Submitted on: 8/1/2014

An unbroken chain: approaches to implementing Linked Open Data in libraries; comparing local, open-source, collaborative and commercial systems

Rurik Greenall
Developer, Section for Information Resources, Norwegian University of Science and Technology (NTNU) Library, Trondheim, Norway.
E-mail address: rurik.greenall@gmail.com

Lukas Koster
Library Systems Coordinator, Library and Information Systems Department, Library of the University of Amsterdam (UvA), Amsterdam, The Netherlands.
E-mail address: l.koster@uva.nl

ABSTRACT:

This paper compares methods for libraries to interact with the Web of data by assessing the benefits and risks associated with local development, free-and-open-source software, collaborative and commercial solutions. Through a number of case studies, we provide insight into how each approach can be implemented and the extent to which these approaches can be reconciled.

Keywords: Linked open data, systems development, systems administration, libraries.

1 Introduction

Libraries face an important decision for the future: continue doing what they have always done and run the risk of becoming irrelevant, or seize the opportunities of the Web and broaden their scope to meet the demands of a developing audience. This section introduces some of the challenges faced by different kinds of libraries that choose the latter route.

One can distinguish a number of types of libraries, among others, public, academic, research, special and national libraries. They all have their own, sometimes overlapping, audiences and objectives and all work towards reaching their audiences and achieving their objectives in a number of well-established ways, focusing on a limited number of content types such as books, articles, multimedia and reference material. The Web is dramatically changing the nature and forms of the distribution of information. All libraries face similar developments; they all have to make choices regarding their business:

- how best to fulfil their responsibilities,
- what content to focus on,
- which services to offer and how to implement them.

1.1 Objectives

If libraries want to remain relevant, they have to make effective use of the Web; it is widely agreed that Linked Open Data (LOD) [LOD] is an appropriate technology for this. For LOD implementations, libraries have to make changes in two areas: content and systems. Roughly speaking, content is about open data, systems are about linking.

There are three objectives for which libraries may want to adopt LOD:

- exposing their own data as LOD for others to consume and re-use,
- consuming and re-using LOD from external data sources to enrich and contextualize their own data,
- using LOD as the basis for a completely new vendor-independent, Web-based infrastructure and workflow for cataloguing (the objective of the Library of Congress BIBFRAME initiative [BIBFRAME]).

We cover different approaches to these objectives in the following sections.

1.2 Content

All libraries depend on external content providers alongside locally produced/owned content. This applies to traditional bibliographic metadata, full-text content and other data.

Data licensing has not traditionally been a core concern of libraries. In many cases, no license exists and ownership of data cannot be ascertained; traditional library systems have rarely recorded data provenance. Intellectual property rights over data are also often unclear. The first job of any party attempting to publish data is to investigate ownership and licensing.

1.3 Systems

For systems, there are a number of possibilities: local development; free-and-open-source software (FOSS); national, regional or thematic portals/consortia/service centres; and commercial vendors. Of course, the typical situation will be a combination of these possibilities.

Most libraries use an integrated library system (ILS, LMS, etc.) for acquisition, cataloguing, circulation, with a public Web interface for finding and requesting material. Academic and research libraries usually have link resolvers, institutional repositories (IR) for disseminating their scholarly publications, and current research information systems (CRIS) for internal research project administration. If they have special collections they will also have systems to expose digitized objects. A growing number of libraries also have discovery tools, in which local data from ILS, IR and possibly other local data sources are presented together, often with data from external commercial and open access databases, to create a unified search environment.

1.4 A note on LOD

The understanding of LOD used in this paper is: RDF [RDF] data containing links to resources so that the data is not a contiguous whole, distinct from the Web of data, but is rather part of the Web of data. Thus, LOD differs from a database that is published online without links, and can rather be understood as a distributed database where each new dataset integrates information into the whole. Further, this understanding assumes that data is consumed from third parties, via the mechanism of dereferencing linked resources (and possibly cached and indexed). This is an open-world approach, which also entails aspects of trust and strategies for cases where this is lacking.

1.5 About mapping, ETL and creating original RDF

RDF should be preferred for data creation, processing and storing, if completely new systems are being developed, typically for cataloguing. In this case, a one-time data-conversion-and-migration action is needed to get started. If legacy systems are to be maintained for the core workflows of the organization, other methods must be employed for working with RDF data.

On-the-fly object mapping [ORM] — the declaration of correspondence between data elements in one or more data stores and another schema — has a low threshold technology-wise as data can quickly be mapped to RDF. A major issue with this approach is that it assumes access to the data; this is often not the case in library contexts. Indirect access in the form of database dumps creates another barrier as these must then be regularly synchronized, an activity which falls under lifecycle planning.

There are some potential limitations in terms of linking, so that this approach may not easily produce links to outside resources. There is a basic conceptual issue: is mapping interesting to an organization wishing to replace a legacy technology stack? In some ways, mapping has an unfortunate side-effect, giving legacy systems a new lease of life while ignoring core benefits available in the new technology. At the same time, slotting this approach into a wider development plan — where generating RDF is the first goal — makes a lot of sense.

In some cases, it may be relevant to consider simply converting data; data is extracted from its original context, transformed to some new format and then loaded into a new context. The extract-transform-load (ETL) process consists of downloading data in some format, creating RDF, adding links and then loading it into a store that serves data to the Web. This approach is development intensive if it is to be done properly; it is simple to do single-shot ETL with linked data, e.g. with concluded datasets; ETL can also provide initial data in new systems, but creating a workflow that does repeated ETL reliably is more complex in nature and requires a development lifecycle plan.

Indeed, data access is the major issue again. In cases where data is available via e.g. OAI-PMH [OAI-PMH], there is some chance of simple updates, while other options may end in the worst-case scenario: bulk re-processing of entire datasets.

Of course, from a development standpoint, the transformation phase of ETL is extremely flexible since anything is possible. There are many tools designed specifically for this job, from graphical desktop packages to command line tools. This freedom means planning is needed to rein in directions of investigation so that the process does not grow out-of-hand.

1.6 Issues

A problem with converting legacy metadata to RDF is that the source data consists mostly of string values that do not easily map to semantic objects, so the result cannot really be considered to be linked data, but “linkable” data at best. Note also that on-the-fly processing of literals produces results that are only as consistent as the original data, and that it is often necessary to process bibliographic data extensively for it to be mappable to datatypes other than plain strings.

Also, content providers typically do not provide open data, even if library systems fully support consuming linked data; thus, there will be no appropriate LOD to re-use and libraries will simply be left empty handed. Similarly, if libraries finally manage to publish their own LOD using built-in library system functionality and nobody wants to re-use that, what has been achieved in the end?

1.7 About the authors

Rurik Thomas Greenall works at NTNU, the Norwegian university of science and technology with software development. The major foci for this work are REST, LOD and Web APIs. His work typically integrates components familiar from the Web and linked-data stacks as well as elements from more traditional library systems. As of August 2014, Greenall will be a member of staff at Computas AS, an IT consultancy, where he will work with semantic Web applications. He will work with Oslo public library’s development team, on behalf of Computas.

As Library System Coordinator, Lukas Koster deals with commercial systems used by the Library of the University of Amsterdam for their core library workflows and functions. These systems are provided by Ex Libris: Aleph ILS, SFX Link Resolver, Primo Discovery Tool, bX Recommender Service. These systems are highly customized by local staff to fulfil local requirements. The Library of the UvA also uses a number of Open Source tools for other functions, among others, developing their own repository systems. Lukas Koster has been active in the independent International Group of Ex Libris Users IGeLU, most recently as coordinator and member of the Linked Open Data Special Interest Working Group which is working with Ex Libris to implement LOD features in their systems.

2 Library LOD and local development, including foss and commercial software libraries

2.1 Context

Local development entails the creation of software architecture that provides functionality that is either entirely novel or extends a less specific software package to a new application.

From a developer’s point of view, there is little difference between using FOSS [FOSS] and commercial software. A commercial package may come with licence restrictions and a support contract; similarly, FOSS has a licence and it is often possible to sign support contracts with specialized consultancies. Using any software package entails some work in implementing it in a wider infrastructure; for both commercial software and FOSS, this may take the form of deploying binary files to an application server and using a configuration interface, or using a software library that provides functionality in a given programming language. In the latter case, there is absolutely no distinction between commercial and FOSS software except when the software library has a particularly restrictive licence, or the product that is being worked on is itself to be released as FOSS. The pros and cons of the software licensing agreement must be weighed against the functionality of the software, its place in wider architecture and terms of support. The only important factor in linked open data is that no claim is made on the ownership of data.

In all cases, the lifecycle of individual software libraries and packages in use must be considered; lifecycle is an important issue in any development process and since software projects evolve rapidly, lifecycle planning must be central. Libraries are traditionally not good at lifecycle planning, as witnessed by the majority of metadata records, which are created, signed and abandoned. Similarly, developing software must change from a "product creation" process to a lifecycle-oriented process, where feedback fuels continuing development.

The work of development is often broken down into phases of planning, analysis, design, implementation and maintenance, where maintenance feeds back into more planning for the future of the product. Formalizing these tasks into a proper development lifecycle is rather challenging for an organization not primarily concerned with software development.

2.2 Cataloguing

While cataloguing currently — and likely for some time in the future — takes place in traditional systems, at some point, (possibly at the outset in the case of new collections) it is relevant to consider whether it is preferable to catalogue directly as linked data. This necessarily entails a re-thinking of the aims of cataloguing: do we want to produce a standardized format as is currently the case, or create a local format that supports local use cases and can be transformed to create a shareable exchange format?

Creating a system for cataloguing in linked data directly is no simple task; there are many options: using ontology modelling tools, file-based systems, database technologies or entirely RDF-based workflows.

For simplicity, ontology modelling tools seem to be a good choice as they support many of the functions needed out of the box. This simplicity belies the fact that such tools provide weaker support for workflows and therefore require more discipline from the users.

If a new backend is developed in file-systems, databases or RDF-databases, there are many issues to be considered, not least of which are the ways in which CRUD (create, read, update, delete) operations are performed and documented. Here it is important to document how the system achieves its tasks as this is an area that is rapidly developing; many current RDF-databases provide support for meta-metadata in a proprietary way, however, there is work underway on standardizing this.

It is important to understand that — in all but the most generic RDF-editing interface — the frontend for any of these systems is tied to the concept of workflow; it represents how resources are understood within the workflow. As such, the interface is a materialization of the workflow, and is something that exists independently of and is largely irrelevant to the concepts of RDF and LOD. This is clearly seen in the cases of the LibrisXL [LibrisXL] and BIBFRAME Scribe [BIBFRAME Scribe] editors (although these are quite likely adaptable to other workflows).

Critically, however, the major pressure point is not in the development team, but the staff interacting with the software, as this is unlikely to be familiar "off-the-shelf" and must be expected to change over time.

One cannot underestimate change management; there should be focus on feeding information back into the development cycle to improve the understandability and functionality of the tools. Development phases depend on feedback to create working products over time. It is wise to plan transition in terms of familiarity: tools incrementally improve and become less like the familiar systems and more flexible and Web-oriented. This kind of lifecycle planning requires different kinds of expertise than plain programming, and a balance in the development team between developers familiar with old and new technologies.

2.3 Exposing

How data is exposed depends largely on the methodology that is used. In cases where data is sourced from an existing non-RDF database, tools used for object mapping typically provide support for interacting with RDF data via a SPARQL [SPARQL] endpoint or some other API [API].

Data may also be transferred into an RDF database that provides access to data via SPARQL or other APIs.

It is common practice to provide content-negotiable views of the data, such that RDF or HTML and other formats may be accessed. This functionality may be provided by mapping tools or be custom-built into a REST [REST] service.

2.4 Consuming

Because the developers have control of the interface, it is simple for data to be consumed from the resources that are contained in the linked data. There are many suggested best practices [Linked data patterns], however, much work needs to be done in order to create systems that function well from a user perspective. Many consumption patterns easily lead to unfamiliar (and less usable?) interfaces unless measures are taken to counterbalance the effects of architectural choices.

2.5 Example: NTNU digital collections

NTNU University Library special collections have a LOD-based online catalogue for their digital collections [NTNU UB / Digital].

Cataloguing is performed in two separate streams, one that revolves around the conceptual metadata for the item and a second that documents the digital files; these streams merge to provide access points to the digital files. The former stream takes place via a Web interface built on top of cloud-based tabular storage that allows data to be created and linked, the latter via an automated process based on XMP [XMP].

The data from tabular storage is transformed to RDF via a simple ETL process and loaded into a triplestore. The triplestore is in turn exposed to the Web via a simple Web application that serves HTML/RDF/JSON+RDF [JSON] via a REST interface. In all views, base data is provided for the selected RDF node; in HTML and JSON+RDF, the data includes data from linked resources.

A search mechanism is provided by allowing an indexer to crawl the JSON+RDF of each and every resource.

The system as described actually comprises four separate applications, a creation module (tabular storage), an RDF store with ETL interface, a REST interface for data and an indexing/search application. It should be noted that all systems interact via REST APIs, either standards-compliant SPARQL and RDF, or proprietary JSON API (in the case of search/indexing).

Another example of a local library development project can be found in Oslo public library, where they are developing various novel semantic Web components, alongside developing software on top of the open-source library system, Koha for various library functions. This merging of two development styles and technology stacks is interesting for a number of reasons, not least because it uses an existing technology stack for areas that are already well served by traditional library systems (lending, ILL and so forth). [Digital Deichman]

3 Library LOD and consortia/service centres

3.1 Context

Libraries often participate in consortia and service centres; these organizations feature skilled staff and administrative routines well suited to software development. This given, there is potential that these collaborations can provide necessary support to libraries interested in exposing LOD.

3.2 Cataloguing

The data in consortia are most often provided through collaborative ILSes, sourced from participating cataloguing departments in the consortia, centralized cataloguing centres, and import of data. Cataloguing typically relies on a familiar mix of local and international standards.

A transition from this situation to LOD is simply a matter of finding systems that provide support for LOD cataloguing. As it stands, there are currently no LOD-oriented library systems, however, there are a number of commercial and open source alternatives that consortia may choose to adapt.

The trend, however, has been for consortia to use ETL processing to transform data sourced from non-LOD systems, and we have seen investment in this particularly in Germany (HBZ [LOBID], BVB/KOBV [B3KAT]). Given that LOD is already being produced, the motivation for a movement towards true LOD cataloguing will be limited, except in cases where value of LOD is proven, but the ETL processes are seen as flawed.

3.3 Exposing

Consortia have typically proven adept at providing standards-compliant APIs for data, adding content negotiation to complement existing systems so that they provide ETL-processed RDF/LOD. Data may also be exposed via SPARQL endpoints and data dumps.

3.4 Consuming

In ETL processes, data is consumed from third parties to provide linking. These links are then dereferenced (accessing the data that the URI (Uniform Resource Identifier) points to) and the data is processed to enrich HTML views, in many cases, via lightweight AJAX services that integrate the LOD external links into an existing non-LOD catalogue.

3.5 Example: BIBSYS-NTNU collaboration

In 2010, NTNU Library and the consortium, BIBSYS, received funding from the Norwegian Archive, Library and Museum Authority to create a LOD dataset based on the consortial personal name authority file (c.f. [RADATANA]).

For NTNU, working with BIBSYS was simple as are both are situated in Trondheim and have generally strong ties and communicate well.

Mappings for the data were produced by NTNU and sent to BIBSYS who produced LOD and exposed this according to project specifications. The process was smooth and the dataset was operational in a within six months. BIBSYS had easy access to the data and a comprehensive understanding of the data structure; this, in addition to professional data administration routines ensured regular updates to the dataset until these were cancelled.

Takeaways from the project: creating and exposing LOD is simple, creating a successful and useful dataset is less simple. The data management and licensing aspects of the project are more difficult and (in the latter case) potentially have no quick fix.

As a footnote to this discussion, it is worth mentioning the developments at the service centre of the Swedish royal library, LIBRIS. LIBRIS is developing Libris XL, of which the Libris XL cataloguing module mentioned above [Libris XL] is a component, an entirely new library system, based on linked data principles. This project is an exciting development that all libraries should pay attention to.

4 Library LOD and commercial system vendors

4.1 Context

Few libraries are in a position to develop, update and administer their own technical infrastructure or library systems. Most libraries depend to some extent on external commercial system providers for their basic workflows.

4.2 Cataloguing

A library using a commercial ILS wanting to switch to cataloguing based on LOD principles must simply wait as no system for this currently exists. All parties are closely monitoring progress in the BIBFRAME initiative before they start investing in a completely new infrastructure. Commercial vendors currently develop “next-generation-library-back-offices-in-the-cloud” based on current library practices and metadata formats. Obviously, these systems will form starting points for any coming (LOD?) infrastructure. It will take some time before LOD is part of the roadmap for commercial library system vendors and open source library systems.

In the short term, libraries may want to expose their own data contained in a commercial ILS or CRIS as LOD, or embed LOD from external sources in their online catalogues or discovery interfaces.

4.3 Exposing

Regarding exposing bibliographic and holdings information as LOD, as pointed out above, no current system directly supports linked data. However, a library system usually offers a number of bulk or selective offline export options: an API and something like a “show/save as MARC/RIS” option for individual results. All of these can, in theory, be used to convert data to some version of processable RDF, depending on the degree of openness of the system. Doing this, however, requires qualified technical developers or a third party who can do it for you. And, as stated above, the result of converting legacy string data to RDF will be linkable data at best.

Another option is to contribute bibliographic and holdings data to Worldcat, which offers LOD for each record, including links to authorities among others. Again, the same limitations apply: what OCLC considers to be LOD may not match local intentions.

4.4 Consuming

Server-side consumption and display of linked data from external sources is not realistic because legacy systems can’t handle and process URIs and RDF. Even a locally developed enrichment add-on has issues as the systems still can’t handle the added information and URIs in the user interface. Client-side development is only possible if the library can customize the frontend of the legacy system by adding client side scripts, getting external data and manipulating HTML. This case, too, requires local, qualified frontend-development capacity.

There is a constant risk that locally developed or implemented add-ons and workarounds for existing systems will stop working with the next upgrade of the core system, if the data structure, system functions or frontend HTML that they depend on changes. This is why it is better to work with the vendor to add enhanced/new functionality to the core system. This applies not only to LOD features, but to any missing functionality.

4.5 Example: Experiences with Ex Libris

Ex Libris is a commercial library system vendor. In 2011, the independent International Group of Ex Libris Users (IGeLU) established the Special Interest Working Group on LOD (LOD SIWG) following a session at the annual IGeLU conference that showcased a number of customer projects implementing LOD add-ons for Ex Libris products.

The group published a manifesto [IGeLU LOD] explaining their objectives. Ex Libris appointed their chief strategy and technology officers as official liaisons. In 2013, the group joined forces with the North American user group, ELUNA, and published a document containing the view of the SIWG as a starting point for soliciting submission of use cases from customers. The document is only accessible for Ex Libris staff and customers.

Since August 2013, monthly calls, open to all Ex Libris staff and customers, are organized where one scenario presented by a customer is discussed. Between 2011 and the present date, the group has also been active in organising discussion sessions and use-case presentations on LOD at the annual IGeLU and ELUNA conferences.

In December 2012, Ex Libris announced its membership of the W3C Schema Bib Extend Community Group in a press release titled "Ex Libris Promotes Linked Data Techniques to Make Library Data More Open and Accessible" [W3C], in which the collaboration with the user groups is explicitly mentioned.

Ex Libris has made it clear that they will focus on LOD-feature implementation in the Primo discovery tool. From a vendor's point of view, this makes sense because it is their most visible product and suitable for quick wins.

Currently, Ex Libris is working on a first project focused on mapping their internal Primo record format to RDF in order to make it possible to request RDF from local holdings in the Primo index. This seems strange because it entails publishing local metadata as linked data using a platform that would be particularly suitable for consuming and re-using external LOD, however, this is a positive development: small steps must be taken to reach the greater objective. The first step is to gain experience expressing proprietary bibliographic and holdings information as RDF; at the same time as replacing some of the simple string values with URIs. A next step could be support for dereferenceable URIs in the internal Primo record format. The first results, JSON-Context files for Primo, have been published by Ex Libris on Github [Ex Libris JSON], whatever these represent.

5 Working with non-library vendors

Of the authors, Greenall has experience working with several non-library vendors on library and non-library LOD projects; these experiences give cause to believe there is value in looking beyond traditional library vendors when looking for technology partners.

Local knowledge of traditional library concepts and formats can be brought into play with people that really understand LOD and semantic Web technologies. The benefit here is that one can stand on a more equal footing with the vendor, who is likely used to working with customers to adapt a LOD-oriented system, rather than attempting to adapt an existing non-LOD system to provide LOD.

This approach requires stable partners because the partnership may run for many years and partners that take the local "small scale" needs seriously. Remotely hosted solutions mean that this approach may be neither costly nor inconvenient to maintain over time.

Perhaps the most intriguing aspect of this approach is that it might shed new light on how to approach doing what libraries do because the external development team does not have library preconceptions.

6 Reconciling approaches

It is not possible to speak of one approach to the exclusion of an other; vendors work with customers, use FOSS in their products, etc. Working with a commercial provider or a consortium is professionally similar; a commercial provider who respects customers and is open to development advice can provide similar experiences to those NTNU had with BIBSYS. And local developers can outsource data storage and indexing to commercial providers.

For many libraries, the reality is working with a commercial partner, who takes the library's ideas and puts them into practice. Although library vendors have been very open in some cases, they have simply not had the skills necessary nor the time to acquire the skills.

Platform-as-a-service and software-as-a-service approaches offer something of a reconciliation here, therefore, there is a clear opportunity for a semantically oriented non-library provider (both in terms of a partnership with libraries and library vendors).

The various roads to implementation of LOD in libraries can be compared using a number of criteria:

- Fulfillment - to what extent can requirements be accomplished?
- Time frame - how long does it take to accomplish requirements?
- User base - how many libraries benefit from the implementation?

In local development situations, possibly with FOSS, libraries have the opportunity to achieve all their requirements within the foreseeable future — they also have the opportunity to subsume themselves in expensive vanity projects. Stakeholders' agreement about requirements is paramount and the development project has to be managed well.

For FOSS development, more users hopefully benefit and there is a valid argument for developing software that is local-infrastructure agnostic; this is particularly pertinent in relation to standards and proprietary extensions, which have caused migration headaches previously.

This pattern is repeated with commercial vendors and consortia/service centres: a large number of libraries and their customers will benefit. There is a trade-off, however: it will take much longer for commercial vendors to develop and apply LOD features in their product suite; the requirements will be addressed in the order determined by vendor prioritisation and only optionally in collaboration with their customers and user groups.

Consortia, service centres and similar co-operatives may be the best of both worlds: they have the requirement to provide value to stakeholders and not shareholders; participating libraries get direct benefit from feeding back to their listening partner.

7 Conclusions

A concise summary of the spectrum from local development to fully commercially supported development is: those that can develop do, those that can't buy services. There are many good reasons for not wanting to develop or host all solutions locally, as well as many good reasons to do so. There are also a lot of bad reasons both for and against; these are typically footed in entrenched attitudes and organizational challenges.

One of the major conclusions is that there is nothing particularly hard about the semantic Web in terms of technology, what is difficult is creating things that are useful and used. The holistic product is difficult to define and produce, thus, it is important that the development lifecycle of the product is maintained throughout the life of the product. For this to happen, the feedback loop must be clear of interference and the direction in which the product is to travel must be well understood; this is true of all types of development.

In order to ensure sustainable lifecycle planning in implementing library LOD, it is vital for individual libraries to adhere to de facto standards and best practices. In the absence of these, influential individual libraries may contribute to their creation. In most cases, however, it is recommended that libraries be part of a larger whole of some kind in order to contribute to and benefit from the widespread promotion and support that collaboratives can offer. These collaboratives may take the form of FOSS communities, library consortia, service centres and commercial vendor usergroups.

8 Acknowledgements

The authors would like to thank the anonymous reviewers for their comments and suggestions to improve the quality of the paper. All errors/omissions remain the sole responsibility of the authors.

9 References

[API] Wikipedia. (2014). Application Programming Interfaces [Website]. Retreived 2014-06-25 from http://en.wikipedia.org/wiki/Application_programming_interface.

[B3KAT] B3Kat. (No date). Gemeinsamer Verbundkatalog von BVB und KOBV [Website]. Retrieved 2014-06-14, from http://www.b3kat.de/

[BIBFRAME] Library of Congress. (2014). Bibliographic Framework Initiative [Website]. Retrieved 2014-03-05, from http://www.loc.gov/bibframe/.

[BIBFRAME Scribe] Zepheira. (No date). BIBFRAME Scribe editor [Website]. Retrieved 2014-06-24 from http://editor.bibframe.zepheira.com/static/.

[Digital Deichman] Westrum, Anne-Lena. (2014). Oslo Public Library chooses RDF linked data as core metadata format [Website]. Retrieved 2014-06-25 from http://digital.deichman.no/blog/2014/06/19/oslo-public-library-chooses-rdf-linked-data-as-core-metadata-format/.

[Ex Libris JSON] Sanders, Shlomo. (2014). JSON LD context for Primo PNX documents [Website]. Retrieved 2014-06-25 from http://github.com/ExLibrisGroup/primo.PNX-context

[FOSS] Wikipedia. (2014). Free and Open Source Software [Website]. Retreived 2014-06-25 from http://en.wikipedia.org/wiki/Free_and_open-source_software.

[IGeLU LOD] IGeLU Linked Open Data Special Interest Working Group. (2012). Manifesto for Linked Open Data in Ex Libris products [Website]. Retrieved 2014-06-25 from http://igelu.org/special-interests/lod/manifesto.

[JSON] Wikipedia. (2014). JavaScript Object Notation [Website]. Retreived 2014-06-25 from http://en.wikipedia.org/wiki/Json.

[Libris XL] Libris. (2014). Libris Katalogisering Beta [Website]. Retreived 2014-06-24 from http://kat.libris.kb.se/login.

[Linked data patterns] Davis, Ian & Dodds, Leigh. (2012). Linked data patterns [Website]. Retrieved 2014-06-23 from http://patterns.dataincubator.org/book/

[LOD] Berners-Lee, Tim. (2006). Design Issues: Linked data [Website]. Retrieved 2014-06-25 from http://www.w3.org/DesignIssues/LinkedData.html.

[LOBID] Lobid. (No date). Linking open bibliographic data [Website]. Retrieved 2014-06-14, from http://lobid.org/

[NTNU UB / Digital] NTNU University Library. (No date). NTNU University Library special collections [Website]. Retrieved 2014-06-23 from http://www.ntnu.no/ub/digital.

[OAI-PMH] Open Archives Initiative. (No date). Open Archives Initiative Protocol for Metadata Harvesting [Website]. Retrieved 2014-06-25 from http://www.openarchives.org/pmh/.

[ORM] Wikipedia. (2014). Object-relational mapping [Website]. Retrieved 2014-06-25 from https://en.wikipedia.org/wiki/Object-relational_mapping.

[RADATANA] Greenall, R. et al. (2011). Rådata nå! [Website]. Retrieved 2013-03-05, from https://docs.google.com/document/d/1a0mBNdKPBMOGjhDYhfh5oqJMtR8if6ifHr2mJ_PGLio.

[RDF] W3C. (2014). RDF [Website]. Retrieved 2014-06-25 from http://www.w3.org/RDF/.

[REST] Wikipedia. (2014). Representational state transfer [Website]. Retrieved 2014-06-25 from http://en.wikipedia.org/wiki/Representational_State_Transfer.

[SPARQL] W3C. (2013). SPARQL Query Language for RDF [Website]. Retrieved 2014-06-25 from http://www.w3.org/TR/rdf-sparql-query/.

[W3C] Ex Libris Group. (2012). Ex Libris Promotes Linked Data Techniques to Make Library Data More Open and Accessible [Website]. Retrieved 2014-03-05, from http://www.exlibrisgroup.com/default.asp?catid=%7B337F23ED-722A-45C2-B6DD-29842B7D5B13%7D&details_type=1&itemid=%7B97C7DCE3-2AF6-442A-88C0-9E7C8299EA9B%7D.

[XMP] Wikipedia. (2014). Extensible Metadata Platform [Website]. Retrieved 2014-06-25 from http://en.wikipedia.org/wiki/Extensible_Metadata_Platform.