Introduction to linked data

Thomas Meehan, Head of Current Cataloguing, University College London

What is Linked Data?

Linked data is a powerful means of structuring and publishing data on the web. The world wide web we are used to is essentially a linked web of documents; linked data provides a means to create a web of data. Rather than a series of locked down catalogue databases using standards specific to libraries, linked data offers the chance to openly share, link, and enrich this data not only with other libraries but beyond libraries and beyond the purposes to which we currently put it. This article will concentrate on what linked data actually looks like and how to read it. This is important to be able to understand conversations about linked data, which is becoming increasingly prominent in libraries and, in initiatives like Bibframe, is seeking to replace MARC as a standard for encoding and exchanging bibliographic data.

Linked

Cataloguers are used to creating bibliographic data. Most of our traditional bibliographic data is recorded in MARC, of which very little is linked at all. There are 856 fields for URLs but these refer mostly to external documents; they do not allow us to find out any more about the data itself. A publisher's name is recorded in a 260 field subfield $b and there is little more we can find out about that publisher except by taking the text of their name and looking somewhere else for it, e.g. by searching on Google. Where MARC cataloguing does have a strong system of linking is in name and subject authority files. However, these links are established using highly changeable forms of names. These names are frequently internationally established but often also confined to local lists. Other than viewing an authority record, it is also not straightforward for a computer to follow those links to find out more information for the benefit of the user or the cataloguer.

Data

HTML, by contrast, is designed to produce linked and structured documents for humans to read, understand, and follow links from. Below is an excerpt of the HTML source from a WorldCat page for a book, showing a row of a table (tr) with a header cell (th) and a normal cell (td):

```
<tr id="bib-publisher-row">
  <th>Publisher:</th>
  <td id="bib-publisher-cell">Boston, Little, Brown, 1945.</td>
</tr>
```
[1]

This table row contains text, not data. A search engine will find this hard to decode, as it has to know that the table header describes a particular book (rather than, say, a recipe for risotto) and refers to the table cell immediately following it. Even then, it may have problems figuring out exactly what the significance of the English text "Publisher:" is, or what exactly it refers to in the context of the document: are there several books described on the page; is "Boston" and "1945" part of the publisher's name? Another catalogue may instead use "Publisher" or "Publication details" or "Imprint" or a term in French or Chinese.

Much of the web's power and reach depends on it being linked and largely open but we have relied on clever search engines with mysterious algorithms to make sense of those basically textual documents and extract their meaning. Linked data aims in part to solve this by allowing metadata producers to share and publish data as structured data rather than documents, or alongside documents.

---

[1]OCLC. *WorldCat page for a 1945 edition of Bridehead Revisited by Evelyn Waugh.* http://www.worldcat.org/title/brideshead-revisited-the-sacred-and-profane-memories-of-captain-charles-ryder-a-novel/oclc/964336&referer=brief_results

Open

The web relies on openness to an extent which most of us take for granted, but a closed web would be largely unworkable. Although it is possible to create and use large amounts of linked data internally, such as to power a web site or an inventory system, there is a strong assumption of openness with linked data. Indeed, links need to be open to be shared and useful. There is also a strong movement for openness, starting with the open software movement and increasingly in open access within academia, and open data is one aspect of this.

The Open Data Institute defines open data as:

> "information that is available for anyone to use, for any purpose, at no cost.
>
> Open data has to have a licence that says it is open data. Without a licence, the data can't be reused."[2]

The linked data version of the British National Bibliography, for instance, has been

> made available under a Creative Commons CC0 1.0 Universal Public Domain Dedication licence. This means that the British Library Board makes no copyright, related or neighbouring rights claims to the data and does not apply any restrictions on subsequent use and reuse of the data.[3]

This greatly increases the likelihood this data will be re-used. Even a small qualification to an open licence can severely restrict the use of data or content. A non-commercial licence, for example, raises so many questions about the meaning of "commercial" and re-use that Wikipedia will not accept photographs released as Creative Commons Attribution-NonCommercial (CC BY-NC).[4]

Linked Data standards

Linked data is not a technical standard in itself but embraces a number of principles as set out by Tim Berners-Lee in 2006:

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- Include links to other URIs so that they can discover more things.[5]

HTTP URIs cause much confusion but are in effect just URLs used as identifiers. For instance, *http://en.wikipedia.org/wiki/Evelyn_Waugh* is a URL for a web page in English about Evelyn Waugh. It tells a browser where to locate the page and is not an identifier. A HTTP URI is much the same in form but different in intent: *http://id.loc.gov/authorities/names/n79049248* is a URI coined by the Library of Congress for Evelyn Waugh. If you put the URI in a browser you will be directed to a human-readable equivalent in HTML with a different URL; a computer making the same request will receive a copy of the data in RDF at another URL. What is important is that the URI is an identifier, and preferably one where more information can be found, potentially with further links to yet more resources.

---

[2]Open Data Institute. *What Is Open Data?* http://theodi.org/guides/what-open-data

[3]British Library. *British Library Catalogue Datasets in RDF*. http://www.bl.uk/bibliographic/pdfs/british_library_catalogue_dataset_tc.pdf

[4]Wikipedia. *Wikipedia:Non-Free Content.* http://en.wikipedia.org/wiki/Wikipedia:Non-free_content#Background

[5] Berners-Lee, Tim. *Linked Data: Design Issues.* http://www.w3.org/DesignIssues/LinkedData

RDF

RDF is the backbone of linked data and will be the focus of the rest of this article. It stands for Resource Description Framework and is a specification of the World Wide Web Consortium (W3C)[6]. The W3C of course also look after web standards like HTML, XML, and CSS. It is important to note that RDF is by no means a library-specific standard. RDF is not a file format or a specific way of marking up data but is rather a model based on representing data in *triples*. This is best explained by working up an example.

Humans can exchange information with each other using languages like English. For example, we can make an assertion about a book by writing or speaking a simple sentence:

Brideshead Revisited was written by Evelyn Waugh.

We can start by dividing this into Entities and Relationships, similar to the modelling behind FRBR. There is a *subject* entity ("Brideshead Revisited"), an *object* entity ("Evelyn Waugh"), and a relationship between the two ("was written by"), normally referred to as the *predicate* in RDF.

Brideshead Revisited *was written by* **Evelyn Waugh**

This is still English text: the entities are textual, unidentified, and not linked. To identify them unambiguously, we can assign them URIs. As we discovered above, the Library of Congress have created URI identifiers for their name authorities: the URI for the work Brideshead Revisited is *http://id.loc.gov/authorities/names/no97080492*. If we replace the English title with the URI, we get:

`http://id.loc.gov/authorities/names/no97080492` *was written by* **Evelyn Waugh**

Of course, the Library of Congress have also created a URI for Evelyn Waugh, so we can include that too:

`http://id.loc.gov/authorities/names/no97080492` *was written by*
`http://id.loc.gov/authorities/names/n79049248`

Library of Congress does not maintain a URI for the "was written by" relationship (setting aside Bibframe for now), but Dublin Core has the *creator* term, so we can use that. We now have three URIs:

`http://id.loc.gov/authorities/names/no97080492`
http://purl.org/dc/terms/creator
`http://id.loc.gov/authorities/names/n79049248`

If we enclose these URIs in angle brackets and add a full stop on the end…

`<http://id.loc.gov/authorities/names/no97080492>`
<http://purl.org/dc/terms/creator>
`<http://id.loc.gov/authorities/names/n79049248>.`

…we have a valid piece of RDF! Because there are three pieces to it, it is called a triple. Triples are the basis for all RDF: everything in RDF is expressed in triples. The three pieces are called, somewhat similarly to the sentence we started off with, the *Subject*, *Predicate*, and *Object*. A triple is an assertion which stands on its own. It is not part of a record and, unlike a MARC field, does not need one to make sense, although in practice the principle of putting useful information about something at the URI means that data becomes aggregated into documents analogous to a record. Although triples by themselves are inherently simple, in combination they can express very complex ideas.

---

[6] W3C. *Resource Description Framework (RDF)*. http://www.w3.org/RDF/

Turtle

To make RDF easier to read, both for humans and computers, it can be written out in several different ways. The example above is written in a format called *N Triples*, of which more later. The most common format for people to read, and the one this article will prefer, is called **Turtle** [or TTL, for Terse Triple Language]. Our example in Turtle would look like this:

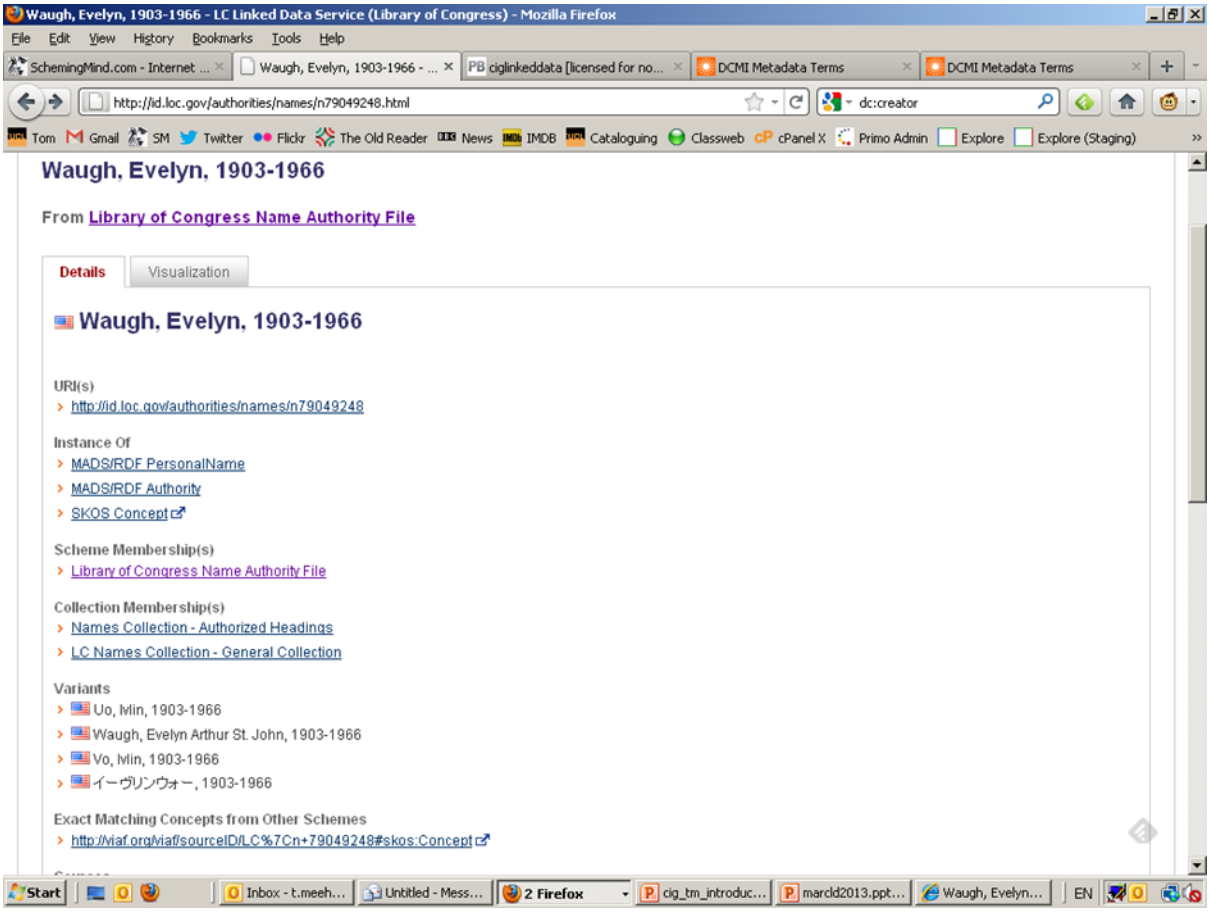
```
@prefix lc_names: <http://id.loc.gov/authorities/names/> .
@prefix dc: <http://purl.org/dc/terms/> .

lc_names:no97080492  dc:creator  lc_names:n79049248 .
```

The first two lines are actually an attempt to make this easier to read! After the word @prefix, a prefix is supplied. This could be anything: it's just to make it easier to read. Lastly there is the base of a URI, followed by the full-stop. Whenever you see that prefix in the document, you can substitute it with the base of the URI. There is still only one triple in the example but now it fits all on one line. The benefits of Turtle become more obvious when we make more assertions about Brideshead Revisited by adding more triples:

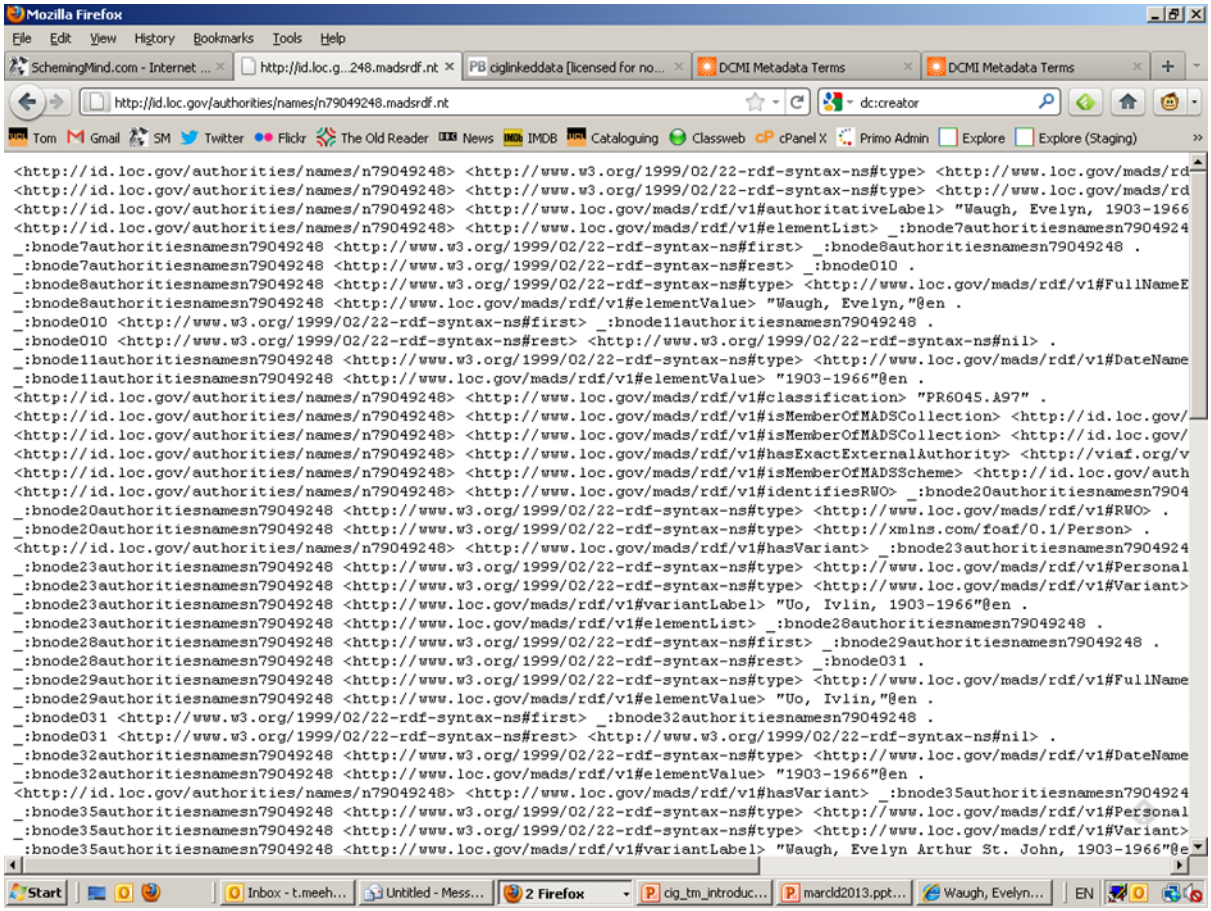
```
@prefix lc_names: <http://id.loc.gov/authorities/names/> .
@prefix lc_languages: <http://id.loc.gov/vocabulary/languages> .
@prefix dc: <http://purl.org/dc/terms/> .

lc_names:no97080492   dc:creator  lc_names:n79049248 ;

                dc:created  "1945" ;
                dc:extent  "1 volume" ;
                dc:language lc_languages:eng ;
                dc:title "Brideshead revisited" ;
                dc:type <http://purl.org/dc/dcmitype/Text> .
```

Because all six triples have the same subject- the URI for the book- it is not repeated. More Dublin Core properties are used: *created* for the date of creation or publication, *extent*, *language*, *title* and *type*. However, for the language we have used the linked data version of the MARC language codes to provide us with a URI. For the type we have used a separate vocabulary maintained by Dublin Core; as it has only been used once in this example, the full URI has been used, although there is nothing stopping us using a @prefix statement for that too. Several data elements here have been expressed as literals. In other words, the values themselves are recorded directly in quotes. Literal values are practically universal in MARC where even the headings are records as literal strings rather than links or ID numbers. As its very name suggests, linked data generally tries to avoid them in preference to URIs identifying and linking to data. However, even a system based on URIs needs to have some literal data in it at some to make the data meaningful.

Much like a HTML document, though, what makes linked data powerful is the ability to follow links. If we put URI for Evelyn Waugh- *http://id.loc.gov/authorities/names/n79049248* – into a browser, we get a page of HTML:



This is because the server knew that we were using a web browser so gave us a page we could read. The URI we typed in has changed to a URL for this page: note that it ends in ".html". A computer programme might make a similar request for the URI and get some raw RDF back. Indeed, further down the page is a list of other formats, one of which is N Triples, which we encountered above. Below is an excerpt:



Note first of all the slightly different URL. Each line of the document is a separate triple and, as such, each triple is relatively easy to read: the first two lines assert what type of thing *http://id.loc.gov/authorities/names/n79049248* is, the third gives an authorised form of his name. However, the very long lines and lack of any helpful formatting make it hard to see what is going on; below is a small excerpt of this data converted to Turtle:

```
@prefix lc_names: <http://id.loc.gov/authorities/names/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix mads: <http://www.loc.gov/mads/rdf/v1#> .
@prefix viaf: <http://viaf.org/viaf/sourceID/> .

lc_names:n79049248  rdf:type mads:PersonalName ;
                    rdf:type mads:Authority ;
                    mads:authoritativeLabel "Waugh, Evelyn, 1903-1966"@en ;
                    mads:hasExactExternalAuthority  viaf:68937142 .
```

Now at least we can see that those types- using the MADS schema- are personal name and authority; to find out more about what those mean, we can follow those URIs. The authoritative label in the third triple is the LC authorised form; the "@en" after it signifies that the literal value is in English. This convention allows for multiple values in different languages. The fourth triple asserts that Evelyn Waugh has an external authority identified by the URI *http://viaf.org/viaf/sourceID/68937142*. We, or a computer programme, can continue to follow such URIs and find more and more information out. For instance, the Virtual International Authority File has links to a variety of national authority schemes as well as Wikipedia. This holds out the possibility of enriching catalogue

displays with biographical information or, going the other way, embedding bibliographic data or catalogue searches into Wikipedia.

Serializations

As discussed above, RDF can be expressed in several different ways. These are called serializations. They are all readable by computers but have differing advantages in various contexts. The most human-readable one is **Turtle**, which can be seen in the last example, repeated below:

```
@prefix lc_names: <http://id.loc.gov/authorities/names/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix mads: <http://www.loc.gov/mads/rdf/v1#> .
@prefix viaf: <http://viaf.org/viaf/sourceID/> .

lc_names:n79049248  rdf:type mads:PersonalName ;
                    rdf:type mads:Authority ;
                    mads:authoritativeLabel "Waugh, Evelyn, 1903-1966"@en ;
                    mads:hasExactExternalAuthority  viaf:68937142 .
```

It is also relatively easy to write out Turtle by hand once you have established some prefixes, although the syntax can become complicated. **Notation3**, or N3, is closely related and looks very similar.

The first serialization used in this article was **N Triples**. The last example re-written as N Triples looks like this:

```
<http://id.loc.gov/authorities/names/n79049248> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.loc.gov/mads/rdf/
v1#PersonalName>. <http://id.loc.gov/authorities/names/n79049248> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.loc.gov/
mads/rdf/v1#Authority>. <http://id.loc.gov/authorities/names/n79049248> <http://www.loc.gov/mads/rdf/v1#authoritativeLabel> "Waugh, Evelyn,
1903-1966"@en. <http://id.loc.gov/authorities/names/n79049248> <http://www.loc.gov/mads/rdf/v1#hasExactExternalAuthority> <http://viaf.org/
viaf/sourceID/68937142>.
```

There is no abbreviation, no prefixes, and no attempt to make it easier to read. It is far easier, however, to see the structure of the underlying triples on each line even if it is very difficult to fit each triple on a separate line!

RDF/XML is the most commonly seen form of RDF. Indeed, there is a common misconception that RDF has to be in XML. This is not the case although RDF/XML was the first serialization approved for use with RDF. Here are the same four triples written as RDF/XML:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:lc_names="http://id.loc.gov/authorities/names/"
 xmlns:mads="http://www.loc.gov/mads/rdf/v1#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax
-ns#" xmlns:viaf="http://viaf.org/viaf/sourceID/">
  <mads:PersonalName rdf:about="http://id.loc.gov/authorities/names/n79049248">
    <rdf:type rdf:resource="http://www.loc.gov/mads/rdf/v1#Authority" />
    <mads:authoritativeLabel xml:lang="en">Waugh, Evelyn, 1903-1966</mads:authoritativeLabel>
    <mads:hasExactExternalAuthority rdf:resource="http://viaf.org/viaf/sourceID/68937142" />
  </mads:PersonalName>
</rdf:RDF>
```

Clearly this is not straightforward to read and the triple structure is largely obscured. However, the ability to process XML is well developed in programming languages and applications so this can be a convenient way of publishing and consuming linked data.

RDF/JSON uses a syntax similar to that used by the programming language Javascript (JSON stands for JavaScript Object Notation) and similar programming languages. This makes it appealing to programmers working in those languages.

```
{
  "http://id.loc.gov/authorities/names/n79049248": {
    "http://www.loc.gov/mads/rdf/v1#hasExactExternalAuthority": [
      {
        "type": "uri",
        "value": "http://viaf.org/viaf/sourceID/68937142"
      }
    ],
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#type": [
      {
        "type": "uri",
        "value": "http://www.loc.gov/mads/rdf/v1#Authority"
      },
      {
        "type": "uri",
        "value": "http://www.loc.gov/mads/rdf/v1#PersonalName"
      }
    ],
    "http://www.loc.gov/mads/rdf/v1#authoritativeLabel": [
      {
        "lang": "en",
        "type": "literal",
        "value": "Waugh, Evelyn, 1903-1966"
      }
    ]
  }
}
```

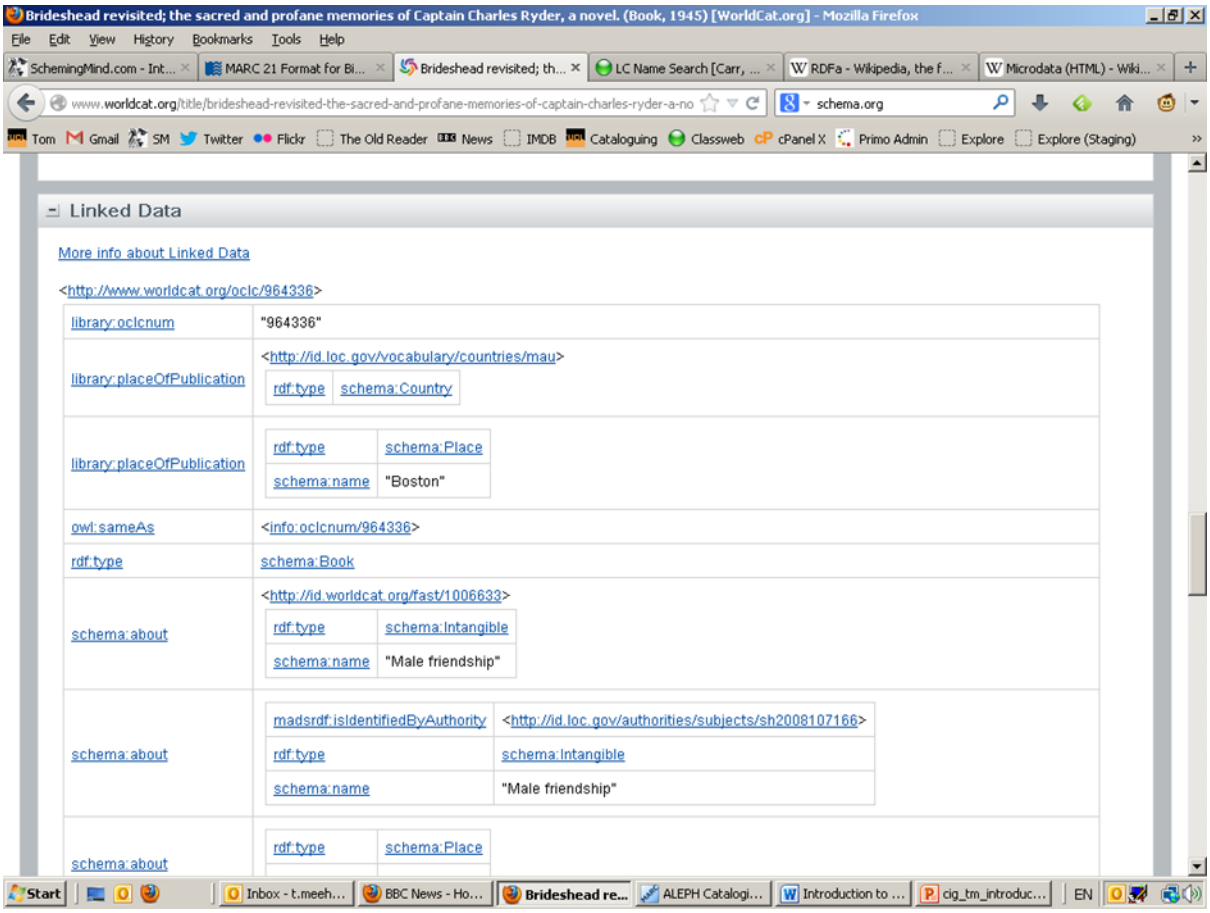
There is a further serialization gaining currency called **JSON-LD** (JSON for Linked Data) which looks similar but is quite different.

RDFa, Microdata, and Schema.org

We started the article with an example of bibliographic data expressed unsatisfactorily as HTML. **RDFa** provides a way of embedding RDF data directly in web pages. Our four-triple example expressed using RDFa in HTML *div* elements might look as follows:

```
<div xmlns="http://www.w3.org/1999/xhtml"
  prefix="
    rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
    mads: http://www.loc.gov/mads/rdf/v1#
    rdfs: http://www.w3.org/2000/01/rdf-schema#"
  >
  <div typeof="mads:PersonalName" about="http://id.loc.gov/authorities/names/n79049248">
    <div rel="rdf:type" resource="http://www.loc.gov/mads/rdf/v1#Authority"></div>
    <div property="mads:authoritativeLabel" xml:lang="en" content="Waugh, Evelyn, 1903-
1966"></div>
    <div rel="mads:hasExactExternalAuthority" resource="http://viaf.org/viaf/
sourceID/68937142"></div>
  </div>
</div>
```

A computer programme reading this web page would be able to extract the underlying triples and therefore the meaning from this. This is highly significant as that computer programme could be a search engine like Google. Instead of having to guess what a web page or an element is about, a search engine would have access to metadata in RDF. Indeed, the Worldcat page from which we took the HTML at the beginning of this article also has linked data embedded in it. At the bottom of the page is a Linked Data section: click on the plus sign (+) and a basic HTML view of it is shown:



This uses the Schema.org vocabulary[7], with some extensions, in RDFa. Schema.org was set up by the search companies Bing, Google, Yahoo!, and Yandex as a standard vocabulary to be used in web pages, in no way limited to libraries. Clearly there is therefore some benefit in including structured linked data into catalogue pages if they are more easily found by search engines, and therefore researchers. Below is a snippet of what a small number of the underlying triples look like in N3:[8]

[7] Schema.org. *What is Schema.org?* https://schema.org/

[8] Extracted using Alex Stoltz's *RDF Translator* at http://rdf-translator.appspot.com. This was also used for many of the translations and extractions of RDF in this article.

```
@prefix library: <http://purl.org/library/> .
@prefix schema: <http://schema.org/> .

<http://www.worldcat.org/oclc/964336> a schema:Book ;
          library:oclcnum "964336"@en ;
          library:placeOfPublication [ a schema:Place ;
                          schema:name "Boston"@en ].
```

The first triple says this is a book in the schema.org vocabulary: using “a” on its own as a predicate is a very common shorthand for the RDFS “type” property, i.e. this particular subject is a “type” book. The second triple uses a library extension to schema.org to specify the OCLC number. The third triple is more complicated and runs across several lines. The square brackets allow several statements to be made about the place of publication without having to give it a URI of its own. Another way of writing this out with the same meaning and more explicit triples would be as follows:

```
<http://www.worldcat.org/oclc/964336> a schema:Book ;
                  library:oclcnum "964336"@en ;
                  library:placeOfPublication _:bnode0
_:bnode001        a schema:Place ;
                  schema:name "Boston"@en .
```

Here we have given the place of publication an arbitrary name called a blank node. We can then make additional statements about this place: the first says that *_:bnode001* is indeed a place; the second that it’s name is “Boston”. Blank nodes are extremely useful for expressing complex concepts in otherwise very simple triples.

Microdata is a similar approach to RDFa for embedding metadata into web pages.

Conclusion: More Than One Way To Do It

Linked data using RDF is a framework for sharing data over the web, not a library-specific vocabulary. It has enormous potential for sharing library data with the world as well as making library data part of a wider web of data. It is powerful, provides precision, but also needs some definition and agreement to work globally. The last example below demonstrates this:

```
@prefix schema: <http://schema.org/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix viaf: <http://viaf.org/viaf/> .
@prefix rdau: <http://rdaregistry.info/Elements/u/>  .
@prefix cam: <http://data.lib.cam.ac.uk/id/entity/> .
@prefix bnb_person: <http://bnb.data.bl.uk/id/person/> .
@prefix foaf: <http://xmlns.com/foaf/spec/#> .

example:book0001 dc:creator cam:cambrdgedb_eeacef63d900c2acffc3daa400f3d4e4 .
example:book0001 dc:creator bnb_person:WaughEvelyn1903-1966 .
example:book0001 schema:creator viaf:68937142 .
example:book0001 rdau:P60447  viaf:68937142 .
example:book0001 dc:creator lc_names:n79049248 .
example:book0001 dc:creator _:bnode001 .
_:bnode001 foaf:name "Waugh, Evelyn, 1903-1966".

example:book0001 example:author example:author0001
```

The same assertion is here made seven times using four different URIs to express the creation relationship

- dc:creator
- schema:creator
- rdau:P60447 (the RDA URI for the creator relationship)
- example:author (one I made up)

and URIs for Evelyn Waugh from five different sources:

- Cambridge
- BNB
- VIAF
- LC
- example (one I made up)

as well as text via a blank node. To be usable the library community will have to choose which ones to re-use and share with others engaged in similar work or whether to make one up a whole new scheme itself. There are clear examples, such as the BNB[9] and Bibframe[10], of both approaches. There are great possibilities ahead.

References

Berners-Lee, Tim. *Linked Data: Design Issues.* http://www.w3.org/DesignIssues/LinkedData

British Library. *British Library Catalogue Datasets in RDF*. http://www.bl.uk/bibliographic/pdfs/british_library_catalogue_dataset_tc.pdf

British Library. *British Library Data Model: Books*. http://www.bl.uk/bibliographic/datafree.html#lod

Chambers, Sally (ed.). *Catalogue 2.0: the Future of the Catalogue*. London: Facet, 2013.

Library of Congress. *BIBFRAME Vocabulary: Category View*. http://bibframe.org/vocab-category/

OCLC. *WorldCat page for a 1945 edition of Bridehead Revisited by Evelyn Waugh.* http://www.worldcat.org/title/brideshead-revisited-the-sacred-and-profane-memories-of-captain-charles-ryder-a-novel/oclc/964336&referer=brief_results

Open Data Institute. *What Is Open Data?* http://theodi.org/guides/what-open-data

Schema.org. *What is Schema.org?* https://schema.org/

Stoltz, Alex. *RDF Translator* at http://rdf-translator.appspot.com.

Wikipedia. *Wikipedia:Non-Free Content.* http://en.wikipedia.org/wiki/Wikipedia:Non-free_content#Background

W3C. *Resource Description Framework (RDF)*. http://www.w3.org/RDF/

9 British Library. *British Library Data Model: Books*. http://www.bl.uk/bibliographic/datafree.html#lod

10 Library of Congress. *BIBFRAME Vocabulary: Category View*. http://bibframe.org/vocab-category/