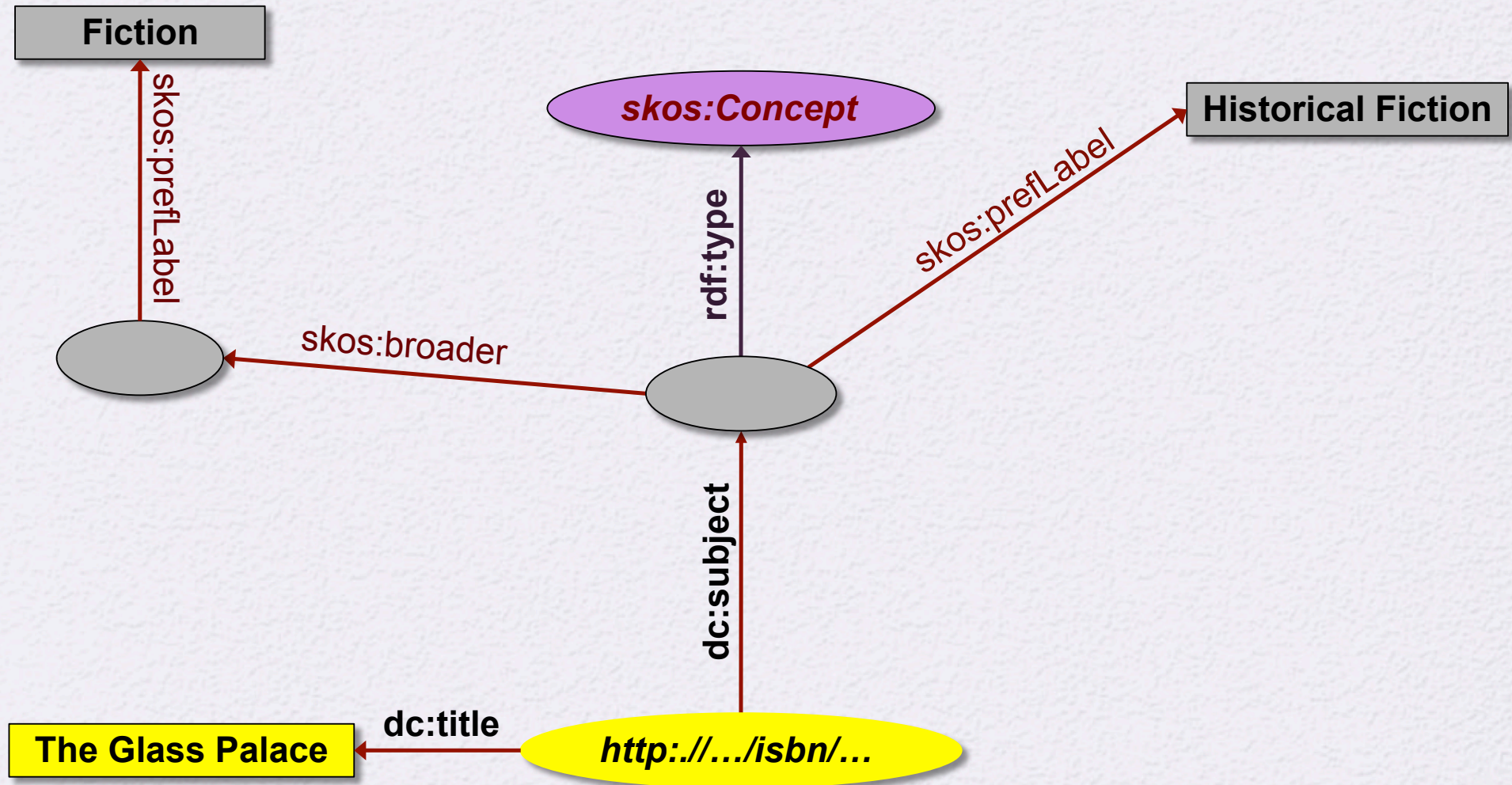
Thesauri have identical structures...

- ▶ The structure of the LOC page is fairly typical
 - ▶ label, alternate label, narrower, broader, ...
 - ▶ there is even an ISO standard for such structures
- ▶ SKOS provides a basic structure to create an RDF representation of these

LOC's "Fiction" in SKOS/RDF



Usage of the LOC graph



Importance of SKOS

- ▶ SKOS provides a simple bridge between the “print world” and the (Semantic) Web
- ▶ Thesauri, glossaries, etc, from the library community can be made available
 - ▶ LOC is a good example
- ▶ SKOS can also be used to organize tags, annotate other vocabularies, ...

Importance of SKOS

- ▶ Anybody in the World can refer to common concepts
 - ▶ they mean the same for everybody
- ▶ Applications may exploit the relationships among concepts
 - ▶ eg, SPARQL queries may be issued on the merge of the library data and the LOC terms

Semantic portal for art collections

Search results - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Jacco.van.Ossenbruggen@cwil.nl

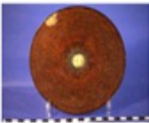
http://e-culture.multimedial.nl/demo/keyword?key=taichang iswc


The MultimediaN E-culture Search


Basic Search | Advanced Search | /facet | Relation Search | My Collection (en nl id ru uk zh)


Search results on **taichang** organized by category (47 results).


Works style/period.period has narrower with matching preferred label (46)



5664-2
(dc:creator unknown)


pop; speelgoedpop
(dc:creator unknown)



mingqi
(dc:creator unknown)



mingqi
(dc:creator unknown)


kommetje
(dc:creator unknown)


kommetje
(dc:creator unknown)

see more>


mingqi
(dc:creator unknown)


draagkist
(dc:creator unknown)

1360 1380 1400 1420 1440 1460 1480 1500 1520 1540 1560 1580 1600 1620 1640

5664-2
pop; speelgoedpop
mingqi

Ontologies

(OWL)

SKOS is not enough...

- ▶ SKOS may be used to provide simple vocabularies
- ▶ But it is not a complete solution
 - ▶ it concentrates on the concepts only
 - ▶ no characterization of properties in general
 - ▶ simple from a logical perspective
 - ▶ ie, few inferences are possible

Application may want more...

- ▶ **Complex applications may want more possibilities:**
 - ▶ characterization of properties
 - ▶ 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.

Web Ontology Language = OWL

- ▶ OWL is an extra layer, a bit like RDFS or SKOS
 - ▶ own namespace, own terms
 - ▶ it relies on RDF Schemas
- ▶ It is a separate recommendation
 - ▶ actually... there is a 2004 version of OWL (“OWL 1”)
 - ▶ and there is an update (“OWL 2”) published in 2009

OWL is complex...

- ▶ OWL is a large set of additional terms
- ▶ We will not cover the whole thing here...

Term equivalences

- ▶ 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?
 - ▶ owl:propertyDisjointWith

Term equivalences

- ▶ For individuals:
 - ▶ owl:sameAs: two URIs refer to the same concept (“individual”)
 - ▶ owl:differentFrom: negation of owl:sameAs

Other example: connecting to French



Typical usage of owl:sameAs

- ▶ Linking our example of Amsterdam from one data set (DBpedia) to the other (Geonames):

```
<http://dbpedia.org/resource/Amsterdam>  
  owl:sameAs <http://sws.geonames.org/2759793>;
```

- ▶ This is the main mechanism of “Linking” in the Linked Open Data project

Property characterization

- ▶ In OWL, one can characterize the behavior of properties (symmetric, transitive, functional, reflexive, inverse functional...)
- ▶ One property can be defined as the “inverse” of another

What this means is...

- ▶ If the following holds in our triples:

```
:email rdf:type owl:InverseFunctionalProperty.
```


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".
```


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, processed through OWL, the following holds, too:

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


Keys

- ▶ Inverse functional properties are important for identification of individuals
 - ▶ think of the email examples
- ▶ But... identification based on one property may not be enough

Keys

“if two persons have the same emails and the same homepages then they are identical”

- ▶ Identification is based on the identical values of two properties
- ▶ The rule applies to persons only

Previous rule in OWL

```
:Person rdf:type owl:Class;  
    owl:hasKey (:email :homepage) .
```


What it means is...

If:

```
<A> rdf:type :Person ;  
    :email    "mailto:a@b.c";  
    :homepage "http://www.ex.org".
```

```
<B> rdf:type :Person ;  
    :email    "mailto:a@b.c";  
    :homepage "http://www.ex.org".
```

then, processed through OWL, the following holds, too:

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


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

Enumerate class content

```
:Currency  
  rdf:type owl:Class;  
  owl:oneOf (:€ :£ :$) .
```

- I.e., the class consists of exactly of those individuals and nothing else

Union of classes

```
:Novel          rdf:type owl:Class.  
:Short_Story    rdf:type owl:Class.  
:Poetry         rdf:type owl:Class.  
:Literature     rdf:type owl:Class;  
                owl:unionOf (:Novel :Short_Story :Poetry) .
```

- Other possibilities: owl:complementOf, owl:intersectionOf, ...

For example...

If:

```
:Novel          rdf:type owl:Class.  
:Short_Story    rdf:type owl:Class.  
:Poetry         rdf:type owl:Class.  
:Literature     rdf:type owl:Class;  
                owl:unionOf (:Novel :Short_Story :Poetry).  
  
<myWork> rdf:type :Novel .
```

then the following holds, too:

```
<myWork> rdf:type :Literature .
```


It can be a bit more complicated...

If:

```
:Novel          rdf:type owl:Class.  
:Short_Story    rdf:type owl:Class.  
:Poetry         rdf:type owl:Class.  
:Literature     rdf:type owl:Class;  
                owl:unionOf (:Novel :Short_Story :Poetry) .  
  
fr:Roman owl:equivalentClass :Novel .  
  
<myWork> rdf:type fr:Roman .
```

then, through the combination of different terms, the following still holds:

```
<myWork> rdf:type :Literature .
```


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.
- ▶ Many inferred relationship can be found using a traditional rule engine

However... that may not be enough

- ▶ Very large vocabularies might require even more complex features
 - ▶ some major issues
 - ▶ the way classes (i.e., “concepts”) are defined
 - ▶ handling of datatypes like intervals
- ▶ OWL includes those extra features but... the inference engines become (much) more complex 😞

Example: property value restrictions

- ▶ New classes are created by restricting the property values on a class
- ▶ For example: how would I characterize a “listed price”?
 - ▶ it is a price that is given in one of the “allowed” currencies (€, £, or \$)
 - ▶ this defines a new class

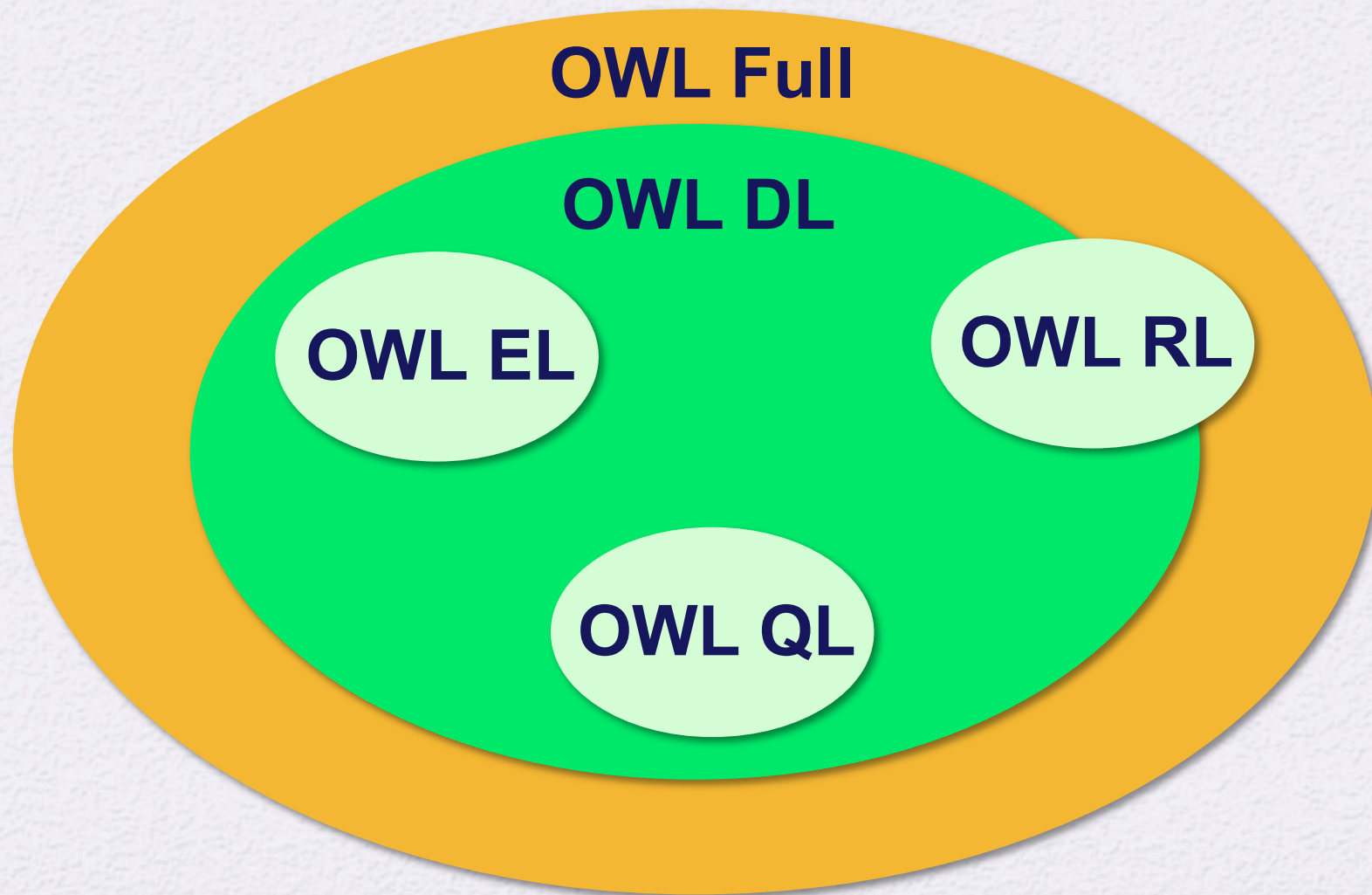
But: OWL is hard!

- ▶ The combination of class constructions with various restrictions is extremely powerful
- ▶ What we have so far follows the same logic as before
 - ▶ extend the basic RDF and RDFS possibilities with new features
 - ▶ define their semantics, ie, what they “mean” in terms of relationships
 - ▶ expect to infer new relationships based on those
- ▶ However... a full inference procedure is hard
 - ▶ not implementable with simple rule engines, for example 😞

OWL “species” or profiles

- ▶ OWL species comes to the fore:
 - ▶ restricting which terms can be used and under what circumstances (restrictions)
 - ▶ if one abides to those restrictions, then simpler inference engines can be used
- ▶ They reflect compromises: expressiveness vs. implementability

OWL Species



OWL RL

- ▶ Goal: to be implementable with rule engines
- ▶ Usage follows a similar approach to RDFS:
 - ▶ merge the ontology and the instance data into an RDF graph
 - ▶ use the rule engine to add new triples (as long as it is possible)

What can be done in OWL RL?

- ▶ Many features are available:
 - ▶ identity of classes, instances, properties
 - ▶ subproperties, subclasses, domains, ranges
 - ▶ union and intersection of classes (but with some restrictions)
 - ▶ property characterizations (functional, symmetric, etc)
 - ▶ property chains
 - ▶ keys
 - ▶ *some* property restrictions
- ▶ All examples so far could be inferred with OWL RL!

Improved Search via Ontology (GoPubMed)

- ▶ Search results are re-ranked using ontologies
- ▶ related terms are highlighted

The screenshot displays the GoPubMed web interface within a Mozilla Firefox browser window. The left sidebar, titled 'what', shows a hierarchical ontology tree. The 'Diseases' category is expanded, and 'Tinnitus' is highlighted with a red circle. The main content area shows search results for 'tinnitus', with 1,000 articles found. The results are ranked and displayed with related terms highlighted in yellow. A blue arrow points to the first result, '5: Pros and cons of tinnitus retraining therapy.', which has a PMID of 18368566. Other results include '1: Gabapentin effectiveness on the sensation of subjective idiopathic tinnitus : a pilot study.', '3: Algorithm for evaluation of pulsatile tinnitus.', and '4: Functional imaging of unilateral tinnitus using fMRI.' The interface includes a search bar, a 'find it!' button, and a 'goPubMed' logo.

Improved Search via Ontology (Go3R)

- ▶ Same dataset, different ontology
- ▶ (ontology is on non-animal experimentation)

The screenshot displays the Go3R web application interface within a Mozilla Firefox browser window. The left sidebar, titled 'what', shows a hierarchical ontology tree. Two categories are circled in red: 'Bioethics [102]' and 'Reduction [90]'. The main content area shows search results for the query 'tinnitus', indicating '1,000 articles' and 'differences (P > 0.05)'. A list of article titles is displayed, including '2: Microvascular decompression of cochleovestibular nerve.', '3: Algorithm for evaluation of pulsatile tinnitus.', '4: Functional imaging of unilateral tinnitus using fMRI.', and '5: Pros and cons of tinnitus retraining therapy.'. A blue arrow points to the word 'tinnitus' in the title of article 4. The bottom of the interface shows a 'Find related categories ...' section.

Rules

(RIF)

Why rules on the Semantic Web?

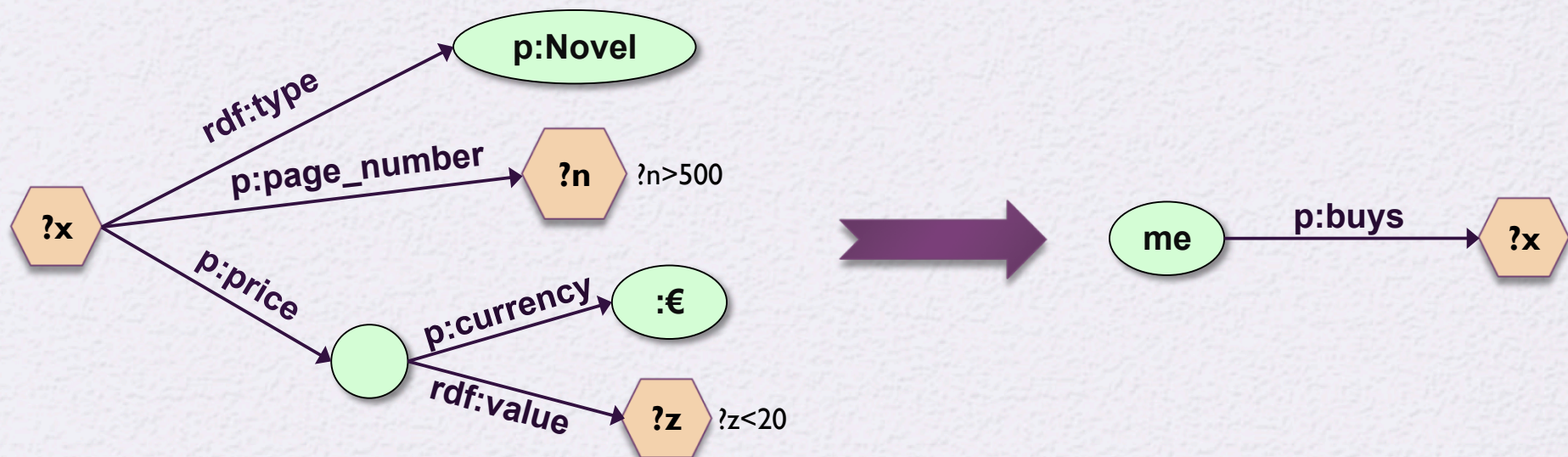
- ▶ Some conditions may be complicated in ontologies (ie, OWL)
 - ▶ eg, Horn rules: $(P1 \ \& \ P2 \ \& \ \dots) \rightarrow C$
- ▶ In many cases applications just want 2-3 rules to complete integration
- ▶ Ie, rules may be an alternative to (OWL based) ontologies

Things you may want to express

- ▶ An example from our bookshop integration:
 - ▶ “I buy a novel with over 500 pages if it costs less than €20”
 - ▶ something like (in an ad-hoc syntax):

```
{
  ?x rdf:type p:Novel;
    p:page_number ?n;
    p:price [
      p:currency :€;
      rdf:value ?z
    ].
  ?n > "500"^^xsd:integer.
  ?z < "20.0"^^xsd:double.
}
=>
{ <me> p:buys ?x }
```


Things you may want to express



RIF (Rule Interchange Format)

- ▶ The goals of the RIF work:
 - ▶ define simple rule language(s) for the (Semantic) Web
 - ▶ define interchange formats for rule based systems
- ▶ RIF defines several “dialects” of languages
- ▶ RIF is not bound to RDF only
 - ▶ eg, relationships may involve more than 2 entities
 - ▶ there are dialects for production rule systems

RIF Core

- ▶ The simplest RIF “dialect”
- ▶ A Core document is
 - ▶ directives like import, prefix settings for URI-s, etc
 - ▶ a sequence of logical implications

RIF Core example

```
Document(  
  Prefix(cpt http://example.com/concepts#)  
  Prefix(person http://example.com/people#)  
  Prefix(isbn http://.../isbn/)  
  
  Group  
  (  
    Forall ?Buyer ?Book ?Seller (  
      cpt:buy(?Buyer ?Book ?Seller):- cpt:sell(?Seller ?Book ?Buyer)  
    )  
    cpt:sell(person:John isbn:000651409X person:Mary)  
  )  
)
```

This infers the following relationship:

```
cpt:buy(person:Mary isbn:000651409X person:John)
```


Expressivity of RIF Core

- ▶ Formally: definite Horn without function symbols, a.k.a. “Datalog”
 - ▶ eg, $p(a,b,c)$ is fine, but $p(f(a),b,c)$ is not
- ▶ Includes some extra features
 - ▶ built-in datatypes and predicates
 - ▶ “local” symbols, a bit like blank nodes

Expressivity of RIF Core

- ▶ There are also “safeness measures”
 - ▶ eg, variable in a consequent should be in the antecedent
 - ▶ this secures a straightforward implementation strategy (“forward chaining”)

RIF Syntaxes

- ▶ RIF defines
 - ▶ a “presentation syntax”
 - ▶ a standard XML syntax to encode and exchange the rules
 - ▶ there is a draft for expressing Core in RDF
 - ▶ just like OWL is represented in RDF

What about RDF and RIF?

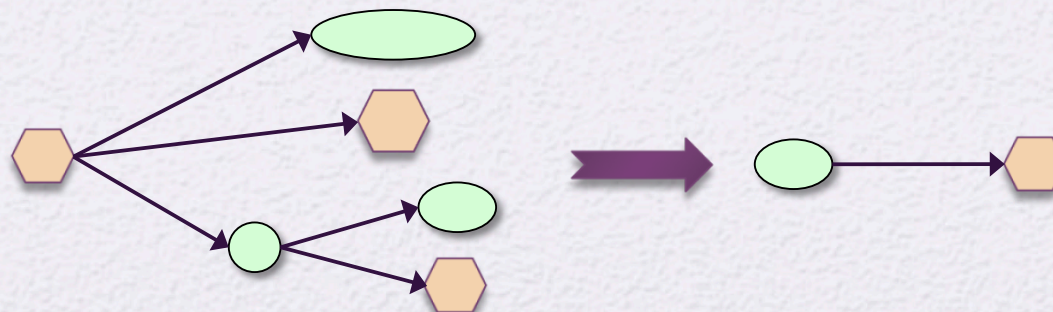
- ▶ **Typical scenario:**
 - ▶ the “data” of the application is available in RDF
 - ▶ rules on that data is described using RIF
 - ▶ the two sets are “bound” (eg, RIF “imports” the data)
 - ▶ a RIF processor produces new relationships

To make RIF/RDF work

- ▶ Some technical issues should be settled:
 - ▶ RDF triples have to be representable in RIF
 - ▶ various constructions (typing, datatypes, lists) should be aligned
 - ▶ the semantics of the two worlds should be compatible
- ▶ There is a separate document that brings these together

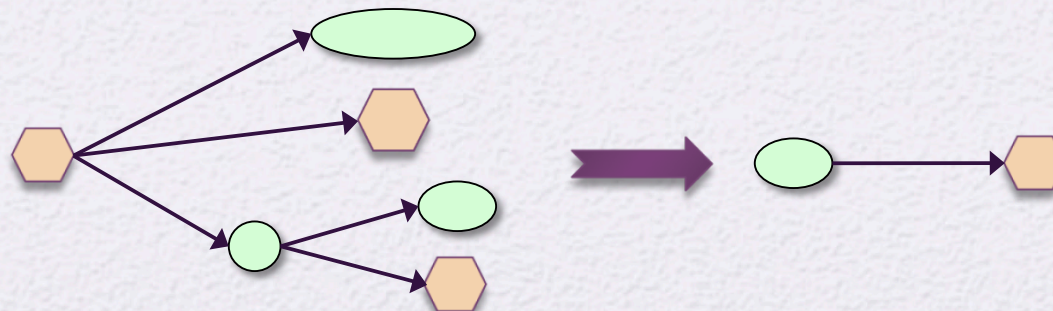
Remember the what we wanted from Rules?

```
{  
  ?x rdf:type p:Novel;  
    p:page_number ?n;  
    p:price [  
      p:currency :€;  
      rdf:value ?z  
    ].  
  ?n > "500"^^xsd:integer.  
  ?z < "20.0"^^xsd:double.  
}  
=>  
{ <me> p:buys ?x }
```



The same with RIF Presentation syntax

```
Document (  
  Prefix ...  
  Group (  
    Forall ?x ?n ?z (  
      <me>[p:buys->?x] :-  
        And(  
          ?x rdf:type p:Novel  
          ?x[p:page_number->?n p:price->_abc]  
          _abc[p:currency->:€ rdf:value->?z]  
          External( pred:numeric-greater-than(?n "500"^^xsd:integer) )  
          External( pred:numeric-less-than(?z "20.0"^^xsd:double) )  
        )  
      )  
    )  
  )  
)
```



Discovering new relationships...

```
Forall ?x ?n ?z (
  <me>[p:buys->?x] :-
    And(
      ?x # p:Novel
      ?x[p:page_number->?n p:price->_abc]
      _abc[p:currency->:€ rdf:value->?z]
      External( pred:numeric-greater-than(?n "500"^^xsd:integer) )
      External( pred:numeric-less-than(?z "20.0"^^xsd:double) )
    )
)
```


Discovering new relationships...

```
Forall ?x ?n ?z (
  <me>[p:buys->?x] :-
    And(
      ?x # p:Novel
      ?x[p:page_number->?n p:price->_abc]
      _abc[p:currency->:€ rdf:value->?z]
      External( pred:numeric-greater-than(?n "500"^^xsd:integer) )
      External( pred:numeric-less-than(?z "20.0"^^xsd:double) )
    )
)
```

combined with:

```
<http://.../isbn/...> a p:Novel;
  p:page_number "600"^^xsd:integer ;
  p:price [ rdf:value "15.0"^^xsd:double ; p:currency :€ ] .
```


Discovering new relationships...

```
forall ?x ?n ?z (
  <me>[p:buys->?x] :-
    And(
      ?x # p:Novel
      ?x[p:page_number->?n p:price->_abc]
      _abc[p:currency->:€ rdf:value->?z]
      External( pred:numeric-greater-than(?n "500"^^xsd:integer) )
      External( pred:numeric-less-than(?z "20.0"^^xsd:double) )
    )
)
```

combined with:

```
<http://.../isbn/...> a p:Novel;
  p:page_number "600"^^xsd:integer ;
  p:price [ rdf:value "15.0"^^xsd:double ; p:currency :€ ] .
```

yields:

```
<me> p:buys <http://.../isbn/...> .
```


RIF vs. OWL?

- ▶ The expressivity of the two is fairly identical
 - ▶ the emphasis are a bit different
- ▶ Using rules vs. ontologies may largely depend on
 - ▶ available tools
 - ▶ personal technical experience and expertise
 - ▶ taste...

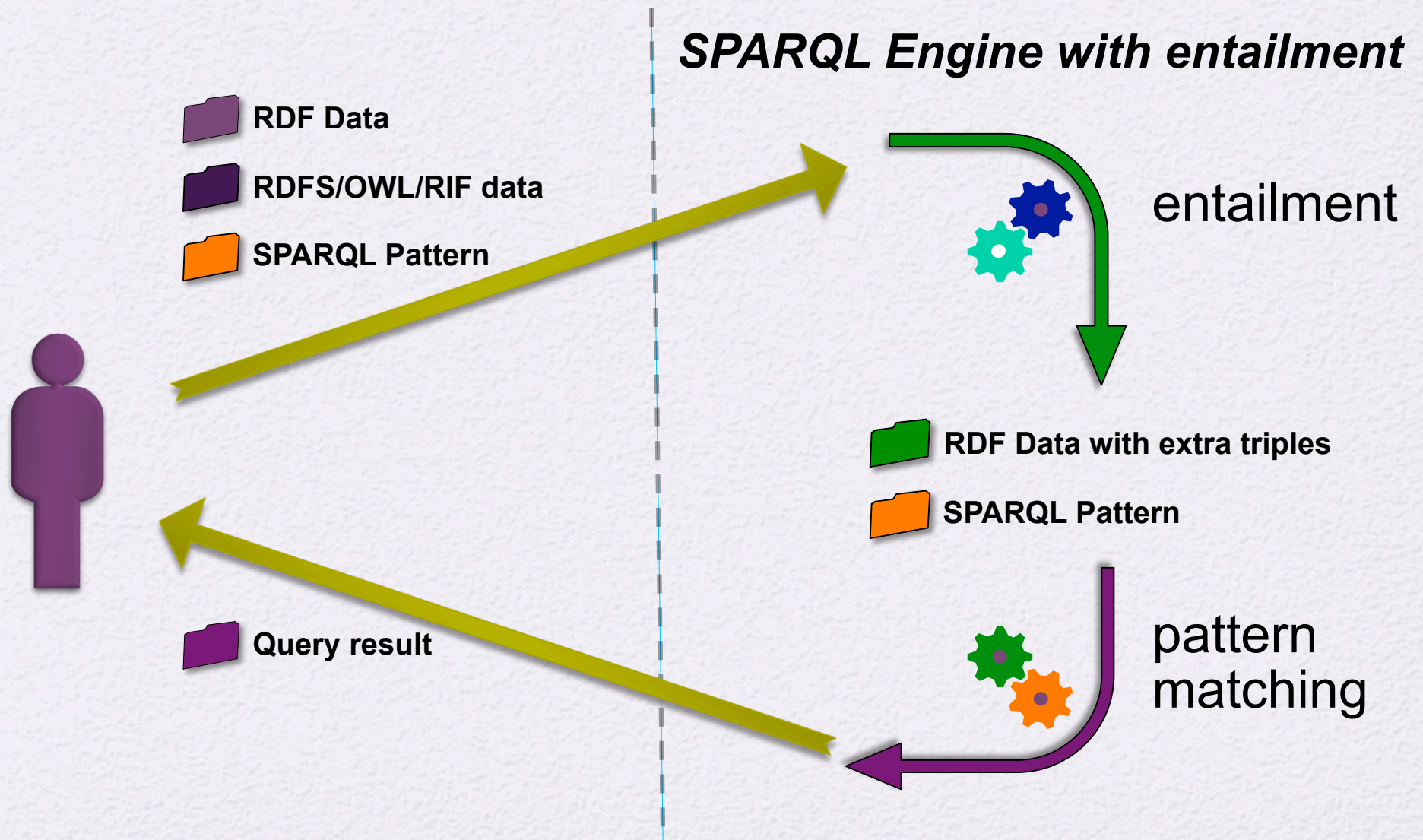
What about OWL RL?

- ▶ OWL RL stands for “Rule Language”...
- ▶ OWL RL is in the intersection of RIF Core and OWL
 - ▶ inferences in OWL RL can be expressed with RIF rules
 - ▶ RIF Core engines can act as OWL RL engines

Inferencing and SPARQL

- ▶ Question: how do SPARQL queries and inferences work together?
 - ▶ RDFS, OWL, and RIF produce new relationships
 - ▶ on what data do we query?
- ▶ Answer: in current SPARQL, that is not defined 😞
- ▶ But, in SPARQL 1.1 it is... 😊

SPARQL 1.1 and RDFS/OWL/RIF



What have we achieved?

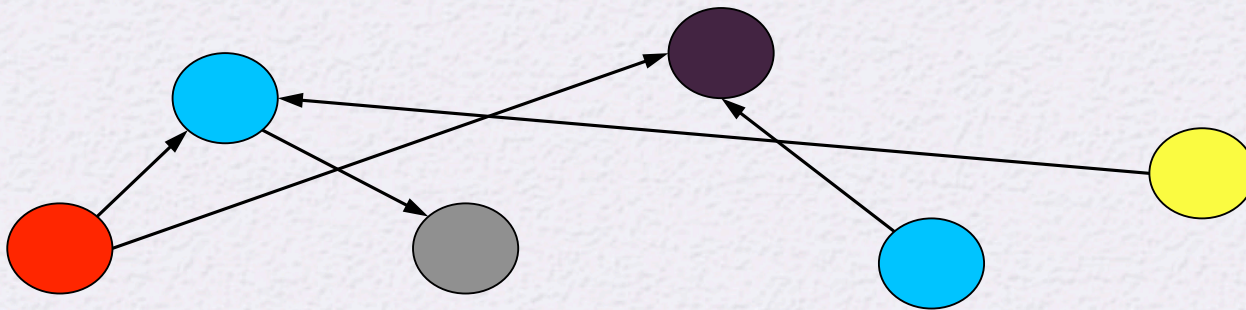
(putting all this together)

Remember the integration example?



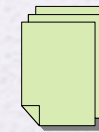
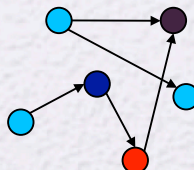
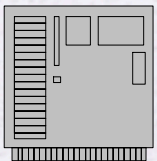
Applications

Manipulate
Query
...



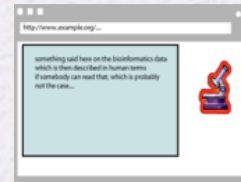
Data represented in abstract format

Map,
Expose,
...



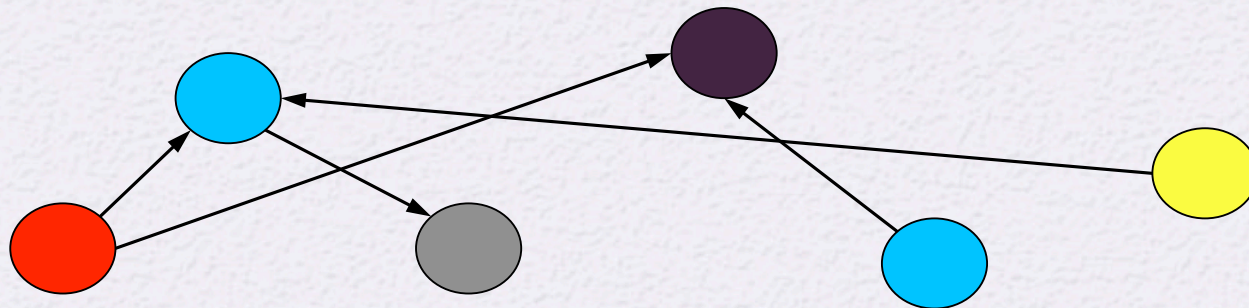
Data in various formats

Same with what we learned



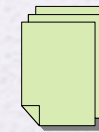
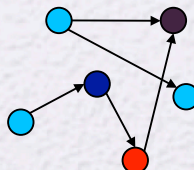
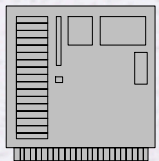
Applications

SPARQL,
Inferences
...



Data represented in RDF with extra knowledge (RDFS, SKOS, RIF, OWL,...)

RDB \Leftrightarrow RDF,
GRDL, RDFa,
...



Data in various formats

eTourism: provide personalized itinerary

The screenshot displays the Zaragoza Turismo website interface. At the top, there's a navigation bar with 'Zaragoza TURISMO' and a header with 'THE CITY COUNCIL', 'FOR PEOPLE', and 'THE CITY'. A sidebar on the left lists various services like 'Tourist Services', 'What to visit', 'Guided Visit', etc. The main content area is titled 'ITINERARY FOR 17/06/08' and shows a 'Proposed route' for that day. It includes a 'Morning' section with a list of tourist sites (e.g., Basilica of the Pilar, Ibercaja Camón Aznar Museum) and an 'Afternoon' section with another list of sites (e.g., Church of la Mantería, Church of San Ildefonso). A 'Zaragoza street plan' section features a map of the city center with numbered markers corresponding to the itinerary. The map includes labels for 'Google Maps', 'IDEEZar', and 'Mapa', 'Satélite', 'Híbrido'. A detailed description of the 'BASILICA OF THE PILAR' is provided, including its full accessibility and historical context. The interface also includes interactive elements like 'Click on each of the monument names to read detailed information below' and buttons for 'Back to Create'.

- Integration of relevant data in Zaragoza (using RDF and ontologies)
- Use rules on the RDF data to provide a proper itinerary

Available documents, resources

Available specifications: Primers, Guides

- ▶ The “RDF Primer” and the “OWL Guide” give a formal introduction to RDF(S) and OWL
- ▶ SKOS has its separate “SKOS Primer”
- ▶ GRDDL Primer and RDFa Primer have been published
- ▶ The W3C [Semantic Web Activity Wiki](#) has links to all the specifications

“Core” vocabularies

- ▶ There are also a number “core vocabularies”
 - ▶ 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

- ▶ J. Pollock: Semantic Web for Dummies, 2009
- ▶ 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
- ▶ P. Hitzler, R. Sebastian, M. Krötzsch: Foundation of Semantic Web Technologies, 2009
- ▶ ...

See the separate [Wiki page collecting book references](#)

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, Talis Platform, ...
- RDF Gateway, RDFLib, Open Anzo, DartGrid, Zitgist, Ontotext, Protégé, ...
- Thetus publisher, SemanticWorks, SWI-Prolog, RDFStore...
- ...

Further information

- ▶ Planet RDF aggregates a number of SW blogs:
 - ▶ <http://planetrdf.com/>
- ▶ Semantic Web Interest Group
 - ▶ a forum developers with archived (and public) mailing list, and a constant IRC presence on [#swig](http://freenode.net) on freenode.net
 - ▶ anybody can sign up on the list
 - ▶ <http://www.w3.org/2001/sw/interest/>

Thank you for your attention!

These slides are also available on the Web:

<http://www.w3.org/2010/Talks/0622-SemTech-IH/>

