Introduction to Linked Open Data

Tutorial
DC-2013 Conference
2nd September, 2013, Lisbon, Portugal
Ivan Herman, W3C
These slides are also available on the Web:

http://www.w3.org/2013/Talks/0902-Lisbon-IH/
Introduction
Situé à l’extrémité Nord-Est du département, les massifs du Concors et de la Sainte-Victoire forment un grand ensemble forestier du Sud de la
Le Pavillon D'Or

Published October 1, 1975 by French & European Pubns.

The Physical Object
Format Paperback

ID Numbers
Open Library OL11039953M
ISBN 10 0785940391
ISBN 13 9780785940395
Goodreads 756054

Read
No readable version available.

Borrow
Physical copy, local WorldCat

Buy
Alibris
Amazon
AbeBooks
Biblio.com
Book Depository
Powells

Lists

Manage Covers

Connecting to www.archive.org...
Take a specific example
The Music site of the BBC
The Music site of the BBC

helped reggae reach a mass market. Two of his most popular recordings were "Layla", recorded by Derek and the Dominos, another band he formed, and Robert Johnson's "Crossroads", recorded by Cream. Following the death of his son Conor in 1991, Clapton's grief was expressed in the song "Tears in Heaven", which featured in his Unplugged album.

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
Youtube at youtube.com/user/ericclapton
Twitter at twitter.com/EricCleptonNews
MySpace at myspace.com/ericclapton
Wikipedia article on Eric Clapton
Last.fm page on Eric Clapton
Discogs at discogs.com/artist/Eric Clapton
MusicBrainz entry on Eric Clapton


Collaborated on: The Dirty Mac (1963)
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- And the site gets soon out-of-date 😞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
  - I.e, write some code to incorporate the new data
- Easily get out of date again… 😞
Editors roam the Web for new data via API-s

Understand those…
- input, output arguments, datatypes used, etc
- hope that the necessary portion of the data is released through the API

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, e.g., 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...
And this is no secret...
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"
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...
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
  - i.e., data can be integrated over the Web
- This is where Linked Data come in
i.e.,... connect the silos
Example: Amsterdam fire brigade routing

- Find the best possible route from the station to the fire
  - e.g., where are the roadblocks?
- Use and integrate available city data
- Also: republish the structured data for others to use!

Courtesy of Bart van Leeuwen, Amsterdam Fire Service, The Netherlands
In what follows...

- We will use a simplistic example to introduce the main technical 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, e.g., queries on the whole!
  - queries not possible on the individual data sets
We start with a book...

'A DOCTOR ZHIVAGO FOR THE FAR EAST' *THE INDEPENDENT*

Amitav Ghosh

**THE GLASS PALACE**

The magnificent, poignant, fascinating novel of three generations that starts in Mandalay...
# A simplified bookstore data (dataset “A”)

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Homepage</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Publisher's name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_qpr</td>
<td>Harper Collins</td>
<td>London</td>
</tr>
</tbody>
</table>
1st: export your data as a set of relations

- The Glass Palace
- a:title
- 2000
- a:year
- London
- a:city
- Harper Collins
- a:p_name
- Ghosh, Amitav
- a:name
- http://www.amitavghosh.com
- a:homepage
- http://…isbn/000651409X
- a:author
- a:publisher
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
Same book in French...
Another bookstore data (dataset “F”)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>Titre</td>
<td>Traducteur</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ID</td>
<td>Auteur</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ISBN 0–00–6511409–X</td>
<td>$A11$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Nom</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Ghosh, Amitav</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Besse, Christianne</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2nd: export your second set of data

http://…isbn/000651409X

Le palais des miroirs

http://…isbn/2020386682

Ghosh, Amitav

Besse, Christianne

f:auteur

f:nom

f:original

f:titre

f:traducteur

f:nom
3rd: start merging your data

- The Glass Palace
- 2000
- London
- Harper Collins
- Ghosh, Amitav

- http://www.amitavghosh.com
- http://isbn/000651409X

- Le palais des miroirs
- Ghosh, Amitav
- Besse, Christianne

- http://isbn/200386682

- f:title
- f:year
- f:city
- f:publisher
- f:name
- f:author
- f:original
- f:homepage

- f:nom
- f:auteur
- f:traducteur
3rd: start merging your data (cont)

Same URI!
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, … « donnes-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
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 vocabularies, extra rules, etc., come in
  - vocabularies 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?

- We identified, uniquely, the pieces of data
- We used URI-s to do that
  - we have the usual Web technologies at our disposal
- De-referencing those URIs usually returns useful information
  - remember the way we could explore the wikipedia/dbpedia data
- We abstracted out of the specific data representation; concentrated on the *links* among data points
- We could also deduce, in some cases, new information
So where is the “Linked Data”? So where is the “Semantic Web”?

- The Semantic Web provides technologies to make such integration possible
- Linked Data is a set of general principles whose realizations are (usually) based on Semantic Web technologies
- 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, i.e., 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:

\(<\text{http://…isbn…6682}>, \text{http://…/original}, \text{http://…isbn…409X}\>\)

- **RDF** is a general model for such triples (with machine readable formats like RDF/XML, Turtle, N3, RDFa, Json, …)
RDF triples (cont.)

- Resources can use *any* URI
  - http://www.example.org/file.html#home
  - http://www.example.org/form?a=b&c=d
  - doi:10.3998/3336451.0004.203

- RDF triples form a directed, labeled graph (the best way to think about them!)
  - the abstract concepts are serialized in XML, Turtle, JSON-LD, …
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> .
```
For example, using Python+RDFLib:
- a “Graph” object is created
- the RDF file is parsed and results stored in the Graph
- the Graph 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 Java, PHP, etc.
# create a graph from a file
graph = rdflib.Graph()
graph.parse("filename.rdf", format="rdfxml")

# take subject with a known URI
subject = rdflib.URIRef("URI_of_Subject")

# process all properties and objects for this subject
for (s,p,o) in graph.triples((subject, None, None)):
    do_something(p, o)
But programming is not for everyone

- Not everyone wants to program
- On a higher level of abstraction:
  - RDF graphs are “stored”
    - physical triple stores, databases, etc.
    - simple RDF files loaded by underlying tools
    - users can query and update the graph
    - etc.
  - Data files in different formats and storage are “viewed” as RDF
    - “bridges”, APIs between RDB and RDF, CSV and RDF
Example: You publish the raw data...
... and others can use it
One level higher up: RDF, 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…
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
RDFS defines the meaning of these terms

- (these are all special URI-s, we just use the namespace abbreviation)
Inferred properties

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

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

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 ;
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
### Section 2: Properties in the /terms/ namespace

<table>
<thead>
<tr>
<th>Term Name</th>
<th>abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI:</td>
<td><a href="http://purl.org/dc/terms/abstract">URI</a></td>
</tr>
<tr>
<td>Label:</td>
<td>Abstract</td>
</tr>
<tr>
<td>Definition:</td>
<td>A summary of the resource.</td>
</tr>
<tr>
<td>Type of Term:</td>
<td>Property</td>
</tr>
<tr>
<td>Refines:</td>
<td><a href="http://purl.org/dc/elements/1.1/description">URI</a></td>
</tr>
<tr>
<td>Refines:</td>
<td><a href="http://purl.org/dc/terms/description">URI</a></td>
</tr>
<tr>
<td>Version:</td>
<td><a href="http://dublincore.org/usage/terms/history/#abstract-003">URI</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term Name</th>
<th>accessRights</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI:</td>
<td><a href="http://purl.org/dc/terms/accessRights">URI</a></td>
</tr>
<tr>
<td>Label:</td>
<td>Access Rights</td>
</tr>
<tr>
<td>Definition:</td>
<td>Information about who can access the resource or an indication of its security status.</td>
</tr>
<tr>
<td>Comment:</td>
<td>Access Rights may include information regarding access or restrictions based on privacy, security, or other policies.</td>
</tr>
<tr>
<td>Type of Term:</td>
<td>Property</td>
</tr>
<tr>
<td>Refines:</td>
<td><a href="http://purl.org/dc/elements/1.1/rights">URI</a></td>
</tr>
<tr>
<td>Refines:</td>
<td><a href="http://purl.org/dc/terms/rights">URI</a></td>
</tr>
<tr>
<td>Has Range:</td>
<td><a href="http://purl.org/dc/terms/RightsStatement">URI</a></td>
</tr>
<tr>
<td>Version:</td>
<td><a href="http://dublincore.org/usage/terms/history/#accessRights-002">URI</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term Name</th>
<th>accrualMethod</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI:</td>
<td><a href="http://purl.org/dc/terms/accrualMethod">URI</a></td>
</tr>
<tr>
<td>Label:</td>
<td>Accrual Method</td>
</tr>
</tbody>
</table>
DCTERMS is an RDFS vocabulary

- Each term is an RDFS Property or Class identified by a URI
- Many property terms have range and domain definitions
- Sub-property relationships are also used

```plaintext
  dcterms:accessRights
  dcterms:issued "2003-02-15"^^xsd:date ;
  a rdf:Property ;
  rdfs:range dcterms:RightsStatement ;
  rdfs:subPropertyOf dcterms:rights ;
  ...  
```

from http://dublincore.org/2012/06/14/dcterms.ttl
### Section 2: Properties in the /terms/ namespace

<table>
<thead>
<tr>
<th>Term Name: abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI: <a href="http://purl.org/dc/terms/abstract">http://purl.org/dc/terms/abstract</a></td>
</tr>
<tr>
<td>Label: Abstract</td>
</tr>
<tr>
<td>Definition: A summary of the resource.</td>
</tr>
<tr>
<td>Type of Term: Property</td>
</tr>
<tr>
<td>Refines: <a href="http://purl.org/dc/elements/1.1/description">http://purl.org/dc/elements/1.1/description</a></td>
</tr>
<tr>
<td>Refines: <a href="http://purl.org/dc/terms/description">http://purl.org/dc/terms/description</a></td>
</tr>
<tr>
<td>Version:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term Name: accessRights</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI: <a href="http://purl.org/dc/terms/accessRights">http://purl.org/dc/terms/accessRights</a></td>
</tr>
<tr>
<td>Label: Access Rights</td>
</tr>
<tr>
<td>Definition: Information about who can access the resource or an indication of its security status.</td>
</tr>
<tr>
<td>Comment: Access Rights may include information regarding access or restrictions based on privacy, security, or other policies.</td>
</tr>
<tr>
<td>Type of Term: Property</td>
</tr>
<tr>
<td>Refines: <a href="http://purl.org/dc/elements/1.1/rights">http://purl.org/dc/elements/1.1/rights</a></td>
</tr>
<tr>
<td>Has Range: <a href="http://purl.org/dc/terms/RightsStatement">http://purl.org/dc/terms/RightsStatement</a></td>
</tr>
<tr>
<td>Version: <a href="http://dublincore.org/usage/terms/history/#accessRights-002">http://dublincore.org/usage/terms/history/#accessRights-002</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term Name: accrualMethod</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI: <a href="http://purl.org/dc/terms/acrualMethod">http://purl.org/dc/terms/acrualMethod</a></td>
</tr>
<tr>
<td>Label: Accrual Method</td>
</tr>
</tbody>
</table>
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 (see later…)
Example: Find the right experts at NASA

- Expertise locator for nearly 70,000 NASA civil servants, using RDF integration techniques over 6 or 7 geographically distributed databases, data sources, and web services...

(78) Michael Grove, Clark & Parsia, LLC, and Andrew Schain, NASA, (SWEO Case Study)
“Linking Open Data” Project

- Goal: “expose” open datasets in RDF
- Set RDF links among the data items from different datasets
The LOD “cloud”
Linked Data Principles
a.k.a. is your data 5 Star?

⭐ Available on the Web in some format (i.e., use URI to access the data)

⭐⭐ Available as machine-readable structured data (e.g., excel instead of an image scan)

⭐⭐⭐ As before, but using a non-proprietary format (e.g., CSV instead of excel)

⭐⭐⭐⭐ All the above, plus use open standards (RDF & Co.) to identify things, so that people could point at your stuff

⭐⭐⭐⭐⭐ All the above, plus link your data to other people’s data to provide context
Linked Data ≈ 5 Star Data
(or a collection of 5 Star Data)

Linked Open Data ≈ Linked Data without access restrictions
An important distinction

- Linked Data is based on RDF, but
  - in pure RDF, URI-s are used “simply” as unique identifiers
    - it is o.k. to use non-referenceable URI-s like ISBN URN-s, or DOI URN-s
  - *in Linked Data URI-s should really really be de-referenceable*
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
Extracting structured data from Wikipedia

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

dbpedia:Kolkata
  dbpprop:officialName "Kolkata" ;
  dbpprop:longd "88" ;
  dbpprop:longm "22" ;
  dbpprop:longew "E" ;
  foaf:homage <http://www.kmcgov.in> ;
  dbpedia-owl:populationTotal "4486679" ;
  dbpprop:areaTotalKm "185" ;
...

dbpedia:Amitav_Ghosh
  dbpedia-owl:birthPlace dbpedia:Kolkata ;
  ...

Kolkata

<table>
<thead>
<tr>
<th>Country</th>
<th>India</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>West Bengal</td>
</tr>
<tr>
<td>Division</td>
<td>Presidency</td>
</tr>
<tr>
<td>District</td>
<td>Kolkata[2]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Government</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Mayor–Council</td>
</tr>
<tr>
<td>Body</td>
<td>KMC</td>
</tr>
<tr>
<td>Mayor</td>
<td>Sovan Chatterjee[1]</td>
</tr>
<tr>
<td>Sheriff</td>
<td>Indrajit Ray[2]</td>
</tr>
<tr>
<td>Police commissioner</td>
<td>Surajit Khar Purkayastha[3]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>City</td>
<td>185 km² (71 sq mi)</td>
</tr>
<tr>
<td>Metro</td>
<td>1,866.67 km² (728.45 sq mi)</td>
</tr>
<tr>
<td>Elevation</td>
<td>9 m (30 ft)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Population (2011)[4]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>City</td>
<td>4,466,679</td>
</tr>
<tr>
<td>Rank</td>
<td>7th</td>
</tr>
<tr>
<td>Density</td>
<td>24,000/km² (63,000/ sq m)</td>
</tr>
<tr>
<td>Metro rank</td>
<td>3rd</td>
</tr>
<tr>
<td>Metropolitan</td>
<td>14,817,882 (3rd)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Demonym</th>
<th>Calcuttan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time zone</td>
<td>IST (UTC+05:30)</td>
</tr>
<tr>
<td>ZIP code(s)</td>
<td>7000 xx, 7001 xx</td>
</tr>
<tr>
<td>Area code(s)</td>
<td>+91-33</td>
</tr>
<tr>
<td>Vehicle registration</td>
<td>WS 01–70</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UN/LOCODE</th>
<th>IN CCU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Official language</td>
<td>Bengali and English</td>
</tr>
<tr>
<td>Major Ethnic Settlements</td>
<td>Bengali, Marwari, Bhari and Others</td>
</tr>
<tr>
<td>Website</td>
<td><a href="http://www.kmcgov.in">www.kmcgov.in</a></td>
</tr>
</tbody>
</table>
Links among open datasets

Processors can switch automatically from one to the other…
The importance of Linked Data

- It provides a core set of data that applications can build on
  - stable references for “things”,
    - e.g., http://dbpedia.org/resource/Kolkata/
  - many many relationships that applications may reuse
    - e.g., the BBC application!
  - a “nucleus” for a larger, semantically enabled Web!
Back to the BBC...

Helped reggae reach a mass market. Two of his most popular recordings were "Layla", recorded by Derek and the Dominos, another band he formed, and Robert Johnson's "Crossroads", recorded by Cream. Following the death of his son Conor in 1991, Clapton's grief was expressed in the song "Tears in Heaven", which featured in his Unplugged album.

Read more at 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 and read more about our use of this data.
Back to the BBC...

About: Eric Clapton
An Entity of Type: artist, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

Eric Patrick Clapton, CBE, (born 30 March 1945) is an English guitarist and singer-songwriter. Clapton is the only three-time inductee to the Rock and Roll Hall of Fame, once as a solo artist, and separately as a member of The Yardbirds and Cream. Clapton has been referred to as one of the most influential guitarists of all time. Clapton ranked second in Rolling Stone magazine’s list of the “100 Greatest Guitarists of All Time” and fourth in Gibson's Top 50 Guitars of All Time.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia-owl:abstract</td>
<td>Eric Patrick Clapton is a British guitarist, composer and singer of blues-, rock- and pop music. Eric Clapton is a musician, chanteur and composer of blues and rock and roll. He was born in 1945. In 2003, he was the first British guitarist to be inducted into the Rock and Roll Hall of Fame. Clapton's style has been influential to many other musicians. He has been called one of the greatest guitarists of all time.</td>
</tr>
</tbody>
</table>
Back to the BBC...

```
@prefix ns13: <http://umbel.org/umbel/rc/> .
dbpedia:Eric_Clapton rdf:type ns13:Artist ,
yago:CommandersOfTheOrderOfTheBritishEmpire ,
<http://dbpedia.org/class/yago/2000sSingers> ,
<http://dbpedia.org/class/yago/1980sSingers> ,
foaf:Person ,

@prefix ns14: <http://schema.org/> .
dbpedia:Eric_Clapton rdf:type ns14:MusicGroup ,
<http://dbpedia.org/class/yago/1990sSingers> ,
<http://dbpedia.org/class/yago/PeopleFromGuildford(district)> ,
yago:AlumniOfKingstonUniversity ,
yago:EnglishPeopleOfCanadianDescent ,
ns13:MusicalPerformer ,
owl:Thing ,
ns14:Person ,
yago:Artist109812338 ,
yago:EnglishRockSingers ,
yago:PeopleAssociatedWithTheBeatles ,
yago:EnglishBluEsSingers ,
yago:EnglishBluEsGuitarists .

@prefix ns15: <http://ja.dbpedia.org/resource/> .
dbpedia:Eric_Clapton owl:sameAs ns15:エリック・クラプトン ,
<http://ru.dbpedia.org/resource/\u041a\u043b\u044d\u043e\u0442\u043e\u043d,_\u042d\u0440\u0438\u043a> ,
<http://rdf.freebase.com/ns/m.02qwg> .

@prefix ns16: <http://sw.cyc.com/concept/> .
dbpedia:Eric_Clapton owl:sameAs ns16:Eric_Clapton .
```

(91)
Same dataset, another site
Same dataset, another site
Simplest way to use Linked Data

- Use Web technologies! E.g., use HTTP…
  - extract the data via HTTP GET
  - interpret the (hopefully RDF) content to, e.g., extract new relationships and resources
- Tools are available to do that
  - although, truth must be said, it is still an evolving world
Example: Simple Application Integration via Linked Data

HTTP GET, HEAD
Example: Simple Application Integration via Linked Data

HTTP GET, HEAD

Public Library

Bookshop

Catalogue in the Centre Pompidou

Catalogue in Library of Congress
Currently, Linked Data is dominated by publishing data for read-only usage
  - creating/updating the data is done “out of band”

The future to read and write Linked Data
Example: Simple Application Integration via Linked Data

Application #1

Linked Data on Server #1

Application #2

Linked Data on Server #2

HTTP GET, HEAD

HTTP PUT, DELETE
Example: Simple Application Integration via Linked Data

Private Email Client

Corporate Email Client

HTTP GET, HEAD

HTTP PUT, DELETE

Private Address Data

Corporate Contact Data
Define an HTTP/RESTful based infrastructure to publish, read, write, or modify linked data

- typical usage: data intensive application in a browser, application integration using shared data...

The infrastructure should be easy to implement and install

- provides an “entry point” for Linked Data applications!

The work is still ongoing at W3C…
Technically: the simple approach

- Write RDF/XML, Turtle, etc. “manually”
  - in some cases that is necessary, but it really does not scale...
Access to Relational Databases
Relational Databases and RDF

- Many of the data on the Web is, in fact, in RDB-s
- Proven technology, huge systems, many vendors…
- Data integration on the Web must provide access to RDB-s
  - RDB data should be “exported” into, e.g., RDF
What is “export”?

“Export” does not necessarily mean physical conversion

- for very large databases a “duplication” would not be an option
- systems may provide “bridges” to make RDF queries on the fly
- result of export is a “logical” view of the RDB content
Simple export: RDF Direct Mapping

- A standard RDF “view” of RDB tables
- Valid for all RDB-s, independently of the RDB schema
- Fundamental approach:
  - each row is turned into a series of triples with a common subject (subject URI based on primary key value)
  - column names provide the predicate names
  - cell contents are the objects as literals
  - cross-referenced tables are expressed through URI subjects
What DM processor does

- An DM processor has access to:
  - an RDB schema
  - a database governed by the schema
- … and produces an RDF graph using a standard mapping
Result of the Direct Mapping

- What do we get?
  - we have an RDF “view” of the RDB tables
  - a query against the RDF view may be transformed into an SQL query against the original tables
Direct mapping of the bookshop tables

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Author</th>
<th>Title</th>
<th>Publisher</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>0006511409X</td>
<td>id_xyz</td>
<td>The Glass Palace</td>
<td>id_qpr</td>
<td>2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Homepage</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_xyz</td>
<td>Ghosh, Amitav</td>
<td><a href="http://www.amitavghosh.com">http://www.amitavghosh.com</a></td>
</tr>
</tbody>
</table>
Result of the Direct Mapping

- What do we miss?
  - an RDF view that is close to our application; a more “natural” view of the data
  - i.e., the result of the Direct Mapping must be transformed, somehow, into an RDF that an application may use
Direct graph must be transformed

- Property names should be mapped
- URI-s should be minted
- Literals should be replaced by URI-s
Enters R2RML

- Separate vocabulary for a finer control of the mapping
  - gets to the final RDF graph with one processing step
- Fundamentals are similar:
  - each row is turned into a series of triples with a common subject
  - cross-referenced tables linked via URI-s
Enters R2RML

- There is a finer control over the structure of the result graph
  - the format of the (common) subject URI can be controlled
  - objects might be URI-s generated on the fly via templates from column names
  - datatypes can be assigned to literal objects
  - “virtual” tables can be generated through SQL before processing them through R2RML

- R2RML can generate the final RDF ready to be used by an application
An R2RML processor has access to:
- an RDB schema
- an R2RML instance
- a database governed by the schema

... and produces an RDF graph
CSV Data

- CSV ("Comma Separated Value") may be the single most widespread format of datasets published on the Web
- There is no standard RDF mapping yet, but…
- … the structure is very close to a relational table
- W3C plans to start a group that will (also) standardize that mapping
  - pretty much along the lines of the Direct Mapping
Structured data in HTML: RDFa & microdata
RDF with HTML

- By adding some “meta” information, the same source can be reused
  - typical example: your personal information, like address, should be readable for humans and processable by machines
- Some solutions have emerged:
  - add extra statements in microdata or RDFa that can be converted to RDF
    - microdata can be used for a (useful) subset of RDF
    - RDFa is, essentially, a complete serialization of RDF
Ivan Herman

Who am I?

I graduated as mathematician at the [Eötvös Loránd University of Budapest](https://en.wikipedia.org/wiki/Eötvös_Loránd_University_of_Budapest), Hungary, in 1979. After a brief scholarship at the Université Paris VI I joined the Hungarian research institute in computer science ([SZTAKI](https://en.wikipedia.org/wiki/SZTAKI)) where I worked for 6 years (and turned into a computer scientist...). I left Hungary in 1986 and, after a few years in industry in Munich, Germany, I joined the [Centre Mathematics and Computer Sciences (CWI)](https://en.wikipedia.org/wiki/Centre_Mathematics_and_Computer_Sciences) in Amsterdam where I have a tenure position since 1988. I received a PhD degree in Computer Science in 1990 at the [University of Leiden](https://en.wikipedia.org/wiki/University_of_Leyden), in the Netherlands. I joined the [World Wide Web Consortium (W3C)](https://www.w3.org) Team as Head of [W3C Offices](https://www.w3.org/2001/offices) in January 2001 while maintaining my position at CWI. I served as Head of Offices until June 2006, when I was asked to take the [Semantic Web Activity](https://www.w3.org/2001/sw/) Lead position, which is now my principal work at W3C.

Before joining W3C I worked in quite different areas (distributed and dataflow programming, language design, system programming), but I spend most of my research years in computer graphics and information visualization. I also participated in various graphics related ISO standardization activities and software developments. My "professional" [homepage](https://www.w3.org/People/ivan/internal.html) contains a list of my publications (see also my [Mendeley account](https://mendeley.com/user/ivan-herman)), my public presentations, and details of the various projects I participated in the past. There is also a [dblp entry for my publications](https://dblp.uni-trier.de/pers/h/hermann) generated automatically (although I am not sure it is complete...). (B.t.w., based on my publications, my [Erdős number](https://en.wikipedia.org/wiki/Erd%C5%91s_number) is ≤4...

In my previous life (i.e., before joining W3C...) I was member of the Executive Committee of the [Eurographics Association](https://en.wikipedia.org/wiki/Eurographics_Association) for 15 years, and I was vice-chair of the Association between 2000 and 2002. I was the co-chair of the 9th [World Wide Web Conference](https://www.w3.org/2001/wiki/WWWConferences), in Amsterdam, May 2000; since then, I have also been member of [IW3C2](https://www.iw3c2.org) (International World Wide Web Conference Committee), responsible for the World Wide Web Conference series. Since autumn 2007 I am also member of [SWSA (Semantic Web Science Association)](https://www.swsa.org), the committee responsible for the International Semantic Web Conferences (better known as “ISWC”) series.

Some personal data

- The Hungarian spelling of my [full name is Herman Iván, i.e. my name is Ivan (well, spelled properly: Iván)](https://en.wikipedia.org/wiki/Iván) and my [surname is Herman](https://en.wikipedia.org/wiki/Hermann) (many in the Netherlands and in Germany mix it up, and use “Herman” as my name... this is aggravated by the fact that, uniquely in Europe, the Hungarian custom is to put surname first).
- Nationalities: French and Hungarian
- Gender: male
- Family: I am married and have a son, David.
- Date and city of birth: 24th February, 1955, [Budapest](https://en.wikipedia.org/wiki/Budapest), Hungary
- Email addresses: `ivan` on my own [ivan-herman.net](https://www.ivan-herman.net) domain, `ivan` on the [w3.org](https://www.w3.org) domain, or `ivan.herman` on the [cwi.nl](https://www.cwi.nl) domain
- (Mobile) Phone: +31-641044153
- Skype ID: ivan_herman
- I live in [Amstelveen](https://en.wikipedia.org/wiki/Amstelveen) (see also [geonames](https://www.geonames.org)), the Netherlands (lat: 52.302663, long: 4.87397). This is a suburb of [Amsterdam](https://en.wikipedia.org/wiki/Amsterdam). The closest airport is Amsterdam Schiphol
- I am the administrator of the Semantic Web Activity Blog at W3C which can either be [accessed directly](https://www.w3.org/2001/sw/) or via its
Ivan Herman

Who am I?

I graduated as mathematician at the Bővös Loránd University of Budapest, Hungary, in 1979. After a scholarship at the Université Paris VI I joined the Hungarian research institute in computer science (SZTAKI) and worked for 6 years (and turned into a computer scientist...). I left Hungary in 1986 and, after a few years in Munich, Germany, I received a PhD degree in Computer Science in 1990 at the University of London and joined the World Wide Web Consortium (W3C) Team as Head of W3C Offices in January 2005. I served as Head of Offices until June 2006, when I was asked to take over principal work at W3C.

Before joining W3C I worked in quite different areas of research, including language design, system visualization, information visualization. I also served as the co-chair of the 9th World Wide Web Conference (IW3C2) (International World Wide Web Conferences) in 2007.

In my previous life (i.e., before joining W3C...)

Some personal data

The Hungarian spelling of my full name is Ivan (well, spelled properly: Iván) and my custom is to put surname as my name... this is (many in the Netherlands and in Germany mix it up, and use "Herman" as my name... this is a suburb of

Nationalities: Hungarian, French (born 1973, long: 4.87397l). This is a suburb of
I graduated as mathematician at the World Wide Web Consortium (W3C) and joined the team as Head of the Semantic Web Activity Lead. Before joining W3C I worked in quite different areas (distributed and dataflow programming, language design, system design). In my previous life (i.e., before joining W3C…) I was member of the Executive Committee of the .

Some personal data.

The Hungarian spelling of my full name is Hungarian: Hornan Iván. My name is Ivan (well, spelled properly: Iván) and my surname is Herman (many in the Netherlands and in Germany mix it up, and use “Herman” as my name... this is aggravated by the fact that...).
<http://www.ivan-herman.net/foaf#me>
  schema:alumniOf <http://www.elte.hu> ;
  foaf:schoolHomePage <http://www.elte.hu> ;
  schema:worksFor <http://www.w3.org/W3C#data> ;
...
<http://www.elte.hu>
  dc:title "Eötvös Loránd University of Budapest" .
...
<http://www.w3.org/W3C#data>
  dc:title "World Wide Web Consortium (W3C)"
...
Oscars 2012: The Artist, review

The Artist, an utterly beguiling silent, black-and-white celebration of early Hollywood won Best Picture at the Oscars 2012.
The Artist, an utterly beguiling silent, black-and-white celebration of early Hollywood won Best Picture at the Oscars 2012.
Oscars 2012: The Artist, review

The Artist, an utterly beguiling silent, black-and-white celebration of early Hollywood won Best Picture at the Oscars 2012.
Yielding...

[ rdf:type schema:Review ;
  schema:name "Oscars 2012: The Artist, review" ;
  schema:description "The Artist, an utterly beguiling..." ;
  schema:ratingValue "5" ;
...
]
Schema.org is a cooperation of search engines (Bing, Google, Yahoo!, and Yandex)

It is a large vocabulary that they all understand

The terms are extracted from HTML5+microdata or HTML5+RDFa

- the various partners use it for different purposes
- it can be used by anyone outside of the search world!
## Review

A review of an item — for example, a restaurant, movie, or store.

<table>
<thead>
<tr>
<th>Property</th>
<th>Expected Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Properties from Thing</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>additionalType</td>
<td>URL</td>
<td>An additional type for the item, typically used for adding more specific types from external vocabularies in microdata syntax. This is a relationship between something and a class that the thing is in. In RDFa syntax, it is better to use the native RDFa syntax – the <code>typeof</code> attribute – for multiple types. Schema.org tools may have only weaker understanding of extra types, in particular those defined externally.</td>
</tr>
<tr>
<td>description</td>
<td>Text</td>
<td>A short description of the item.</td>
</tr>
<tr>
<td>image</td>
<td>URL</td>
<td>URL of an image of the item.</td>
</tr>
<tr>
<td>name</td>
<td>Text</td>
<td>The name of the item.</td>
</tr>
<tr>
<td>sameAs</td>
<td>URL</td>
<td>URL of a reference Web page that unambiguously indicates the item's identity. E.g. the URL of the item's Wikipedia page, Freebase page, or official website.</td>
</tr>
<tr>
<td>url</td>
<td>URL</td>
<td>URL of the item.</td>
</tr>
<tr>
<td><strong>Properties from CreativeWork</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>about</td>
<td>Thing</td>
<td>The subject matter of the content.</td>
</tr>
<tr>
<td>accountablePerson</td>
<td>Person</td>
<td>Specifies the Person that is legally accountable for the CreativeWork.</td>
</tr>
<tr>
<td>aggregateRating</td>
<td>AggregateRating</td>
<td>The overall rating, based on a collection of reviews or ratings, of the item.</td>
</tr>
<tr>
<td>alternativeHeadline</td>
<td>Text</td>
<td>A secondary title of the CreativeWork.</td>
</tr>
<tr>
<td>associatedMedia</td>
<td>MediaObject</td>
<td>The media objects that encode this creative work. This property is a synonym for encodings.</td>
</tr>
<tr>
<td>audience</td>
<td>Audience</td>
<td>The intended audience of the item, i.e. the group for whom the item was created.</td>
</tr>
<tr>
<td>audio</td>
<td>AudioObject</td>
<td>An embedded audio object.</td>
</tr>
<tr>
<td>author</td>
<td>Person</td>
<td>The author of this content. Please note that author is special in that HTML 5 provides a special mechanism for indicating authorship via the rel tag. That is equivalent to this and may be used interchangeably.</td>
</tr>
<tr>
<td>award</td>
<td>Text</td>
<td>An award won by this person or for this creative work.</td>
</tr>
<tr>
<td>awards</td>
<td>Text</td>
<td>Awards won by this person or for this creative work. (Legacy spelling; see singular form, award)</td>
</tr>
<tr>
<td>citationRef</td>
<td>CreativeWork</td>
<td>A citation or reference to another creative work, such as another publication, web page, scholarly article, etc.</td>
</tr>
</tbody>
</table>
The Artist showtimes for Amsterdam

Pathé Tuschinski - Reguliersbreestraat 26-34, Amsterdam - Map
11:50 - 14:05 - 19:10

Filmtheater "De Uitkijk" - Prinsengracht 452, Amsterdam - Map
12:15 - 19:00 - 21:15

Filmtheater Rialto - Ceintuurbaan 338, Amsterdam - Map
12:45

+ Show more theaters

The Artist (2011) - IMDb
www.imdb.com/title/tt1655442/

Silent movie star George Valentin bemoans the coming era of talking ... Still of Jean Dujardin and Missi Pyle in The Artist Still of Bérénice Bejo in The Artist Reem ...

→ Full cast and crew - The Artist Trailer (Official ... - Bérénice Bejo - Jean Dujardin

Amsterdam
Change location

The Artist (film) - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/The_Artist_(film)
The Artist is a 2011 French romantic comedy drama in the style of a black-and-white silent film written and directed by Michel Hazanavicius, starring Jean ...

→ Jean Dujardin - Bérénice Bejo - Uggie - Dieggesis

The Artist Trailer 2011 HD - YouTube
www.youtube.com/watch?v=O8K9AZcSQJE
25 Aug 2011 - 3 min - Uploaded by TrailersApplecom
I love how George Clooney, and Brad Pitt, lost the Best actor catagory to this film. It just shows that there is ...

More videos for the artist movie »

Oscars 2012: The Artist, review - Telegraph
www.telegraph.co.uk › Culture › Film › Movie Reviews

★★★★★ Review by Robbie Collin
27 Feb 2012 – The Artist, the final film to be released in 2011 and also the most heart-swellingly joyful one, is a silent movie, screened in black and white and ...

The Artist is the perfect film about Hollywood | Hadley Freeman
HTML+* can also be generated

- CMS systems may generate such data automatically
  - e.g., Drupal 7 generates pages with RDFa included
- There are a number of plugins to blogging systems
  - generate HTML+RDFa or HTML+microdata
How to Publish Data?
Some things to remember if you publish data

- Publish your data first, care about sexy user interfaces later!
  - the “raw data” can become useful on its own right and others may use it
  - you can add your added value later by providing nice user access

- If possible, publish your data in RDF but if you cannot, others may help you in conversions
  - trust the community…

- Add links to other data. “Just” publishing isn’t enough…
Some things to remember if you publish data

- Think about persistence and versioning
  - others may depend on the data you publish...

- Be careful about URIs you choose
  - avoid being very domain dependent; e.g., your institution’s name may change at some point
  - use some consistent URI structure
    - e.g. http://publications.europa.edu/{type}/{subtype}
    - etc.

- Avoid reinventing the wheel when choosing vocabularies
Some things to remember if you publish data

- *Document your data*, i.e., provide metadata
  - there are vocabularies to do this
    - Data Catalog Vocabulary (DCAT)
    - Vocabulary of Interlinked Datasets (VoID)
    - DCTERMS (of course 😊)
    - vocabularies for licensing (Open Data Commons, government licenses)
      - this area is still very much in development…

- A new group on Best Practices will start at W3C soon
Northern Ireland Hospital Waiting Lists

This data has achieved Pilot level on 11 June 2013 which means extra effort went in to support and encourage feedback from people who use this open data.

General Information

This data is described at http://data.gov.uk/dataset/northern_ireland_waiting_lists

The data curator's website is http://www.dhsspsni.gov.uk/

Legal Information

This data was originally created or generated by its curator

This data is available under UK Open Government Licence

This data contains aggregated data

This data is curated by

Department of Health, Social Services and Public Safety

Copyright and database rights are described at http://data.gov.uk/dataset/northern_ireland_waiting_lists

The content is available under UK Open Government Licence
<table>
<thead>
<tr>
<th>Practical Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data appears in this collection <a href="http://data.gov.uk/data/search?q=waiting-list&amp;tags=hospital">http://data.gov.uk/data/search?q=waiting-list&amp;tags=hospital</a></td>
</tr>
<tr>
<td>The lag between creation and publication of this data is minimal</td>
</tr>
<tr>
<td>Data quality issues are documented at <a href="http://www.dhsspsni.gov.uk/index/stats_research/hospital-stats/...">http://www.dhsspsni.gov.uk/index/stats_research/hospital-stats/...</a></td>
</tr>
<tr>
<td>Quality control processes are described at <a href="http://www.dhsspsni.gov.uk/index/stats_research/hospital-stats/...">http://www.dhsspsni.gov.uk/index/stats_research/hospital-stats/...</a></td>
</tr>
<tr>
<td>The data is available for a long time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technical Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Releases follow this consistent URL pattern <a href="http://www.dhsspsni.gov.uk/northern_ireland_waiting_times_ce...">http://www.dhsspsni.gov.uk/northern_ireland_waiting_times_ce...</a></td>
</tr>
<tr>
<td>Releases of this data are listed at <a href="http://www.dhsspsni.gov.uk/index/stats_research/hospital-stats/...">http://www.dhsspsni.gov.uk/index/stats_research/hospital-stats/...</a></td>
</tr>
<tr>
<td>This data is machine-readable</td>
</tr>
<tr>
<td>Statistical data is published in a presentation format</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Social Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send questions about this data to <a href="mailto:statistics@dhsspsni.gov.uk">statistics@dhsspsni.gov.uk</a></td>
</tr>
<tr>
<td>Send questions about privacy to <a href="mailto:statistics@dhsspsni.gov.uk">statistics@dhsspsni.gov.uk</a></td>
</tr>
<tr>
<td>Find out how to suggest improvements to publication at <a href="http://www.dhsspsni.gov.uk/index/stats_research/hospital-stats/...">http://www.dhsspsni.gov.uk/index/stats_research/hospital-stats/...</a></td>
</tr>
</tbody>
</table>
Query and Update RDF: (SPARQL)
The LDP approach is great to build up simple applications

But as data grows, complexity of relationships grows

- more sophisticated query possibilities are necessary
- triples are stored in (possibly very large) databases ("triple stores")

How do I query the RDF data?
Querying RDF graphs

- Remember the Python+RDFLib idiom:

```python
for (s, p, o) in graph.triples((subject, None, None)):
    do_something(p, o)
```
In practice, more complex queries into the RDF data are necessary

- something like: “give me the (a, b) pair of resources, for which there is an x such that (x parent a) and (b brother x) holds” (i.e., return the uncles)
  - these rules may become quite complex

The goal of SPARQL (Query Language for RDF)
for (s, p, o) in graph.triples((subject, None, None)):
    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 the bound resources
Our Python 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

```sparql
SELECT ?isbn ?price ?currency # note: not ?x!
```
Simple SPARQL example

```
SELECT ?isbn ?price ?currency # note: not ?x!
```

Returns: [<...409X>, 33, £]
Simple SPARQL example

```
SELECT ?isbn ?price ?currency # note: not ?x!
```

Returns: [<...409X>,33,:£], [<...409X>,50,:€]
Simple SPARQL example

SELECT ?isbn ?price ?currency # note: not ?x!

Returns: [<…409X>,33,:£], [<…409X>,50,:€], [<…6682>,60,:€]
Simple SPARQL example

SELECT ?isbn ?price ?currency # note: not ?x!

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

```
SELECT ?isbn ?price ?currency # note: not ?x!
    FILTER(?currency == :€) }
```

Returns: [<…409X>,50,:€], [<…6682>,60,:€]
Other SPARQL features

- Limit the number of returned results; remove duplicates, sort them, …
- Optional patterns
- CONSTRUCT new graphs, not only return data
- Use datatypes and/or language tags when matching a pattern
- Aggregation of the results (min, max, average, etc.)
- Path expressions (a bit like regular expressions)
SPARQL usage in practice

- SPARQL is usually used over the network
  - HTTP request is sent to a SPARQL endpoint
  - return is the result of the SELECT, the CONSTRUCT,…

- 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

- Many big datasets usually offer “SPARQL endpoints” using this protocol
SPARQL Update

- SPARQL CONSTRUCT returns a new, modified graph
  - the original data remains unchanged!
- SPARQL Update *modifies the original dataset!*
Bathing Water Data Explorer

Throughout the summer, The Environment Agency of England and Wales collects detailed scientific data on the cleanliness of our bathing waters (mostly beaches). Weekly and annual compliance ratings are given:

- meets the higher standard
- meets the minimum standard
- fails to meet the minimum standard

Use this website to search for a bathing water by name, county or postcode, or by browsing around the map. You can then see the detailed data the agency has collected about that site.
Bathing Water Data Explorer

Bathing water at Felixstowe South

Suffolk, England Local authority: Suffolk Coastal District
Year of designation: 1989

Felixstowe is a resort town but is also home to the country's largest container port on the Orwell Estuary. Felixstowe South is a sand and shingle beach gently sloping to the sea in a sheltered bay. There is sand at low-tide. The beach is backed by a promenade and gardens.

Water sampling point location:
Latitude 51.9643004697953, Longitude 1.34, 623700, 233700
View in: Google Maps, Bing Maps, OpenStreetMap

Nearby bathing waters:
Felixstowe North, Dovercourt, Walton, Frinton

Bathing water quality results
Recent results from water quality assessments under the Bathing Water Directive appear below, or view more details of assessment results

<table>
<thead>
<tr>
<th>Annual Compliance Results</th>
<th>Latest weekly in season</th>
</tr>
</thead>
<tbody>
<tr>
<td>Higher</td>
<td>Higher</td>
</tr>
</tbody>
</table>

Catchment map

Bathing water map

Catchment description
The River Deben is 5km to the north and this drains a large mixed catchment. The Orwell and Stour estuaries are 30km to the south and drains a large catchment containing some industry and the Port of Felixstowe.
But what about RDFS?

- RDFS is enough for many vocabularies
- But not for all!
Some technologies have emerged

- To re-use thesauri, glossaries, etc: SKOS
- To define more complex vocabularies with a strong logical underpinning: OWL
Thesauri, glossaries (SKOS)

Photo credit "scarletgreen", Flickr
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
The term “Fiction”, as defined by the Library of Congress

From Library of Congress Subject Headings

Fiction

URL(s)
- http://id.loc.gov/authorities/subjects/sh85048050
- http://id.loc.gov/authorities/authority/ap85048050
- http://id.loc.gov/authority/sh85048050#concept

Instance Of
- MADS/RDF Topic
- MADS/RDF Authority
- SKOS Concept

Scheme Membership(s)
- Library of Congress Subject Headings

Collection Membership(s)
- LCSH Collection - Authorized Headings
- LCSH Collection - General Collection

Variants
- Fiction--Philosophy
- Metafiction
- Novellas (Short novels)
- Novela
- Stories

Broader Terms
- Literature

Narrower Terms
- Adventure stories
- Allegories

(159)
Thesauri have identical structures...

- The structure of the LOC page is fairly typical
  - label, variants/alternate label, narrower, broader, …
  - there is even an ISO standard for these
- SKOS provides a basic structure to create an RDF representation of these
LOC’s “Fiction” in SKOS/RDF
Usage of the LOC graph

- Fiction
  - skos:prefLabel
- skos:Concept
  - rdf:type
  - skos:prefLabel
- Historical Fiction
- The Glass Palace
  - dc:title
- http://.../isbn/...
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, e.g., tags, annotate other vocabularies, …
Anybody in the World can refer to common concepts
  - they mean the same for everybody

Applications may exploit the relationships among concepts
  - e.g., I can make SPARQL searches for, say, works that are categorized as “Fiction”, albeit indirectly only
Example: FAO Journal portal

- Improved search on journal content based on an agricultural ontology and thesaurus (AGROVOC)

(165) Courtesy of Gauri Salokhe, Margherita Sini, and Johannes Keizer, FAO, (SWEO Case Study)
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
    - i.e., only a few inferences are possible
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 RDF Schemas
  - own namespace, own terms
  - it relies on RDF Schemas
- It is a separate recommendation
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
Connecting to French

- a:author \(\xrightarrow{owl:equivalentProperty}\) f:auteur
- a:Novel \(\xrightarrow{owl:equivalentClass}\) f:Roman
Typical usage of `owl:sameAs`

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

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

- This is a major mechanism of “Linking” in the Linked Open Data project
Property characterization

- In OWL, one can characterize the behavior of properties (symmetric, transitive, functional, inverse functional, reflexive, irreflexive, ...)
- OWL also separates data and object properties
  - “datatype property” means that its range are typed literals
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>.
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;

- Other possibilities: complementOf, 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;

<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 owlClass;


<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
  - typical usage example: definition of all concepts in a health care environment
  - some major issues
    - the way classes (i.e., “concepts”) are defined
    - handling of datatypes

- OWL includes those extra features but... the inference engines become (much) more complex 😞
Example: Organ Failure Risk Detection

System by IO Informatics and UBC:
- data integrated from experimental data, clinical endpoints, public ontologies, LOD, etc.
- statistical analysis is performed on the data
- SPARQL is used to query the results
  - a visual interface is provided
  - for clinicians, a simple web-based alerting of “hits” is provided with statistical scores

Courtesy of Robert Stanley, et al, IO Informatics, USA, and UBC, Canada, (SWEO Case Study)
Example: Organ Failure Risk Detection

(Courtesy of Robert Stanley, et al, IO Informatics, USA, and UBC, Canada, (SWEO Case Study))
What have we achieved?
(putting all together)
Remember the integration example?

Web of Data Applications

- Stand Alone Applications
- Browser Applications

Query and Update

Common "Graph" Format & Common Vocabularies

Inferencing

"Bridges"

Data on the Web
The same with what we learned

Linked Data and Semantic Web Applications

- Stand Alone Applications
- Browser Applications

SPARQL, RDF

RDF Graph with vocabularies in RDFS, SKOS, OWL, ...

Inferencing

RDFa, µData, R2RML, DM ...

Data on the Web

W3C
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
There are also a number “core vocabularies”

- *Dublin Core*
- FOAF: about people and their organizations
- DOAP: on the descriptions of software projects
- SIOC: Semantically-Interlinked Online Communities
- vCard in RDF
- ...
Some books

- T. Heath and C. Bizer: *Linked Data: Evolving the Web into a Global Data Space*, 2011
- M. Watson: *Practical Semantic Web and Linked data Applications*, 2010
- ...

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

- Planet RDF aggregates a number of SW blogs:
  - [http://planetrdf.com/](http://planetrdf.com/)

- Semantic Web Interest Group
  - a forum developers with a publicly archived mailing list, and a constant IRC presence on freenode.net#swig
  - anybody can sign up on the list
    - [http://www.w3.org/2001/sw/interest/](http://www.w3.org/2001/sw/interest/)

- Linked Data mailing list
  - a forum concentrating on linked data with a public archive
  - anybody can sign up on the list
    - [http://lists.w3.org/Archives/Public/public-lod/](http://lists.w3.org/Archives/Public/public-lod/)
Lots of Tools *(not an exhaustive list!)*

- **Some names:**
  - Jena, AllegroGraph, Mulgara, Sesame, flickurl, 4Store, ...
  - 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...
  - ...

- **Categories:**
  - Triple Stores
  - Inference engines
  - Converters
  - Search engines
  - Middleware
  - CMS
  - Semantic Web browsers
  - Development environments
  - Semantic Wikis
  - ...

Conclusions
Linked Data, the Semantic Web, and related technologies are there to integrate data on the Web.

The goal is the creation of a Web of Data.
Lot remain to be done…

- Lots of issues to be solved
- But... W3C needs experts!
  - consider joining W3C, as well as the work done there!
ENJOY THE CONFERENCE!
Thank you for your attention

These slides are also available on the Web:

http://www.w3.org/2013/Talks/0902-Lisbon-IH/