University of British Columbia
School of Library, Archival and Information Studies
Master of Library and Information Studies

LIBR 594 - Assignment 1

Linked Data in Libraries and Archives

Carolina Román Amigo
Supervisor: Richard Arias Hernandez

October 2017

Table of Contents

Main Linked Data Concepts

Linked Data

Linked data (LD) is the term used to refer to the set of technologies and best practices aimed to prepare and publish data in a way it can be automatically interlinked and shared on the web (Hooland & Verborgh, 2014). By using unique resource identifiers and a data structure based on triples, linked data provides meaningful links among related objects of different provenances, offering more information to the user and improving the discoverability of resources. LD makes use of common vocabularies to ensure understanding across a community. It is also machine-readable, enabling automated agents to interpret data semantically in a similar way a human would do. For that reason, Linked Data, and specifically, Linked Open Data (LD made freely available on the web), can be seen as a building block or a practical implementation of a primitive version of the Semantic Web (Miller, 2011; Southwick, 2015).

Linked Data Principles

The Linked Data principles, or building blocks, popularized by Berners-Lee are:

- *"Use URIs as names for things."*: URIs are uniform resource identifiers, allowing resources to be identified in a unique way anywhere in the universe (Hooland & Verborgh, 2014). Each element in a rdf triple should have a unique identifier: subject, predicate and object.
- *"Use HTTP URIs so that people can look up those names."*: Adding an HTTP method to an URI allows users to access those names to get more information.
- *"When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)."*: Information provided should be relevant about the thing identified, it should be data that someone would like to know about the resource.
- *"Include links to other URIs so that they can discover more things."*: Provide the relationships and other things that the URI is related to.

URL versus URI

In summary, URIs are identifiers, while URLs are addresses. According to Hooland & Verborgh (2014):

> *"A URI, Uniform Resource Identifier, is a generalization of the concept that permits resources anywhere in the universe to be given a unique identification."*
> *"A URL is a uniform resource locator, which, as the name says, enables to locate resources in a unique way."*

Every URL is also an URI, but not every URI is an URL. An URL is an URI added of a method (such as HTTP) that provides access to a resource over a network. Technical standards such as W3C do not endorse the subdivision of URI in URLs, using rather a nomenclature such as HTTP URIs to define URIs pointing to a network location. (Uniform Resource Identifier, 2017) However, although informal, URLs are recognized by the community as a useful concept and, according to Miessler (2015), it is best to use URL when referring to a URI containing both the resource name and the method to access it, while URI is best used when referring directly to a resource name.

Semantic Web

The Semantic Web is the next level of evolution of the internet as we use it today, where machines will be able to understand the semantic meaning of information provided by humans. It is a *"framework for creating, managing, publishing and searching semantically rich information about web resources"* (Alistair et al., 2005). This, according to Berners-Lee, Hendler and Lassila (2001), will *"enable intelligent agents to autonomously perform tasks for us"* (Berners-Lee et al., 2001). It is important to note that the terms "understand" and "intelligent" do not mean here the same for machines and humans. Machines are able to behave "intelligently" only in an operational sense, meaning that the are able to use the data provided according to predefined rules and to infer relationships using logic. In the semantic web, the data is annotated and ontologies, relationships and vocabularies (shared repositories of meaning) are provided so the web, that is constituted today mostly of human-readable information, becomes accessible to software agents.

Regarding information retrieval for humans, the Semantic Web overcomes several of the limitations of keyword based search engines (Alistair et al., 2005). It increases precision when searching for terms with multiple meanings, since the terms have extra information associated with them that allows the search to specify which meaning is the one he is looking for. It provides a better recall as synonyms are taken in account, and, in a similar way, enables search for terms across languages. Limitations regarding retrieval of images, audio and video remain though, since they only become searchable when metadata is added to them.

Metadata Schemas and Metadata Application Profiles (MAP)

In order to be machine-interoperable, metadata has to be structured and atomized. A metadata schema ensures the common interpretation of each metadata element, subelement and attributes, as well as its requirements, content guidelines, controlled vocabularies adopted, etc. (Hooland & Verborgh, 2014). When documented, a metadata schema becomes a metadata application profile (Miller, 2011). The term metadata schema can refer both to formally standardized element sets such as the Dublin Core, VRA 3.0 Core Categories, or DPLA MAP, or to locally established element sets developed to fulfill specific needs.

Some initiatives such the Europeana adopt the term “data model” for designating their metadata application profile documentation. Although technically correct, we find the term too generic to this application as a data model is anything that provides guidelines to structure data. Moreover, using data model to designate a metadata application profile can cause confusion since on the literature, the term is also used to designate data structures such as tabular data, relational data model, meta-markup and RDF.

Namespaces

A namespace is a component of a metadata application profile. According to Hay (2006), *“an XML namespace is the URI that describes an ontology from which terms are taken.”*(Hay, 2006) It allows consistent reuse of elements of metadata description already developed by someone else, ensuring a common understanding of how the data should be interpreted (Hooland & Verborgh, 2014). A prefix is usually added to each element indicating the namespace it comes from, and the full namespaces URIs are indicated at the beginning of the metadata schema file. The example below is an excerpt of the Portland Common Data Model RDF file available on Github (https://github.com/duraspace/pcdm/blob/master/models.rdf).

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="rdfs2html.xsl"?>
<rdf:RDF
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:ldp="http://www.w3.org/ns/ldp#"
    xmlns:ore="http://www.openarchives.org/ore/terms/"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:pcdm="http://pcdm.org/models#">

    <rdf:Description rdf:about="http://pcdm.org/models#">
    <dcterms:title xml:lang="en">Portland Common Data Model</dcterms:title>
    <dcterms:publisher rdf:resource="http://www.duraspace.org/"/>
    <rdfs:seeAlso rdf:resource="https://github.com/duraspace/pcdm/wiki"/>
     <rdfs:comment xml:lang="en">Ontology for the Portland Common Data Model,
intended to underlie a wide array of repository and DAMS
applications.</rdfs:comment>
    <owl:versionInfo>2016/04/18</owl:versionInfo>
    <owl:priorVersion rdf:resource="http://pcdm.org/2015/09/28/models"/>
    </rdf:Description>
```

Controlled vocabularies

According to Hooland & Verborgh (2014), a controlled vocabulary *"represents a restricted subset of language which has been explicitly created to avoid the problems which arise with the use of natural language during the indexing and retrieval of information"* (Hooland & Verborgh, 2014). That means that standardized words are used to represent concepts, establishing preferred terms to promote consistency. (Harpring & Baca, 2010) Controlled vocabularies are a type of Knowledge Organization Systems (KOS). There are mainly three types of controlled vocabularies as described below.

- **Classification schemes:** offer a way to group physically documents of similar content, using classes arranged systematically (Broughton, 2004).
  Example: Dewey Decimal Classification (DDC)

- **Subject headings:** describe the subject of specific resources in a succinct way, using one or few words (Hooland & Verborgh, 2014).
  Example: Library of Congress Subject Headings (LCSH)

- **Thesauri:** represent an application domain in a logical way, building a structure of preferred and non preferred terms, related terms, broader and narrower terms (Hooland & Verborgh, 2014).
  Example: Arts and Architecture Thesaurus (AAT)

Controlled vocabularies increase the chances of finding desired content even with imprecise keywords (related terms). They also increase recall (proportion of the documents relevant to the search that were successfully retrieved) and precision (proportion of retrieved documents relevant to the search). Better recall is possible because thesauri provide synonymy control, meaning that it takes in account different words that may represent the same or similar concepts. Greater precision is possible because of polysemy control, which means that when the same term is used to represent different concepts the thesauri allow for disambiguation because of the hierarchical structure it provides. For example, apple the fruit would be in a different location (under a different broader term) than Apple the company.

However, controlled vocabularies are expensive to create and difficult to maintain, as it takes time and resources to keep them up to date. They are subjective and thus prone to express a specific world view, usually biased. Also, they can be difficult to users to understand and use. They definitely have value in the linked data context, but each application should be evaluated in the light of pros and cons they offer.

SKOS

Simple Knowledge Organization System (SKOS) is a simplified language to represent controlled vocabularies on the web. It is designed to be easier to use than more complex language ontologies such as OWL, but still powerful enough to support semantically enhanced search

(machine-understandable). It is meant to describe content of books and other resources, not to formally describe aspects of the world by axioms and facts as ontologies do (Alistair et al., 2005). The figure 1 below is an example of how SKOS may be used to represent a thesaurus entry. Note the use of a prefix (skos), as in namespaces, and the label of the elements (predicates of the triples) representing classical thesauri terms such as related, broader and narrower.

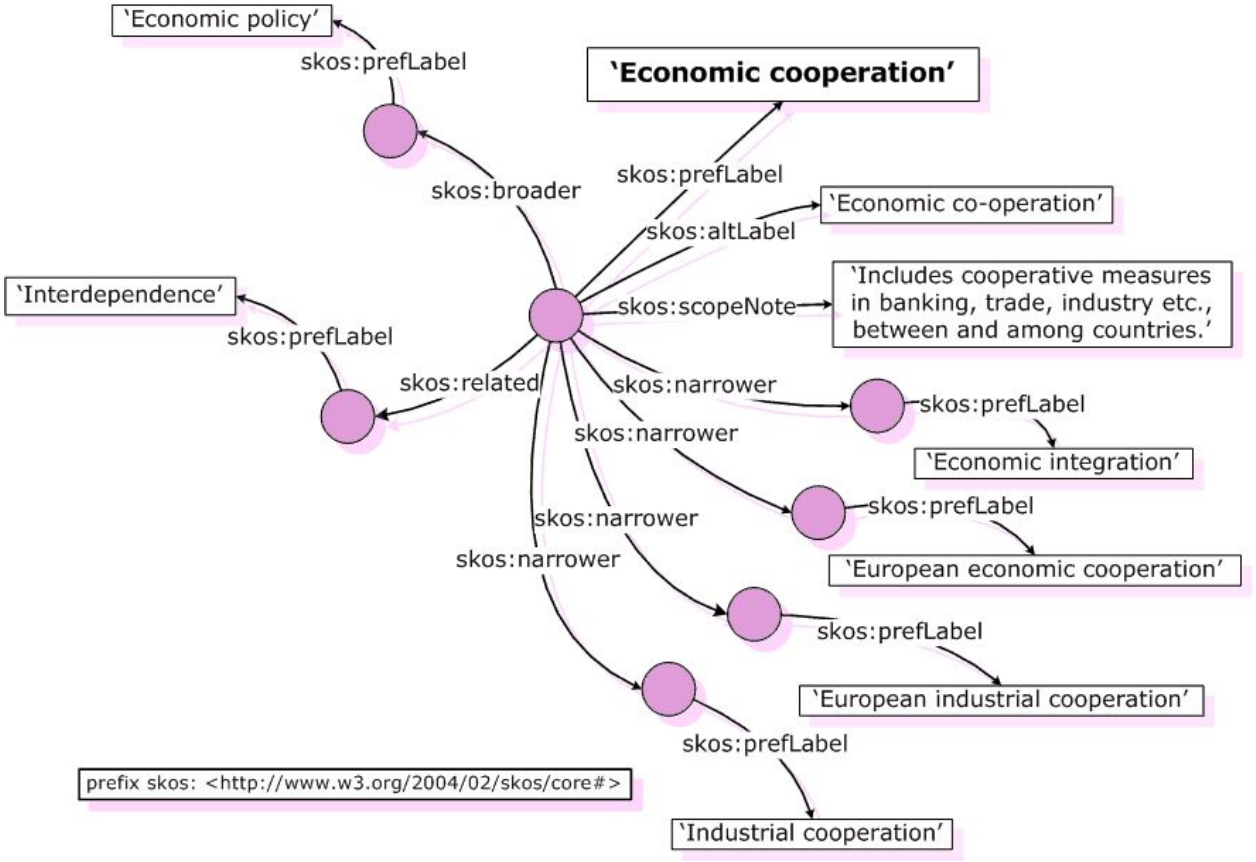

Figure 1 - Example of a thesaurus entry represented in SKOS (source: https://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/)

Ontologies

According to Harpring & Baca (2010), *"an ontology is a formal, machine-readable specification of a conceptual model in which concepts, properties, relationships, functions, constraints, and axioms are all explicitly defined"* (Harpring & Baca, 2010). Ontologies express knowledge about a domain and the relation between its concepts based upon an open world assumption, meaning that inference is allowed based on the information explicitly asserted. Differently from the closed world assumption, where only that of is asserted is considered known, ontologies are able to complete incomplete information entered by applying its rules logically

(Hay, 2006).

Ontologies should not be confused with controlled vocabularies. Controlled vocabularies are used by ontologies to express the vocabulary of a given domain, which is by its turn used according to the grammar defined by the ontology. While controlled vocabularies aim to provide means to cataloging and retrieval, ontologies aim to represent knowledge in a machine-readable form. Concepts are organized in classes, individuals, attributes, relations and events (Harpring & Baca, 2010).

Representational State Transfer (REST) Application Programming Interfaces (APIs)

APIs are a set of defined methods that make data accessible to machines. They have their own vocabulary and syntax, defining property names and labels and how the information is arranged (University of British Columbia, n.d.). APIs receive programming instructions and provide data answers in XML or JSON. HTTP APIs, that preceded REST APIs, didn't offer a way to integrate human access interface and machine access interface, keeping them both distinct. REST, by other hand, provide access for human and machine consumers in the same way, avoiding duplications and minimizing maintenance. This is achieved by uniform interface constraints, consisting on:

- URLs that refers to an object (piece of content) instead of a representation.
- Resource manipulation though representations, returning the representation most appropriate for each request. For example an HTML page for a person and a JSON for a Javascript application.
- Self-descriptive messages which contain all the information necessary to understand and process it. For example, the second page of search results doesn't require the first page in order to be accessed.
- Hypermedia as the engine of application state, meaning that links are provided instead of identificators, dispensing the need for documentation to understand the information (Hooland & Verborgh, 2014).

Main Data Models (Data Structures)

There are four types of data structures: tabular data, relational model, meta-markup languages and RDF triples. Figure 2 synthesizes the differences among these four types.

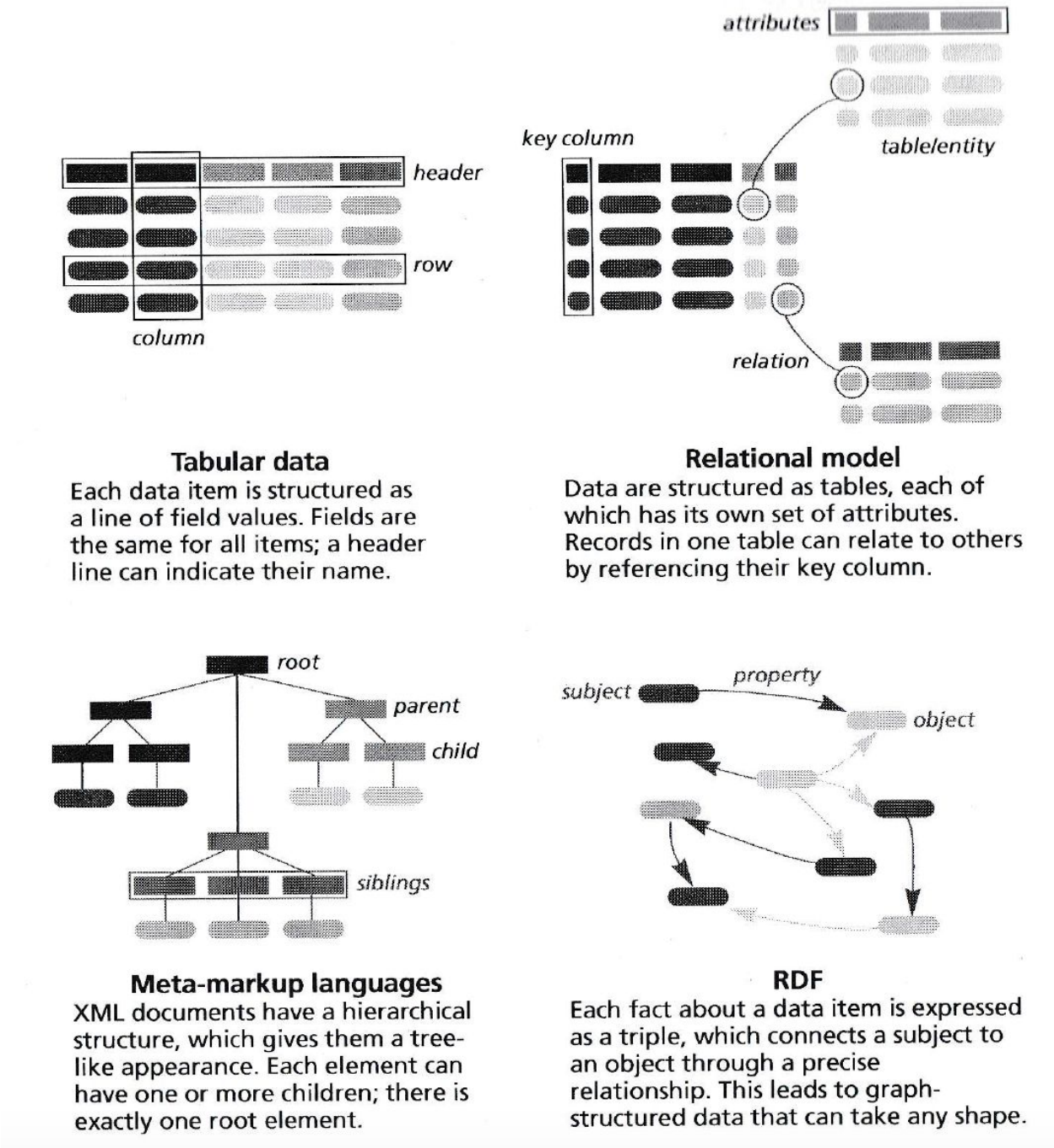

Figure 2 - Schematic comparison of the four major data models (Hooland & Verborgh, 2014).

Tabular Data

Data is organized in columns and rows, and the data in the intersections (cells), has its meaning defined by the columns and row it belongs to. Each item (row) has the same fields

(columns), and a header line can indicate their name. Tabular data is useful to import and export data with a simple structure. It is intuitive to use, portable and independent on the technology used. However, search and retrieval is inefficient and data has to be repeated in many instances, increasing the risk of inconsistencies (Hooland & Verborgh, 2014).

Relational Model

More than one table is used to structure data, each with its own set of fields, and tables are interlinked by using key columns. The relational model minimizes redundancies and inconsistencies of the tabular model. It is used to normalize and manage complex data. It allows better search and retrieval functions, through queries, but is schema-dependent (Hooland & Verborgh, 2014).

Meta-markup languages

Meta-markup languages structure data in a hierarchical way, starting with a single root that breaks down in children elements. It is employed to import and export complex data. It is machine-readable, and also human readable with some training. It is independent of any platform but can be hard to implement for complex data. Its main disadvantage is its verbosity (Hooland & Verborgh, 2014).

RDF (Triples)

Triples, or RDF (resource description framework), structure data in statements consisting of subject, predicate and object. Each line connects a subject to an object through a predicate, expressing a precise relationship. There are no constraints to what can be connected to what and the structure is easily extended by the addition of more triples. Triples are schema-neutral, and the triple is complete semantically (no need for additional documentation). Triples can be expressed in a graph format, and this data model allow logical inference and the linking of data. However, normalization is lost when using a triple data structure, and the software market is still immature to work with data in this format (Hooland & Verborgh, 2014).

SPARQL

SPARQL Protocol and RDF Query Language is a query language to query data structured in triples (RDF) in any serialization format. Queries in SPARQL are based on graph patterns and follow the subject-predicate-object triples structure (Hooland & Verborgh, 2014; Southwick, 2015).

BIBFRAME

BIBFRAME (Bibliographic Framework) is a data model for bibliographic description expressed in RDF vocabulary with classes and properties. It is compatible with linked data principles and it is meant to replace MARC standards. BIBFRAME is based on Functional Requirements for Bibliographic Records (FRBR) model of work, expression, manifestation and item, consisting, however, of three core categories (work, instance, item), with additional key concepts related to the core classes (agent, subject, event) (Figure 3). Properties in BIBFRAME describe characteristics of the resource described as well the relationship among resources (ex: instance of, translation of). (Library of Congress, 2016)

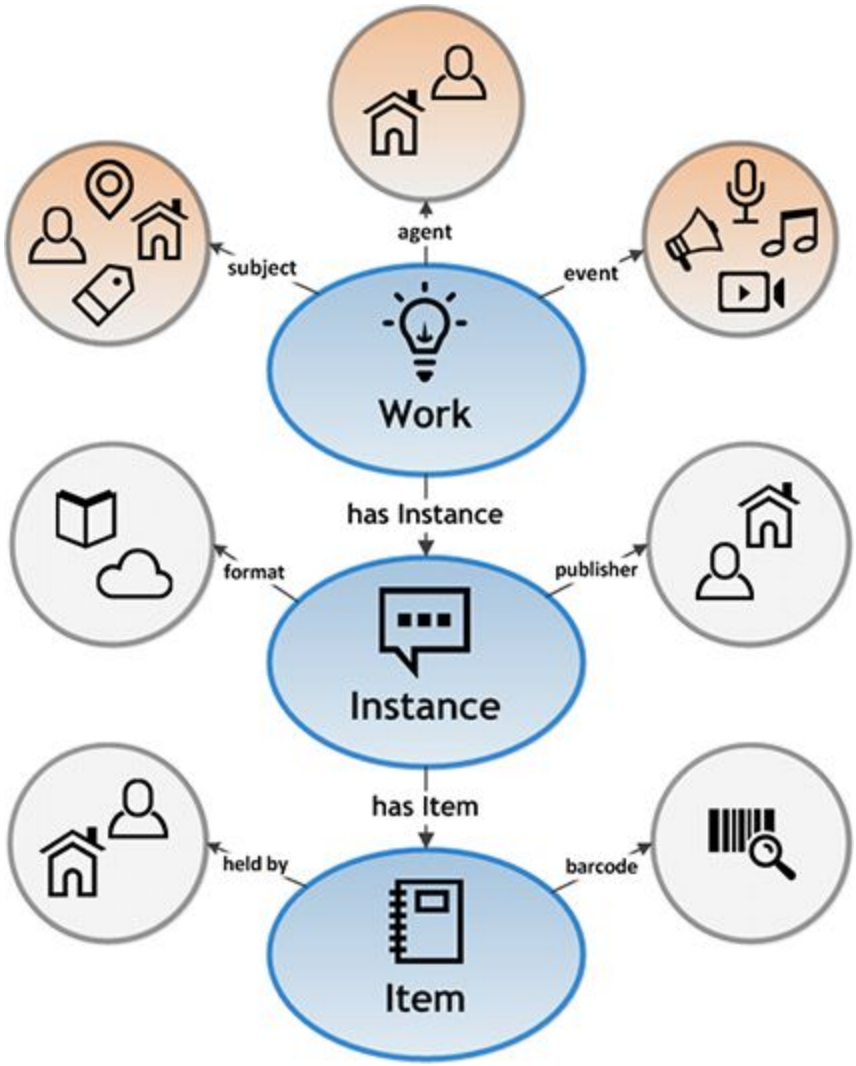

Figure 3 - Illustration of BIBFRAME 2.0 model, with three core levels of abstraction (in blue)—Work, Instance, Item—and three related classes (in orange)—Agent, Subject, Event. (Source: https://en.wikipedia.org/wiki/BIBFRAME)

Serialization Formats for Linked Data

Data structures have to be converted in a stream of bits in order to manipulated by software or shared over a network. Serialization is the process of translating the data structure into a format (Hooland & Verborgh, 2014). The most relevant serialization formats to linked data purposes are briefly described below.

XML

In XML the content is annotated with tags that describe their meaning. Tags are similar to the ones used in other markup languages such as HTML. XML enables data portability, however it lacks semantics (needs an schema in order to be interpreted) and is a verbose serialization format (Hay, 2006).

XSD XML Schema

XML Schema Definition (XSD) provides the semantics that are lacking in a XML format. It lists element and attributes names, relationships, data structure, and data types (Legg, 2007).

JSON and JSON-LD

JSON-LD, or JavaScript Object Notation for Linked Data, was developed based on JSON, enhancing it by providing additional mappings to an RDF model. It is aimed to encode linked data. The additional mappings provide context by linking object properties to concepts in an ontology (JSON-LD, n.d.).

RDF/XML

Resource Description Framework is the main serialization format for linked data. As in XML files, content is annotated with tags for describing semantics. Main tags are rdf:subject, rdf:predicate and rdf:object, forming a triple, with values being expressed as URIs (Hay, 2006).

RDF Schema

Resource Description Framework Schema (RDFS) extends RDF by adding tags to define domain, range, classes and subclasses. According to Ray (2006), *"in RDFS, attributes and relationships are properties that are defined before assigning them to classes. Note that all relationships and attributes are considered optional many-to-many. There are no cardinality constraints in RDF"* (Hay, 2006).

Turtle and N-Triples

Turtle (Terse RDF Triple Language) is a serialization format for RDF (triple) data structure in a less verbose way than RDF/XML, what makes it more compact and easier to read. It *"provides ways to abbreviate such information, for example by factoring out common portions of URIs"* (Turtle syntax, n.d.).
N-triples is a subset of Turtle, line-based. Each line is a triple statement, composed by subject, predicate and object separated by a white space, and terminated with a full stop. Predicates have to be always expressed by URI, while subjects may be a URI or a blank nde, and objects may be a URI, blank node or a literal (string) (N-Triples, n.d.).

OWL

OWL (Web Ontology Language) allows the precise definition of concepts of an ontology (Hooland & Verborgh, 2014), extending RDF by allowing the definition of relationships between classes (union, intersection, complement, etc.), class cardinality, equality for both classes and individuals, properties characteristics (symmetry, transitivity, functionality, etc.), and restrictions on property behaviour by class (e.g. assign class UBC alumni to every record that has "UBC" as institution issuing degree) (Legg, 2007). OWL os based on open world assertion, meaning that *"anything can be true unless asserted otherwise"* (Hay, 2006).

Linked Data Process

Linked data (and linked open data) projects should follow the process depicted in figure 4. The phases are succinctly described below, with exception of the implementation phase, which is more detailed as it has more specificities when compared to standard metadata projects. The information in this section was gathered from Hooland and Verborgh (2014) and Southwick (2015).

Figure 4 - Linked Data process diagram, based on Southwick (2015) (Southwick, 2015).

Planning

- Literature review
- Benchmarking

- Stakeholders
- Proof of concept and preliminary testing
- Securing resources and funds
- Ensure top-level commitment to the project

Designing

- selecting technologies
- defining a data model (aka Metadata Application Profile MAP, Namespaces)
- mapping
- defining the rules to create URIs

Implementing

- Modelling: The first step to build a linked data application is to have data structured following a RDF data model, consisting of triples, with URIs as names for things.
- Cleaning: Ensuring that your metadata is consistent and well structured is of crucial importance since it will affect the quality of the outputs of the reconciling and enriching steps of linked data process. Data profiling methods and tools can be used to help diagnose problems in your metadata in a semi-automated manner, and to deduplicate, normalize and clean it.
- Reconciling: "Terms used in your metadata records can be reconciled with existing and well established vocabularies." This is an easier to implement approach (when compared to full ontologies) to aggregate some level of semantics to your metadata, providing URIs with useful information in a standardized format and links to related URIs. String matching can be used as a low-cost approach to connect your metadata to a controlled vocabulary.
- Enriching: Enriching consists in obtaining structured metadata from unstructured data. Using OCR and named-entity recognition on a full text of a digitized document, for example, more metadata about the contents of the text can be extracted and added in a structured way to the record, becoming available for linking. It is specially useful when dealing with large digitization projects, big data linking or in the realm of digital humanities.

    > Named-entity recognition (NER): *"NER currently provides the easiest and cheapest method of identifying and disambiguating topics in large volumes of unstructured textual documents"* (Hooland & Verborgh, 2014).

Publishing

- Publishing: Linked data should ideally be published in a format that allows both human and machine-interpretation. REST APIs allows you to do that in an elegant and sustainable way, avoiding needless duplication and maintenance.

- The data set should:
    - be linked to other data sets
    - provide provenance of the metadata
    - explicitly indicate license for use
    - adopt terms from well-established controlled vocabularies
    - use dereferenceable URIs
    - map local vocabulary terms to other vocabularies
    - provide set-level metadata
    - provide more than one way to access the data set (e.g., SPARQL endpoint and RDF dumps)

Consuming

- Final user interface
- APIs

Value and Challenges for Libraries and Archives

According to the results of the survey on LD adoption conducted by Online Computer Library Center (OCLC) (Mitchell, 2016a), libraries and archives engaging in linked data projects are looking for:

- *"enriching bibliographic metadata or descriptions,"*
- *"interlinking,"*
- *"a reference source that harmonize data from multiple sources,"*
- *"automate authority control,"*
- *"enrich an application,"*
- *"to publish data more widely,"*
- *"to demonstrate potential use cases and impact."*

The same survey highlights some of the challenges Libraries and Archives are facing in order to implement those projects:

- Inexistence of a formalized and established implementation approach across institutions.
- Lack of an easy-to-implement approach demand high level technological skills from staff.
- Immature software market and tools.
- Non standard approach and guidelines to data licensing for published data.
- Lack of integration of authority resources to linked data tools and services.

Linked Data Cases in Libraries and Archives

The cases described below were selected based on the number of requests per day, meaning that they are the most popular resources among the ones listed on OCLC's 2014 survey (Mitchell, 2016a; 2016b). Examples from both libraries and archives were selected. Some interesting results from the survey are summarized below:

- *"The most commonly used LD data sources (vocabularies) were id.loc.gov, DBpedia, GeoNames, and VIAF."*
- *"Data in the projects analyzed was often bibliographic or descriptive in nature."*
- *"The most common organizational schemas used were Simple Knowledge Organization System (SKOS), Friend of a Friend (FOAF), Dublin Core and Dublin Core terms, and Schema.org."*
- *"Resource Description Framework (RDF) serialized in the eXtensible Markup Language (XML) was commonly used, as was RDF serialized in JavaScript Object Notation (JSON) and Terse RDF Triple Language (Turtle)."*

WorldCat Linked Data Project (Libraries)

Link: https://www.worldcat.org/
Number of requests/day: an average of 16 million (OCLC Research, 2014)
Technology used: URIs, RDF, keyword search.

WorldCat was enhanced in 2014 by the *"addition of URIs from WorldCatWorks (OCLC 2014d), an RDF dataset that is automatically generated from WorldCat catalog records and identifies common content in the editions and formats of particular books, sound recordings, and other resources held in library collections."* (Godby et al., 2015) The motivation behind the project was to make WorldCat records more useful, *"—especially to search engines, developers, and services on the wider Web, beyond the library community"* and *"easier for search engines to connect non-library organizations to library data"*. (Godby et al., 2015)

Library of Congress's (LoC) id.loc.gov service (Libraries)

Link: http://id.loc.gov/about/
Number of requests/day: over 100,000 (OCLC Research, 2014)
Technology used: URIs, keyword search, REST API. RDF/XML, Turtle, or N-triples, are available for bulk download for the authorities and vocabularies (MADS/RDF and SKOS/RDF representations of the data).

"The Library of Congress Linked Data Service enables both humans and machines to programmatically access authority data at the Library of Congress. The scope of the Linked

Data Service is to provide access to commonly found standards and vocabularies promulgated by the Library of Congress. This includes data values and the controlled vocabularies that house them. The main application provides resolvability to values and vocabularies by assigning URIs. Each vocabulary possesses a resolvable URI, as does each data value within it. URIs accessible at id.loc.gov only link to authority data -- that is, controlled vocabularies and the values within them. Therefore, users will not find identifiers for electronic bibliographic resources. The Library of Congress uses other identifier schemes such as Handles for this purpose." (Library of Congress, n.d.)

British Library's British National Bibliography (Libraries)

Link: http://bnb.data.bl.uk/
Number of requests/day: 10,000 – 50,000 (OCLC Research, 2014)
Technology used: URIs, RDF, keyword search, SPARQL queries.

"The BNB Linked Data Platform provides access to the British National Bibliography published as linked open data and made available through SPARQL services. Two different interfaces are provided: a SPARQL editor, and /sparql a service endpoint for remote queries. The Linked Open BNB is a subset of the full British National Bibliography. It includes published books (including monographs published over time), serial publications and new and forthcoming books, representing approximately 3.9 million records. The dataset is available under a Creative Commons CC0 1.0 Universal Public Domain Dedication licence." (British Library, n.d.)

American Numismatic Society's thesaurus (Archives)

Link: http://nomisma.org/
Number of requests/day: 10,000 – 50,000
Technology used: URIs, RDF/XML, JSON-LD, Turtle, KML, SPARQL queries.

"Nomisma.org is a collaborative project to provide stable digital representations of numismatic (relating to or consisting of coins, paper currency, and medals) concepts according to the principles of Linked Open Data. These take the form of http URIs that also provide access to reusable information about those concepts, along with links to other resources. The canonical format of nomisma.org is RDF/XML, with serializations available in JSON-LD (including geoJSON-LD for complex geographic features), Turtle, KML (when applicable), and HTML5+RDFa 1.1." (Nomisma, n.d.)

Archaeology Data Service Linked Open Data (Archives)

Link: http://data.archaeologydataservice.ac.uk/page/
Number of requests/day: fewer than 1,000 (Mitchell, 2016a)
Technology used: URIs, RDF/XML, SPARQL queries.

The Archaeology Data Service *"preserves digital data in the long term, and promotes and disseminating a broad range of data in archaeology, using a variety of avenues, including Linked Open Data. Linked Data at the ADS was initially made available through the STELLAR project (http://hypermedia.research.southwales.ac.uk/kos/stellar/), a joint project between the University of South Wales, the ADS and Historic England. The STELLAR project developed an enhanced mapping tool for non-specialist users to map and extract archaeological datasets into RDF/XML, conforming to the CRM-EH ontology (an extension of CIDOC CRM for archaeology). The results of the STELLAR project are published from the ADS SPARQL endpoint. ADS also consumes LOD from other sources (Library of Congress, Ordnance Survey, GeoNames, DBpedia and the vocabularies developed as part of the SENESCHAL project - http://www.heritagedata.org/blog/about-heritage-data/seneschal) to populate the metadata held within our Collection Management System with URIs, and then publishes the resource discovery metadata for all our archives via our SPARQL endpoint."* (The University of York, n.d.)

Sources of data about LD projects in Libraries, Archives and Museums

These sources offer a comprehensive list of linked data initiatives in libraries, archives and museums, as well as use cases and frameworks for application.

OCLC survey on LD adoption (2015)

"In 2014, OCLC staff conducted a survey on LD adoption, a survey that is being repeated for 2015. The analyzed results from the 2014 survey are captured in a series of blog posts on the site hangingtogether.org and provide a substantial window into the state of LD deployment in LAM institutions.1 The survey surfaced 172 projects, of which 76 included substantial description. Of those 76 projects, over a third (27) were in development." (Mitchell, 2016a)

Links:
Home: http://www.oclc.org/research/themes/data-science/linkeddata.html
Blog: http://hangingtogether.org/?p=4137

Library Linked Data Incubator Group (LLD XG) wiki (2011)

The mission of the LLD XG, chartered from May 2010 through August 2011, has been *"to help increase global interoperability of library data on the Web, by bringing together people involved in Semantic Web activities — focusing on Linked Data — in the library community and beyond, building on existing initiatives, and identifying collaboration tracks for the future."* (W3C Incubator, 2011). They offer a series of generalized and individual use cases of linked data.

Links:
Final report: https://www.w3.org/2005/Incubator/lld/XGR-lld-20111025/
Use cases page: https://www.w3.org/2005/Incubator/lld/XGR-lld-usecase-20111025/
Use cases wiki: https://www.w3.org/2005/Incubator/lld/wiki/Use_Cases

Linked Data for Libraries (LD4L) (2016)

"The goal of the project is to create a Scholarly Resource Semantic Information Store (SRSIS) model that works both within individual institutions and through a coordinated, extensible network of Linked Open Data to capture the intellectual value that librarians and other domain experts and scholars add to information resources when they describe, annotate, organize, select, and use those resources, together with the social value evident from patterns of usage." (Duraspace, 2016) They offer a series of generalized cases, clustered into six main areas including "Bibliographic + Curation" data, "Bibliographic + Person" data, "Leveraging external data including authorities," "Leveraging the deeper graph," "Leveraging usage data," and "Three-site services" (e.g., enabling a user to combine data from multiple sources)." (Mitchell, 2016a)

Links:
Project wiki: https://wiki.duraspace.org/display/ld4l/LD4L+Use+Cases
Paper about the project: http://ceur-ws.org/Vol-1486/paper_53.pdf

References

Alistair, M., Matthews, B., Beckett, D., Brickley, D., Wilson, M. and Rogers, N. (2005) SKOS: a language to describe simple knowledge structures for the web, http://epubs.cclrc.ac.uk/bitstream/685/SKOS-XTech2005.pdf

Berners-Lee, T. (2009). Linked Data. Retrieved October 10, 2017, from https://www.w3.org/DesignIssues/LinkedData.html

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) The Semantic Web, Scientific American, 284 (5), 34-43.

Bray, T. Hollander, D., Layman, A. and Tobin, R. (2006) Namespaces in XML 1.1, 2nd edn, W3C Recommendation, http://www.w3.org/TR/xml-names11/.

British Library. (n.d.). Welcome to bnb.data.bl.uk. Retrieved October 10, 2017, from http://bnb.data.bl.uk/

Broughton, V. (2004) Essential Classification, Facet Publishing.

Duraspace. (2016). LD4L Use Cases. Retrieved October 10, 2017, from https://wiki.duraspace.org/display/ld4l/LD4L+Use+Cases

Fielding, R. T. (2000) Architectural Styles and the Design of Network-based Software Architectures, PhD thesis, University of California, Irvine, CA.

Godby, C. J., Wang, S., & Mixter, J. K. (2015). Library Linked Data in the Cloud: OCLC's Experiments with New Models of Resource Description. Synthesis Lectures on the Semantic Web: Theory and Technology, 5(2), 1–154. http://doi.org/10.2200/S00620ED1V01Y201412WBE012

Harpring, P., & Baca, M. (2010). 1. Controlled Vocabularies in Context. In Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works (pp. 1–11). Retrieved from http://www.getty.edu/research/publications/electronic_publications/intro_controlled_vocab/

Hay, D. C. (2006). Data Modeling, RDF, & OWL - Part One: An Introduction To Ontologies. The Data Administration Newsletter, (April). Retrieved from http://www.tdan.com/view-articles/5025

Hooland, S.; Verborgh, R. (2014). Linked data for libraries, archives and museums : how to clean, link and publish your metadata. Neal-Schuman.

JSON-LD. (2017, October 17). In Wikipedia, The Free Encyclopedia. Retrieved October 18, 2017, from https://en.wikipedia.org/w/index.php?title=JSON-LD&oldid=805736276

Legg, C. (2007). Ontologies on the Semantic Web. Annual Review of Information Science and Technology, 41(1), 407–451. http://doi.org/10.1002/aris.2007.1440410116

Library of Congress. (2016). Overview of the BIBFRAME 2.0 Model. Retrieved October 10, 2017, from https://www.loc.gov/bibframe/docs/bibframe2-model.html

Library of Congress. (n.d.). About Linked Data Service. Retrieved October 10, 2017, from http://id.loc.gov/about/

Miessler, D. (2015). The Difference Between URLs and URIs. Retrieved October 10, 2017, from https://danielmiessler.com/study/url-uri/

Miller, S. J. (2011). Metadata, Linked Data, and the Semantic Web. In Metadata for Digital Collections (pp. 303–324).

Mitchell, E. T. (2016a). Library Linked Data: Early Activity and Development. Library Technology Reports (Vol. 52). http://doi.org/10.5860/ltr.52n1

Mitchell, E. T. (2016b). Chapter 1. The Current State of Linked Data in Libraries, Archives, and Museums. Retrieved October 10, 2017, from https://journals.ala.org/index.php/ltr/article/view/5892/7446

N-Triples. (2017, September 24). In Wikipedia, The Free Encyclopedia. Retrieved October 18, 2017, from https://en.wikipedia.org/w/index.php?title=N-Triples&oldid=802208118

Nomisma. (n.d.). Retrieved October 10, 2017, from http://nomisma.org/

OCLC Research. (2014). Linked Data Survey results 1 – Who's doing it (Updated). Retrieved October 10, 2017, from http://hangingtogether.org/?p=4137

Olson, J. (2003) Data Quality: the accuracy dimension, Morgan Kaufmann.

Southwick, S. B. . (2015). A Guide for Transforming Digital Collections Metadata into Linked Data Using Open Source Technologies. Journal of Library Metadata, 15(1), 1–35. http://doi.org/10.1080/19386389.2015.1007009

The University of York. (n.d.). Archaeology Data Service Linked Open Data. Retrieved October 10, 2017, from http://data.archaeologydataservice.ac.uk/page/

Turtle (syntax). (2017, September 24). In Wikipedia, The Free Encyclopedia. Retrieved October 18, 2017, from https://en.wikipedia.org/w/index.php?title=Turtle_(syntax)&oldid=802208209

Uniform Resource Identifier. (2017, October 14). In Wikipedia, The Free Encyclopedia. Retrieved October 18, 2017, from https://en.wikipedia.org/w/index.php?title=Uniform_Resource_Identifier&oldid=805285595

University of British Columbia. (n.d.). Open Collections API Documentation. Retrieved October 10, 2017, from https://open.library.ubc.ca/docs

W3C Incubator. (2011). Library Linked Data Incubator Group Final Report. Retrieved October 10, 2017, from https://www.w3.org/2005/Incubator/lld/XGR-lld-20111025/