Introduction
Fast recap
Why Linked Open Data (LOD)?
Use cases
Elsevier
Audi
Swiss Life
Personal experience
Benefits
Conclusion
Acronyms

Semantic Web & Linked Open Data use cases

Dr David Croft

Coventry University

david.croft@coventry.ac.uk

2017

Overview

1 Introduction

2 Fast recap
- Why LOD?

3 Use cases
- Elsevier
- Audi
- Swiss Life
- Personal experience

4 Benefits

5 Conclusion

Fast recap

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

–Tim Berners-Lee

Here's one I made earlier

Latest attempt to represent human knowledge in a compatible, linkable, intuitively searchable format.

1895 Repertoire Bibliographique Universel (Universal bibliographic directory).

1927 Statistical Machine.

1939 Memex.

1987 Hypercard.

1989 Information management: A proposal (became the Web).

Repertoire Bibliographique Universel [1]

[1] http://www.udcc.org/index.php/site/page?view=about_history

Web

Doesn't the World Wide Web (Web) do this already?

- Web is clearly the greatest collection of knowledge ever produced.
- Early search engines just did text matching.
  - Improvements made but are still basically relying on advanced text matching.
- Too much data.
  - Because you can find everything, is difficult to find anything (specific).

In order to be really effective at searching need to understand meaning of data being searched.

- Solutions?
  1. Solve general purpose Artificial Intelligence (AI).
  2. Cheat.

Linked Open Data (LOD)

The current attempt to achieve the ideals of the semantic web.

- Massive collection of standards formatted data.
- Importantly the meaning of the data is also included.
  - Don't try and deduce the meaning of the data.
  - Just tell our search tools the meaning of everything.
  - Horray for cheating.

Artificial Intelligence (AI)

If any of this sounds familiar to the expert systems of the 1970s.

- Far more limited claims regarding ability.
- No claims of general purpose AI.
- Old approaches failed due to difficulty of producing sufficient knowledge base[2].
  - LOD offers distributed generation.
  - Each additional resource increases the value of everything else.

[2] And other issues.

eXtensible Markup Language (XML)

User definable markup language.

- Strong unicode support.
  - Non-english character sets.
- Often confused with HTML
  - Data presentation vs. data storage.
  - Both use angle bracket contained tags to denote structure.

```
<quiz>
  <question num="1">
    What is anatidaephobia?
    <answer>Fear of a duck watching you.</answer>
  </question>
</quiz>
```

Resource Description Framework (RDF)

Key features of Resource Description Framework (RDF):

- Representation of information as triples.
  - Object - Attribute - Value.
- Chained triples form a graph.
- Use of Uniform Resource Identifiers (URIs).
- Predicates/properties are also URIs.

Alternative notations

RDF generally made available as XML. Fine for automated tools but unpleasant to read.

Alternative notations do exist.

- Turtle.
- N3.
- N-Triples.
- RDF/JSON.

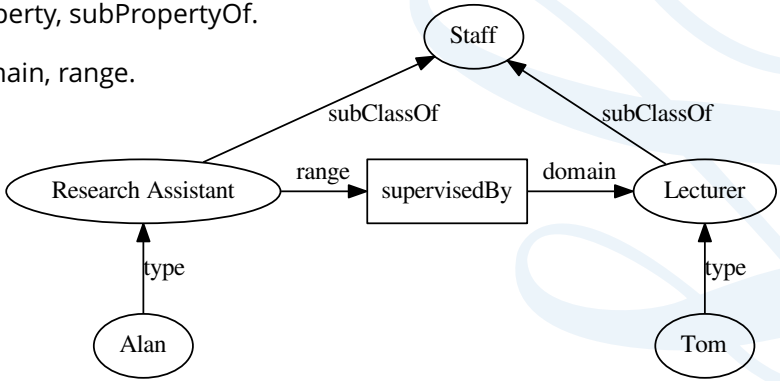
```
<#green-goblin>
    rel:enemyOf <#spiderman>;
    a foaf:Person;
    foaf:name "Green Goblin" .

<#spiderman>
    rel:enemyOf <#green-goblin>;
    a foaf:Person;
    foaf:name "Spiderman".
```

RDF-Schema (RDFS)

Defined vocabulary for RDF.

- Hierarchical organisation.
  - Class, subClassOf, type.
  - Property, subPropertyOf.
  - domain, range.



Web Ontology Language (OWL)

Don't get me started on the acronym.

- Describe term similarity and/or difference.
- Construct new classes.
  - Membership requirements.
  - As intersection, union, complement of existing ones.
- Reason about terms
  - if <Person> <A> and <Person> <B> have same <foaf:email> then <A> and <B> are same individual.

SPARQL Protocol and RDF Query Language (SPARQL)

Structured Query Language (SQL) like syntax for querying RDF sources.

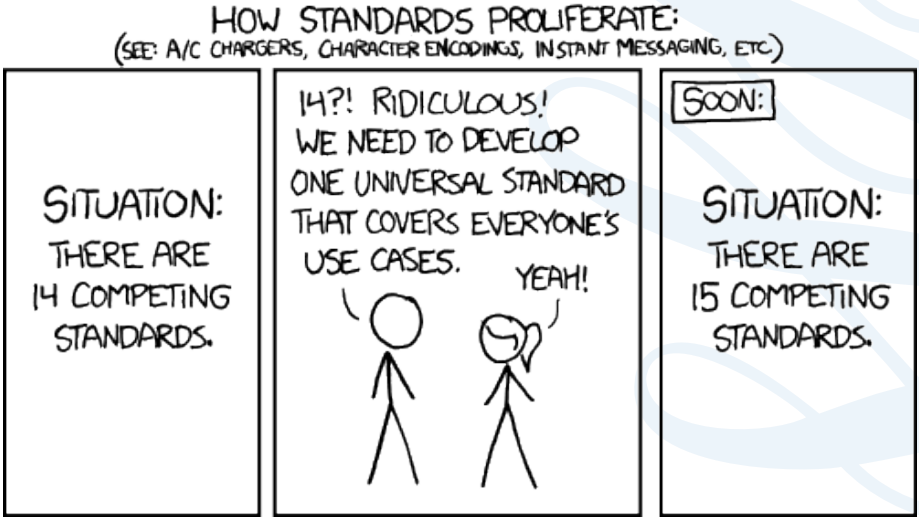
- Querying and transformation only.
  - Not updating, deleting, database admin etc.
- Can get data from RDF graphs.
- Can create new graphs based on existing ones.

Another set of standards, really?

The LOD tech stack succeeds where others have failed because:

- You don't have to migrate your data to a new format.
  - Keeps using whatever weird in-house developed format you want.
- Doesn't try to develop an ontology for everything.
  - Jack of all trades, master of none problem.
- Specify how individual elements of it map to other schemas.
  - Or let others map their schema onto yours.
- Can do conversion on the fly.

Standards



https://xkcd.com/927/

Why LOD?

Why use LOD?

- Do not need to use full LOD tech stack.
- Open data format.
  - Non-propriatory.
  - Easy of future migration.
- Includes the schema.
- Includes the ontology.
- Self documenting.
  - Just follow the URIs.

Real world use cases

Elsevier

- Major academic journal publisher.

Problem

- Wide range of different domains.
  - Subscriptions to individual journals/domains.
- Increase in interdisciplinary research.
  - Information may be spread over multiple subject areas.
- Want everything related to topic e.g. Alzheimers.
  - Information spread - biology, chemistry, medicine, sociology.
  - Differing terminologies (keyword search problematic).

Solution

- Mapping article topics to EMTREE thesaurus.
  - Unique IDentifier (UID) for individual concepts allows cross domain searching despite differing domain terminology.
  - Mapped to existing domain specific ontologies e.g. Medical Subject Headings (MeSH).
  - So keep the existing annotations.
- EMTREE has RDF representation.
  - RDF acts and interoperability format between data sets.

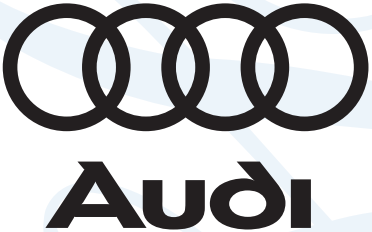
More use cases

Audi

- ≈88 000 employees (2016).
- ≈€60 billion annual revenue.
- ≈2 million cars sold annually.

Problem

- 1 000+ databases.
  - Isolated data.
  - Duplicated data.
- Can't be easily queried.
  - Reliance on manual code generation.
  - Extract, Transform & Load (ETL) processes.



Solution

- You could create a gigantic data warehouse which everything should be stored in.
  - Data warehouse projects fail a lot i.e. the failed care.data NHS project.
- Or ontology based approach.
  - Annotate existing data fields with meaning.
  - Existing applications are unaffected (and unaware).

Imagine two data sources.

Source A

```
<SLR rdf:ID="Olympus-OM-10">
  <viewFinder>twin mirror</viewFinder>
  <optics>
    <Lens>
      <focal-length>75-300mm zoom</focal-length>
      <f-stop>4.0-4.5</f-stop>
    </Lens>
  </optics>
</SLR>
```

Source B

```
<Camera rdf:ID="Olympus-OM-10">
  <viewFinder>twin mirror</viewFinder>
  <optics>
    <Lens>
      <size>30cm zoom</size>
      <aperture>4.5</aperture>
    </Lens>
  </optics>
</SLR>
```

- No way for source A to know that source B's `aperture` is equivalent to its `f-stop`.
- No way for source B to know that an SLR is type of camera.

Application B starts processing some data from source A.

- What is an SLR?
- What is focal-length?
- What is f-stop?
- Could program these rules into each program.
  - 2 sources means explaing it 4 times.
  - 1000 sources means explaining it 1 000 000 times.
- Or use an ontology approach.
  - Encounters unknown term, look it up.
  - Oh that's what it means.

```
<owl:Class rdf:ID="SLR">
  <rdfs:subClassOf rdf:resource="#Camera" />
</owl:Class>

<owl:DatatypeProperty rdf:ID="f-stop">
  <rdfs:domain rdf:resources="#Lens" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="aperture">
  <rdfs:equivalentProperty rdf:resources="#f-stop" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="focal-length">
  <rdfs:domain rdf:resources="#Lens" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="size">
  <rdfs:domain rdf:resources="#focal-length" />
</owl:DatatypeProperty>
```

Unit conversion

How much you can achieve depends on level of detail in ontology.

Start including unit type information in the ontology and we can start doing unit convertion on the fly too.

- Quantities, Units, Dimensions & Types (QUDT), SPARQL Inferencing Notation (SPIN)
- Now not just understanding the equivilence of the fields.
  - Understanding the equivilence of the values.

```
<Lens>
  <size>30</size>
</Lens>
```

```
<owl:DatatypeProperty rdf:ID="size">
  <rdfs:domain rdf:resources="#focal-length" />
  <qudt:unit rdf:resource="http://qudt.org/vocab/unit#Centimeter"/>
</owl:DatatypeProperty>
```

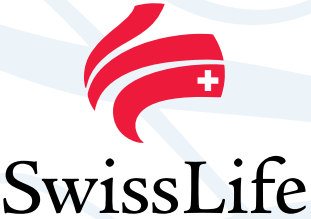
Even more use cases

Switzerland's largest life insurance group.

- ≈11 000 employees.
- ≈$14 billion in premiums.
- Branches in 50 countries.

Problem

- Distributed widely culturally and geographically.
- Creating company wide skills repository is tricky.
  - Who has what skills, qualifications.
  - How are different skills described in different countries?
  - Equivilent qualifications, UK GCSE <> US High School Diploma?



Solution

- Hand built ontology.
- Cover skills from 3 organisational units.
  - IT
  - Private insurance
  - HR
- 700 concepts + 180 educational concepts.
- 130 job function concepts overall.
  - I.e. Administration / Office.
  - Associated job titles; “executive assistant”, “office manager”…
  - Responsibilities; basic clerical, purchasing, facilities management…
  - Skills; word processing, filing, customer relations…

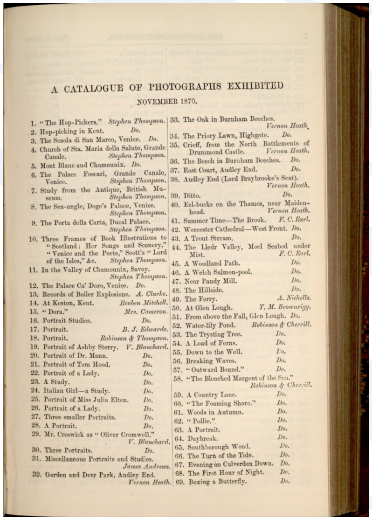
Personal experience

Did my PhD and subsequent research in an area adjacent to Semantic web research.

- FuzzyPhoto project
  `http://fuzzyphoto.edublogs.org/`

Trying to do co-reference identification of photographic artefacts across disparate Gallery, Library, Archive & Museum (GLAM) collections.

- No digitised images so entirely metadata based.
- Success .... kind of.



A CATALOGUE OF PHOTOGRAPHS EXHIBITED

NOVEMBER 1870.

1. "The Hop-Pickers." *Stephen Thompson.*
2. Hop-picking in Kent. *Do.*
3. The Scuola di San Marco, Venice. *Do.*
4. Church of Sta. Maria della Salute, Grande Canale. *Stephen Thompson.*
5. Mont Blanc and Chamounix. *Do.*
6. The Palace Foscari, Grande Canale, Venice. *Stephen Thompson.*
7. Study from the Antique, British Museum. *Stephen Thompson.*
8. The Sea-angle, Doge's Palace, Venice. *Stephen Thompson.*
9. The Porta della Carta, Ducal Palace. *Stephen Thompson.*
10. Three Frames of Book Illustrations to "Scotland: Her Songs and Scenery," "Venice and the Poets," Scott's "Lord of the Isles," &c. *Stephen Thompson.*
11. In the Valley of Chamounix, Savoy. *Stephen Thompson.*
12. The Palace Ca' Doro, Venice. *Do.*
13. Records of Boiler Explosions. *A. Clarke.*
14. At Keston, Kent. *Reuben Mitchell.*
15. "Dora." *Mrs. Cameron.*
16. Portrait Studies. *Do.*
17. Portrait. *B. J. Edwards.*
18. Portrait. *Robinson & Thompson.*
19. Portrait of Ashby Sterry. *V. Blanchard.*
20. Portrait of Dr. Mann. *Do.*
21. Portrait of Tom Hood. *Do.*
22. Portrait of a Lady. *Do.*
23. A Study. *Do.*
24. Italian Girl—a Study. *Do.*
25. Portrait of Miss Julia Elton. *Do.*
26. Portrait of a Lady. *Do.*
27. Three smaller Portraits. *Do.*
28. A Portrait. *Do.*
29. Mr. Creswick as "Oliver Cromwell." *V. Blanchard.*
30. Three Portraits. *Do.*
31. Miscellaneous Portraits and Studies. *James Anderson.*
32. Garden and Deer Park, Audley End. *Vernon Heath.*
33. The Oak in Burnham Beeches. *Vernon Heath.*
34. The Priory Lawn, Highgate. *Do.*
35. Crieff, from the North Battlements of Drummond Castle. *Vernon Heath.*
36. The Beech in Burnham Beeches. *Do.*
37. East Court, Audley End. *Do.*
38. Audley End (Lord Braybrooke's Seat). *Vernon Heath.*
39. Ditto. *Do.*
40. Eel-barks on the Thames, near Maidenhead. *Vernon Heath.*
41. Summer Time—The Brook. *F. C. Burt.*
42. Worcester Cathedral—West Front. *Do.*
43. A Trout Stream. *Do.*
44. The Lledr Valley, Moel Seabod under Mist. *F. C. Burt.*
45. A Woodland Path. *Do.*
46. A Welsh Salmon-pool. *Do.*
47. Near Pandy Mill. *Do.*
48. The Hillside. *Do.*
49. The Ferry. *A. Nicholls.*
50. At Glen Lough. *T. M. Rosenriggs.*
51. From above the Fall, Glen Lough. *Do.*
52. Water-lily Pond. *Robinson & Cherrill.*
53. The Tryeting Tree. *Do.*
54. A Load of Ferns. *Do.*
55. Down to the Well. *Do.*
56. Breaking Waves. *Do.*
57. "Outward Bound." *Do.*
58. "The Bleeched Margaret of the Sea." *Robinson & Cherrill.*
59. A Country Lane. *Do.*
60. "The Foaming Shore." *Do.*
61. Woods in Autumn. *Do.*
62. "Pollie." *Do.*
63. A Portrait. *Do.*
64. Daybreak. *Do.*
65. Southborough Wool. *Do.*
66. The Turn of the Tide. *Do.*
67. Evening on Culverden Down. *Do.*
68. The First Hour of Night. *Do.*
69. Boxing a Butterfly. *Do.*

GLAMs problem

Galleries, Libraries, Archives & Museums (GLAMs) store cultural artefacts. Millions of items.

Problem

- Large digitisation efforts for decades.
  - For conservation.
  - For organisation.
  - For access.
- But no standardisation.
- Every collection has its own schemas and ontologies.
  - If they have any at all.
- Metadata is itself a historical artefact.

GLAMs improvement

Situation is better now.

- Awareness of need for standardisation and cross-collection compatibility.
  - Lightweight Information Describing Objects (LIDO) schema, CIDOC-CRM ontology.
  - ★★★ source but can be mapped over to RDF/linked data etc.
- Awareness of desirability of URIs for referencing/research etc.
- Desire to cross-reference collections.
  - For research, authentication.
- New collections may be created in Linked Data formats.
  - Or large institutions
    British Library - `http://bnb.data.bl.uk/flint-sparql`
    British Museum - `http://collection.britishmuseum.org/`
  - Unfortunately old data is not easily converted.

FuzzyPhoto

FuzzyPhoto, unable to use Semantic web technology.

- GLAM metadata is often too poor quality to be automatically converted to RDF.
- Direct transcription of human readable text with no standardised syntax.
  - I.e. date columns with 21 different date formats in them.
  - No standardised schema.
- Inability to trust the data.
  - Imprecise and/or incorrect.

Semantic web technology would have been brilliant.

- But data couldn't support it.
- Used Natural Language Processing (NLP), Fuzzy logic and clustering instead.

Imprecise/Incorrect

Examples of issues. All examples are real field values.

- Data field says “03/05/10”
  - Most places - That’s the 3rd of May 2010.
  - USA - 5th of March 2010.
  - China, Canada, Korea - 10th of May 2003.
  - But dealing with GLAM data, could be anywhen, 1903, 1803 etc.
- “Tuesday July 7th”.
  - Narrows it to 1 day every 7 years across all of recorded history.
- “18th, 19th or 20th century”.
  - 100 years less precise than if the field was blank.
- “Before the second world war”.

Just the date information, comparable issues exist for all field types used.

Does demonstrate an issue with the Linked Data approach to achieving the Semantic Web.

- Requires high quality data.
  - Structured data
  - If pre-existing data is lacking, can be very expensive to add it in.
- Requires annotated data.
  - If pre-existing annotations are lacking, can be very expensive to add in.

Unstructured problems

Limited benefits when dealing with unstructured data.

- I.e. Web 2.0 material.
  - Facebook posts may be annotated with date, location author.
  - No-one is going to annotate the meaning of the text.
- Data trust.
  - Authorative sources.
  - People lie (spam).

LOD principles

Even if not using the entire LOD tech stack, consider LOD principles and ★ rating.

Principle

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information.
4. Include links to other URIs, so that they can discover more things.

Advantages

1. URIs make it trivial to link objects.
   - DOI, ISBN, UPC etc.
2. Means that URIs become Globally Unique IDentifiers (GUIDs), don't have to worry about domain disambiguation.
3. When someone looks up a URI, provide useful information.
   - Why make it hard to learn out about an item/object/resource?
4. Include links to other URIs, so that they can discover more things.
   - Place things in context.

LOD ★ rating

Rating

★ Data is online.
★★ Data is online in a structured format.
★★★ Data is online in a non-proprietary structured format.
★★★★ Uses URIs so data can be referenced from elsewhere.
★★★★★ Contains links to data sources elsewhere.

Advantages

★ Humans can read it.
★★ Computers can read it, if they have the right software.
★★★ Computers can read it, if you tell them what the data is.
★★★★ Data sources elsewhere can refer to your data.
★★★★★ Your data can be fully understood by computers.

The End
Any questions?

Glossary

AI	Artificial Intelligence
CIDOC	ICOM International Committee for Documentation
CIDOC-CRM	CIDOC-Conceptual Reference Model
DOI	Document On Internet
ETL	Extract, Transform & Load
GLAM	Gallery, Library, Archive & Museum
GUID	Globally Unique IDentifier
HTTP	HyperText Transfer Protocol
ICOM	International Council of Museums
ISBN	International Standard Book Number
JSON	JavaScript Object Notation
LIDO	Lightweight Information Describing Objects
LOD	Linked Open Data
MeSH	Medical Subject Headings
NHS	National Health Service
NLP	Natural Language Processing
OWL	Web Ontology Language
QUDT	Quantities, Units, Dimensions & Types
RDF	Resource Description Framework
RDF/JSON	RDF/JavaScript Object Notation
RDFS	RDF-Schema
SPARQL	SPARQL Protocol and RDF Query Language
SPIN	SPARQL Inferencing Notation
SQL	Structured Query Language
UID	Unique IDentifier
UPC	Universal Product Code
URI	Uniform Resource Identifier
Web	World Wide Web
XML	eXtensible Markup Language