

Transitioning from XML to RDF: Considerations for an Effective Move Towards Linked Data and the Semantic Web

Juliet L. Hardesty

INTRODUCTION

Metadata, particularly within the academic library setting, is often expressed in eXtensible Markup Language (XML) and managed with XML tools, technologies, and workflows. Software tools such as the Oxygen XML Editor and querying languages such as XPath and XQuery over time have become capable of helping that management. However, managing a library's metadata currently takes on a greater level of complexity as libraries are increasingly adopting the Resource Description Framework (RDF). Semantic Web initiatives are surfacing in the library context with experiments in publishing metadata as Linked Data sets, BIBFRAME development using RDF, and software developments such as the Fedora 4 digital repository using RDF.

Challenges are evident when considering examples of transitions from XML into RDF and show the need for communication and coordination between efforts to incorporate and implement RDF. This article outlines these challenges using different use cases from the literature and first-hand experience. The follow-up discussion considers ways to progress forward from metadata formatted in XML to metadata expressed in RDF. The options explored are not only targeted to metadata practitioners considering this transition but also to programmers, librarians, and managers.

LITERATURE REVIEW AND CONCEPTS

As an initial example of the challenges faced when considering RDF, clarifying terminology is still a helpful activity. RDF focuses on sets of statements describing relationships and meaning. These statements consist of a subject, a predicate, and an object (i.e., an article, has an author, Jane Smith). These statement parts are also referred to as a resource, a property, and a property value. Since there are three parts to RDF statements, they are referred to as triples. The predicate or property of an RDF statement defines the relationship between the subject and the object. RDF ontologies are sets of properties for a particular domain. For example, Darwin Core has an RDF ontology to express biological properties,¹ and EBUCore has an RDF ontology to express properties about audiovisual materials.²

Pulling apart the many issues involved in moving from XML to RDF is an exploration into the

Juliet L. Hardesty (jlharden@iu.edu) is Metadata Analyst at Indiana University Libraries, Bloomington, Indiana.

purpose of metadata, the tools available and their capabilities, and the various strategies that can be employed. Poupeau rightly states that XML provides structural logic in its hierarchical identification of elements and attributes, where RDF provides data logic declaring resources that relate to each other using properties.³ These properties are ideally all identified with single reference points (Uniform Resource Identifiers or URIs) rather than a description encased in an encoding. A source of honest confusion, however, is that RDF can be expressed as XML. Lassila's note regarding the Resource Description Framework specification from the World Wide Web Consortium (W3C) states, "RDF encourages the view of 'metadata being data' by using XML (eXtensible Markup Language) as its encoding syntax."⁴ So even though RDF can use XML to express resources that relate to each other via properties, identified with single reference points (URIs), RDF is itself not an XML schema. RDF has an XML language (sometimes called, confusingly, RDF, and from here forward called RDF/XML). Additionally, RDF Schema (RDFS) declares a schema or vocabulary as an extension of RDF/XML to express application-specific classes and properties.⁵ Simply speaking, RDF defines entities and their relationships using statements. There are various ways to make these statements, but the original way formulated by the W3C is using an XML language (RDF/XML) that can be extended by an additional XML schema (RDFS) to better define those relationships. Ideally, all parts of that relationship (the subject, predicate, object, or the resource, property, property value) are URIs pointing to an authority for that resource, that property, or that property value.

An additional concept worth covering is *serialization*. This term is used as a way to describe how RDF data is expressed using various formatting languages. RDF/XML, N-triples, Turtle, and JSON-LD are all examples of RDF serializations.⁶ Describing something as being in RDF really means the framework of subject, predicate, object is being used. Describing something as being expressed in RDF/XML or JSON-LD means that the RDF statements have been serialized into either of those formatting languages. Using "RDF" to refer not only to the framework to describe something (RDF) but also the serialization of that description (RDF/XML) can easily muddle the discussion.

Other thoughts about the difference between XML and RDF or moving metadata from XML into RDF point to the difference in perspective and the change in thinking that is required to manage such a move. In an online discussion about RDF in relation to TEI (Text Encoding Initiative), Cummings talks about the need for both XML and RDF, using XML to encode text and RDF to extract that data and make it more useful.⁷ Yee, in her in-depth look at bibliographic data as part of the Semantic Web, points out that RDF is designed to encode knowledge, not information.⁸ The RDF Primer 1.0 also states "RDF directly represents only binary relationships."⁹ XML describes what something is by encoding it with descriptive elements and attributes. RDF, on the other hand, constructs statements about something using direct references—a reference to the thing itself, a reference to the descriptor, and a reference to the descriptor's value. As Farnel discussed in her 2015 Open Repositories presentation about the University of Alberta's move to RDF, they learned they were moving from a records-based framework in XML to a things-based framework in RDF.¹⁰ What is pointed out here time and again is something else Farnel discussed—moving from XML to

RDF is not simply a conversion between encoding formats; it is a translation between two different ways of organizing knowledge. It involves understanding the meaning of the metadata encoded in XML and representing that meaning with appropriate RDF statements.

The tools most commonly employed for reworking XML into RDF are OpenRefine when accompanied by its RDF extension; a triplestore database such as OpenLink Virtuoso,¹¹ Apache Fuseki,¹² or Sesame¹³; Oxygen XML Editor¹⁴; and Protégé,¹⁵ an ontology editor. OpenRefine is, according to the website, “a powerful tool for working with messy data: cleaning it; transforming it from one format into another; and extending it with web services and external data.”¹⁶ The RDF extension, called RDF Refine, allows for importing existing vocabularies and reconciling against SPARQL endpoints (web services that accept SPARQL queries and return results).^{17,18} SPARQL is similar to SQL as a language for querying a database, but the syntax is specifically designed to allow for querying data formatted in triple statements instead of tables with columns.¹⁹ Triplestore databases such as OpenLink Virtuoso can store and index RDF statements for searching as a SPARQL endpoint, offering a way to retrieve information and visualize connections across a collection of triples. Oxygen XML Editor has proven helpful in formulating eXtensible Stylesheet Language (XSL) transformations to move metadata from a particular XML schema or format into RDF/XML or other serializations such as JSON-LD (JavaScript Object Notation for Linking Data).²⁰ Protégé is a tool developed by Stanford University that supports the OWL 2 Web Ontology Language and has helped to convert XML schemas to RDF ontologies and establish ways to express XML metadata in RDF. These tools provide the technical means to take metadata expressed in XML and physically reformat it to metadata expressed in an RDF serialization. What that reformatting also encompasses, however, is a review of the information expressed in XML and a set of decisions as to how to express that information as RDF statements.

Strategic approaches and ideas for handling data transformations into RDF have involved the XML schema or document type definition (DTD). These include Thuy, Lee, and Lee’s approach to map an XML schema (the XSD) to RDF, associating simpleType’s XSD in XML with properties in RDF, defining complexType’s XSD in XML as classes in RDF, and handling a hierarchy of XML schema elements with top levels as domains and lower-level elements and attributes as container classes or subproperties in those domains.²¹ Thuy et al. earlier worked on a method to transform XML to RDF by translating the DTD using RDFS (ELEMENTs in the DTD are RDF classes or subclasses, ATTLISTs are RDF properties, and ENTITIES—preset variables in the DTD—are called up for use in RDF as encountered).²² Similarly, Hacherouf, Bahloul, and Cruz translate an XML schema into an OWL ontology.²³ Klein et al. point out that while ontologies serve to describe a domain, XML schemas are meant to provide constraints on documents or structure for data so it can be advantageous to work out an RDF expression this way.²⁴ Tim Berners-Lee puts it simply: “the same RDF tree results from many XML trees,” meaning the same single statement in RDF (an article has an author Jane Smith) can be expressed in many ways in XML and can vary on the basis of the source of the XML, any schemas involved, and the people creating the metadata.²⁵ Transitioning from XML to RDF using the XML schema might serve to ensure all XML elements are

replicated in RDF but does not necessarily establish the relationships meant by that XML encoding without additional evaluation. There is no single strategy that will always work to move XML metadata into RDF, even within the same set of tools (such as Fedora/Hydra) or the same area of concern (libraries, archives, or museums).

USE CASES FOR RDF

The following use cases explain approaches to transition to RDF taken from two differing perspectives. The first set describes efforts to express XML schemas or standards as RDF ontologies. The second set describes efforts by various library or cultural-heritage digital collections to transform metadata records into RDF statements. They also show that strategies to transform XML to RDF cannot occur without a shift in view from structure to relationships and, likewise, from descriptive encoding to direct meaning.

Moving an XML Schema/Standard to an RDF Ontology

As a graduate student at Kent State University, Mixer took on converting the descriptive metadata standard VRA Core 4.0 from an XML schema to an RDF ontology.²⁶ Using the VRA Data Standards Committee Guidelines to ensure all minimum fields were included,²⁷ Mixer mapped VRA XML elements and attributes to schema.org, FOAF, VoID, and DC Terms ontologies. This process is known as “cherry-picking,” or combining various ontologies that already exist to represent properties or relationships (the predicates in RDF statements) as RDF instead of creating new proprietary RDF properties. Using OWL and RDFS as metavocabularies in Protégé, this created an ontology that could “retain the granularity required to describe library, archive, or museum items” of VRA Core 4.0’s design in XML without being a straight conversion of VRA Core 4.0 from XML to RDF.²⁸ The outcome was an XSLT stylesheet that was tested on VRA Core 4.0 XML records to produce that same information as RDF statements. One point that seemed to help in testing was the fact that all controlled vocabulary terms had reference identifiers in the XML (ready-made URIs). Something not discussed in the outcomes was that dates resulted in complex RDF (RDF statements that encompass additional RDF statements or blank nodes) and there was no discussion about this complexity or its effect on using those particular RDF statements.

VRA Core 4.0 now has an RDF ontology in draft form, with Mixer as one of its authors.²⁹ The OWL ontology still points to schema.org, FOAF, and VoID for equivalent classes and properties, but everything is now named within a VRA RDF ontology and namespace and translates to such when VRA Core 4.0 XML is transformed to RDF.

Another case in the category of going from an XML standard to an RDF ontology is the development of the BIBFRAME model for bibliographic description from the Library of Congress. The BIBFRAME model is expressed as RDF. According to the BIBFRAME site, “in addition to being a replacement for MARC, BIBFRAME serves as a general model for expressing and connecting bibliographic data.”³⁰ MARC has its own format of expression with numbered fields and subfields but can be expressed or serialized in XML and is often shared that way. The BIBFRAME model,

while revamping the way a bibliographic record is described on the basis of work, instance, authority, and annotation, also provides tools to transform records from MARC/XML to the RDF statements of BIBFRAME.³¹ A single namespace serves the BIBFRAME model and is explained as a long-term strategy to ensure namespace persistence over the next forty-plus years.³² The transformations produced from Library of Congress MARC records and local MARC records contain complex hierarchical RDF statements, particularly when ascribing authority sources to names, subjects, and types of identifiers. As it is still a work in progress there are no tools making use of BIBFRAME records in RDF.

An additional example is the work happening with PBCore, the public broadcasting metadata standard managed by the Corporation for Public Broadcasting.³³ Public broadcasting stations and other institutions across the United States provide descriptive, technical, and structural metadata for audiovisual materials using this XML standard. In Boston, WGBH's use of PBCore coincides with its digital asset management system, HydraDAM, built on Fedora 3 and the Hydra technology stack (based on Blacklight, Solr, and the Fedora Digital Repository).³⁴ Fedora 3 does not natively support RDF statements as properties on objects like Fedora 4. Building off an interest to move HydraDAM to Fedora 4 and leverage RDF for metadata about audiovisual collections, WGBH began exploring transitioning the PBCore XML metadata standard into an RDF ontology. EBUCore, the European Broadcasting Union's metadata standard, is already expressed as an RDF ontology.³⁵ A comparison between the XML standard of PBCore and the classes and properties expressed in EBUCore revealed that most PBCore elements were covered by the EBUCore ontology.³⁶ Efforts are ongoing to offer PBCore 3.0 as an RDF ontology that uses EBUCore with the addition of a smaller set of properties along with a way to transform PBCore XML to PBCore 3.0 in RDF.³⁷

The Hydra community, in an effort to help the transition from Fedora 3 with its XML binary files of descriptive metadata to Fedora 4 using RDF statements as properties on objects, is working on a recommendation and transformation to move descriptive metadata in MODS XML into RDF that is usable in Fedora 4.³⁸ The MODS standard has a draft of an RDF ontology and a stylesheet transformation available,³⁹ but the complex hierarchical RDF produced from this transformation is unmanageable with the current Fedora 4 architecture. The Hydra MODS and RDF Descriptive Metadata Subgroup is attempting to reflect the MODS elements in simple RDF statements that can be incorporated as properties on a Fedora 4 digital object.⁴⁰ Led by Steven Anderson at the Boston Public Library, this group is moving through MODS element by element, asking the question, "If you had to express this MODS element from your metadata in RDF today, how would you do that?" Participating institutions are reviewing their MODS records and exploring the possible RDF predicates that could be used to represent the meaning of that information. Some are even considering how to construct those RDF statements so that MODS XML can be re-created as close to the original MODS as possible (this is called "round tripping"). There are still questions as to whether every single MODS element will be reflected in this transformation, how exactly Fedora 4 will make use of these descriptive RDF statements, and if the original MODS XML will need to be preserved as part of the digital object in Fedora, but this group is recognizing that moving from

Fedora 3 to Fedora 4 requires a major shift in thinking about descriptive metadata. This transformation tool is an effort to help make that transition possible.

The Avalon Media System is an open source system for managing and providing access to large collections of digital audio and video.⁴¹ It is built on Fedora 3 and the Hydra technology stack and uses MODS XML to store descriptive metadata. As development progresses and the available descriptive fields expand, maintaining the workflow to update XML records in Fedora and reindexing objects in the Hydra interface becomes increasingly complicated. Each time an update is made to descriptive information about an audiovisual item through the Avalon interface, the entire XML record for that object, stored as a binary text file, is rewritten in Fedora 3 and reindexed in Solr. In considering advantages to using Fedora 4, it appears that descriptive metadata properties stored in RDF are easier to manage programmatically (updating content, adding new fields, more focused reindexing) because descriptive information would not be stored in a single binary file but as individual properties on the object.

Turning XML metadata into RDF or Linked Data for publishing, search and discovery, and management

As Southwick describes the process, the library at the University of Nevada Las Vegas (UNLV) took a collection with descriptive records from CONTENTdm and published them as a single RDF Linked Open Data set.⁴² After cleaning up controlled vocabulary terms across collections and solidifying locally controlled vocabularies, they exported tab-delimited CSV records from CONTENTdm. These records were brought into OpenRefine with its RDF extension where they reviewed the data and mapped to various properties within the Europeana Data Model (EDM). Controlled vocabulary terms were in text form and had to be reconciled against a SPARQL endpoint, either locally from downloaded data or from the controlled vocabulary service, to gather the URIs to use as the object or value in the RDF statement. OpenRefine was then used to create RDF files that were uploaded to a triplestore (first Mulgara then OpenLink Virtuoso). This provided public access to the Linked Open Data set and a SPARQL endpoint for querying the data set. After publishing the data set they experimented with PivotViewer from OpenLink Virtuoso and RelFinder to see what kinds of connections and relationships could be visualized from the data as Linked Open Data. The outlined steps are clear and the outcomes are described, but interestingly the data set itself no longer appears to be available online.⁴³ Although the UNLV use case relies on CSV instead of XML as the data source, the tools and workflows enlisted to transform the data set into RDF Linked Open Data are still applicable. OpenRefine can import XML just as it imports CSV, so this described case shows the tools that can be used and decisions to be made in processing that data into RDF statements.

In Oregon Digital,⁴⁴ XML from Qualified Dublin Core, VRA Core, and MODS at two different institutions (University of Oregon and Oregon State University) were mapped as Linked Open Data and stored in a triplestore to be served up in a new web application using the Hydra technology stack.⁴⁵ An inventory of metadata fields across all collections was first mapped to existing Linked

Data terms, or properties (those with available URIs), then properties that were needed in the new web application but did not have available corresponding URIs were mapped to a newly devised local namespace for Oregon Digital. Any properties that were not used were kept in the original static XML file for the record as part of the digital object in Fedora. The focus here appears to be on mapping properties without as much detail provided on whether the objects were kept as text or mapped to URI values where possible. From the sample record provided the objects appear to be text and not URIs. The real power of this project is finding common properties to describe objects from diverse collections and institutions. What also comes out in the example mappings is the use of many different namespaces or ontologies (DC Terms, MARC Relators, but also MODS and MADS that produce complex RDF).

The University of Alberta also combined a variety of XML metadata from different sources into a new digital asset management system based on Fedora 4 and the Hydra technology stack, called the Education and Research Archive.⁴⁶ Reporting on the experience at Open Repositories 2015, Farnel described the process as working in phases.⁴⁷ Beginning with item types, languages, and licenses, then moving to place names and controlled subject terms, and finally person names and free-form subjects, they made multiple passes converting XML metadata into RDF statements and incorporating URIs whenever possible. They are combining all of this into a single data dictionary,⁴⁸ making use of several RDF ontologies to cover the various metadata properties that are being described about objects and collections.

University of California at San Diego (UCSD) has developed a local data model using a mix of external (MADS, VRA Core, Darwin Core, PREMIS) and local ontologies. They published a data dictionary and are working on a substantially different revision as part of the metadata workflow they use to bring digital objects into their digital asset management system from a variety of source metadata formats including XML.⁴⁹ This allows metadata to be created from disparate source formats and makes it possible to bring them together as RDF for delivery, management, and preservation.

DISCUSSION

If metadata is in XML form and the desire is to express it as RDF, this is not merely a transformation from one XML schema to another. It is changing the expression of that data and changing its use. Having metadata in XML means information is encoded in a specific way that allows for interchange and sharing. Having metadata in RDF is making statements that have direct meaning and can be used independently.

There are different perspectives involved in metadata when approaching RDF: those that manage metadata standards (the XML standard side) and those that have metadata encoded using those XML standards (the data management side). Depending on the desired outcomes, the needs of these two perspectives can conflict. When managing a metadata standard the RDF transition tends to follow certain patterns:

-
- Transform an XML standard into a new RDF ontology
 - Examples: Dublin Core (DC), Darwin Core (DWC), MODS, VRA Core
 - Establish a move to RDF that incorporates another existing ontology
 - Example: PBCore, Hydra community

From the data management side, the RDF transition means different patterns occur. These scenarios often start by reviewing the needed outcome, deciding how much metadata needs to be expressed in RDF, and what works best to get the metadata to that point. Cases include the following commonalities:

- Creating new search and discovery end user applications
 - Example: Oregon Digital, University of Alberta
- Publishing Linked Data sets
 - Example: UNLV, University of Alberta
- Managing metadata using software that supports RDF
 - Example: University of Alberta, UCSD, Hydra community

Conflicts are occurring when the needed outcome on the data management side is not supported by the RDF ontology transitions that have occurred for the XML standards being used. An example of this is how RDF is handled in Fedora 4. When RDF is complex (the object of one statement is another entire RDF statement), Fedora produces blank nodes as new objects within the repository. While not technically problematic, descriptive metadata with complex RDF can result in a situation where a digital object ends up referencing a blank node that then points to, for example, a subject or a genre. This subject or genre has been created as its own object within the digital repository even though that subject or genre is only meant to provide meaning for the digital object. MODS RDF produces this complexity and thus is not workable to use with Fedora 4. In contrast, other standards such as DC or DWC in RDF produce simple statements that Fedora 4 can apply to a digital object without any additional processing.

Complications in transitioning from XML to RDF also occur when the original XML does not include URIs or authority-controlled sources. Converting this metadata to RDF can mean locally minting URIs or bringing data over as literals (strings of text) without using URIs at all. Ideally, the result is somewhere in the middle with externally controlled vocabularies incorporated as much as possible and literals or locally minted URIs only used where absolutely necessary. Translating strings to authoritative sources is intensive work. If the XML standard cannot be expressed as a single RDF ontology, work is further complicated by the need to map XML elements to different RDF ontologies using logic that is often decided locally.

While it is possible to transition XML to RDF, the process is not uniform and the pathway involves a lot of labor. Potential alleviators for this labor might involve a more user-centered approach by XML standard bodies to consider the ways their standards can be used when translated into RDF (“users” in this context meaning the users of the standards, not the end users searching and

discovering digital content). Triplestores can manage queries for complex RDF, but digital repository systems are not there yet. Those that support RDF for description of objects do so on the basis of simple property statements. A complex RDF ontology is going to be a challenge to support over time.

Another way to progress forward is for the data management side of the equation to focus efforts on showing, in an end user search and discovery format, what is currently possible when XML is transitioned into RDF. Published Linked Data sets need to have interfaces for access and use, showing the value of what is currently available and any needs or gaps that remain. Libraries and cultural-heritage organization engaged in this work should also openly share the processes that both work and do not work so others contemplating this transformation can consider how to forge ahead themselves. Libraries and cultural-heritage organizations moving metadata from XML to RDF should provide feedback to XML standard bodies regarding the usefulness or complications of any RDF transitional help an XML standard might provide.

Technologies for incorporating RDF into web applications and truly connecting triples across the web also require further work. Triplestores have so far been the main way to expose data sets but have not been incorporated into common library or cultural-heritage end user search and discovery web applications. Additionally, triplestore use does not seem to extend to management or long-term storage of complete data about digital objects. There seems to be a decision to either reduce the data stored in a triplestore down to simple statements or use the triplestore more like an isolated index or SPARQL endpoint only and manage the complete metadata record separately (in a static file containing text or in a separate database). That aligns triples in RDF more with relational database storage than with catalog records. Triple statements focus on relationships and not the complete unique details of the thing being described. Triplestores can handle complex hierarchical RDF graphs and provide responses on the basis of queries against those complexities,⁵⁰ but triplestores do not appear to be taking over as either the main search and discovery mechanism for online digital resources or for digital object management. Software using RDF natively is also not currently widespread. A project such as the BIBFRAME Initiative that plans to incorporate RDF needs to make sure the complexity of its data model in RDF is manageable by any tools it produces and that it is possible for vendors and suppliers to encompass the data model in their software development.

CONCLUSION

The reasons for deciding metadata should transition to RDF are just as important as determining the best process for implementing that transition. Reasons for transitioning to RDF are conceptually based around making data more easily shareable and setting up data to have meaning and relationships as opposed to local static description that requires programmatic interpretation. The use cases outlined in this article show the reality does not quite yet match the concept. Transitioning an XML standard to RDF does not make that data more shareable or more easily understood unless there are end user applications for using that data in RDF. Publishing

Linked Data involves going through transitional steps, but the endpoint seems to be more of a byproduct. The real goal is going through the process of producing Linked Data to learn how that works. Self-contained projects that aim to express collections in RDF for the purpose of a new search and discovery interface are more successful in implementing RDF that has that new level of meaning and relationship. Beyond the borders of these projects, however, the data is not being shared or used.

The use cases described above show some examples of what is happening now when transitioning from XML to RDF. Approaches include XML standards converting to RDF expression as well as digital collections with metadata in XML that have an interest in producing that metadata as RDF. Software that incorporates RDF is still developing and maturing. Helping that process along by providing a pathway from XML to functionally usable RDF improves the chances of the Semantic Web becoming a real and useful thing. It is vital to understand that transitioning from XML to RDF requires a shift in perspective from replicating structures in XML to defining meaningful relationships in RDF. Metadata work is never easy, and for metadata to move from encoded strings of text to statements with semantic relationships requires coordination and communication. How best to achieve this coordination and communication is a topic worth engaging as the move to use RDF, produce Linked Data, and approach the Semantic Web continues.

BIBLIOGRAPHY

- Berners-Lee, Tim. "Linked Data." *Linked Data - Design Issues*, June 18, 2009.
<http://www.w3.org/DesignIssues/LinkedData.html>.
- . "Why RDF Model Is Different from the XML Model." *Semantic Web*, September 1998.
<http://www.w3.org/DesignIssues/RDF-XML.html>.
- Estlund, Karen, and Tom Johnson. "Link It or Don't Use It: Transitioning Metadata to Linked Data in Hydra," July 2013. <http://ir.library.oregonstate.edu/xmlui/handle/1957/44856>.
- Farnel, Sharon. "Metadata at a Crossroads: Shifting 'from Strings to Things' for Hydra North." Slideshow presented at the Open Repositories, Indianapolis, Indiana, 2015.
<http://slideplayer.com/slide/5384520/>.
- Hacherouf, Mokhtaria, Safia Nait Bahloul, and Christophe Cruz. "Transforming XML Documents to OWL Ontologies: A Survey." *Journal of Information Science* 41, no. 2 (April 1, 2015): 242–59.
doi:[10.1177/0165551514565972](https://doi.org/10.1177/0165551514565972).
- Klein, Michel, Dieter Fensel, Frank van Harmelen, and Ian Horrocks. "The Relation between Ontologies and XML Schemas." In *Linköping Electronic Articles in Computer and Information Science*, 2001. doi:[10.1.1.14.1037](https://doi.org/10.1.1.14.1037).
- Lassila, Ora. "Introduction to RDF Metadata." W3C, November 13, 1997.
<http://www.w3.org/TR/NOTE-rdf-simple-intro-971113.html>.

-
- Manola, Frank, and Eric Miller. "RDF Primer 1.0, Section 2.3 Structured Property Values and Blank Nodes." *W3C Recommendation*, February 10, 2004. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#structuredproperties>.
- Mixer, Jeff. "Using a Common Model: Mapping VRA Core 4.0 Into an RDF Ontology." *Journal of Library Metadata* 14, no. 1 (January 2014): 1–23. [10.1080/19386389.2014.891890](https://doi.org/10.1080/19386389.2014.891890).
- Poupeau, Gautier. "XML vs RDF: logique structurelle contre logique des données (XML vs RDF: structural logic against logic data)." *Les Petites Cases*, August 29, 2010. <http://www.lespetitescases.net/xml-vs-rdf>.
- "RDF and TEI XML," October 13, 2010. <https://listserv.brown.edu/archives/cgi-bin/wa?A2=ind1010&L=TEI-L&D=0&P=28928>.
- Southwick, Silvia B. "A Guide for Transforming Digital Collections Metadata into Linked Data Using Open Source Technologies." *Journal of Library Metadata* 15, no. 1 (March 2015): 1–35. doi: [10.1080/19386389.2015.1007009](https://doi.org/10.1080/19386389.2015.1007009).
- Thuy, Pham Thi Thu, Young-Koo Lee, and Sungyoung Lee. "A Semantic Approach for Transforming XML Data into RDF Ontology." *Wireless Personal Communications* 73, no. 4 (2013): 1387–1402. doi: [10.1007/s11277-013-1256-z](https://doi.org/10.1007/s11277-013-1256-z).
- Thuy, Pham Thi Thu, Young-Koo Lee, Sungyoung Lee, and Byeong-Soo Jeong. "Transforming Valid XML Documents into RDF via RDF Schema." In *Next Generation Web Services Practices, International Conference on*, 0:35–40. Los Alamitos, CA: IEEE Computer Society, 2007. doi: [10.1109/NWESP.2007.23](https://doi.org/10.1109/NWESP.2007.23).
- "XML RDF." W3Schools. Accessed September 30, 2015. http://www.w3schools.com/xml/xml_rdf.asp.
- Yee, Martha M. "Can Bibliographic Data Be Put Directly onto the Semantic Web?" *Information Technology and Libraries* 28, no. 2 (March 1, 2013): 55–80. doi: [10.6017/ital.v28i2.3175](https://doi.org/10.6017/ital.v28i2.3175).

NOTES

1. "Darwin Core," Darwin Core Task Group, Biodiversity Information Standards, last modified May 5, 2015, <http://rs.tdwg.org/dwc/>.
2. "Metadata specifications," European Broadcasting Union, <https://tech.ebu.ch/MetadataEbuCore>.
3. Gautier Poupeau, "XML vs RDF: logique structurelle contre logique des données (XML vs RDF: structural logic against logic data)," *Les Petites Cases* (blog), August 29, 2010, <http://www.lespetitescases.net/xml-vs-rdf>.
4. Ora Lassila, "Introduction to RDF Metadata," W3C, November 13, 1997, <http://www.w3.org/TR/NOTE-rdf-simple-intro-971113.html>.

-
5. "XML RDF," W3Schools, accessed September 30, 2015, http://www.w3schools.com/xml/xml_rdf.asp.
 6. See "Serialization formats" from Resource Description Framework on Wikipedia. "Resource Description Framework," Wikipedia, March 18, 2016, https://en.wikipedia.org/wiki/Resource_Description_Framework#Serialization_formats.
 7. "RDF and TEI XML," email thread on TEI-L@listserv.brown.edu, October 13–18, 2010, <https://listserv.brown.edu/archives/cgi-bin/wa?A2=ind1010&L=TEI-L&D=0&P=28928>.
 8. Martha M. Yee, "Can Bibliographic Data Be Put Directly onto the Semantic Web?" *Information Technology & Libraries* 28, no. 2 (March 1, 2013): 57, doi:[10.6017/ital.v28i2.3175](https://doi.org/10.6017/ital.v28i2.3175).
 9. Frank Manola and Eric Miller, "RDF Primer 1.0, Section 2.3 Structured Property Values and Blank Nodes," W3C Recommendation, February 10, 2004, <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#structuredproperties>.
 10. Sharon Farnel, "Metadata at a Crossroads: Shifting 'from Strings to Things' for Hydra North" (slideshow presentation, Open Repositories, Indianapolis, Indiana, 2015), <http://slideplayer.com/slide/5384520/>.
 11. <http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/>.
 12. <https://jena.apache.org/documentation/fuseki2/>.
 13. <http://rdf4j.org>.
 14. <http://www.oxygenxml.com>.
 15. <http://protege.stanford.edu>.
 16. <http://openrefine.org>.
 17. <https://en.wikipedia.org/wiki/SPARQL>.
 18. <http://refine.deri.ie>.
 19. <https://jena.apache.org/tutorials/sparql.html>.
 20. <http://json-ld.org>.
 21. Pham Thi Thu Thuy, Young-Koo Lee, and Sungyoung Lee, "A Semantic Approach for Transforming XML Data into RDF Ontology," *Wireless Personal Communications* 73, no. 4 (2013): 1392–95, doi:[10.1007/s11277-013-1256-z](https://doi.org/10.1007/s11277-013-1256-z).
 22. Pham Thi Thu Thuy et al., "Transforming Valid XML Documents into RDF via RDF Schema," in *Next Generation Web Services Practices, International Conference on*, vol. 0 (Los Alamitos, CA: IEEE Computer Society, 2007), 37, doi:[10.1109/NWESP.2007.23](https://doi.org/10.1109/NWESP.2007.23).

-
23. See Mokhtaria Hacherouf, Safia Nait Bahloul, and Christophe Cruz, "Transforming XML Documents to OWL Ontologies: A Survey," *Journal of Information Science* 41, no. 2 (April 1, 2015): 242–59, doi:[10.1177/0165551514565972](https://doi.org/10.1177/0165551514565972).
 24. Michel Klein et al., "The Relation between Ontologies and XML Schemas," section 5 in *Linköping Electronic Articles in Computer and Information Science*, 6 (2001), doi:[10.1.1.108.7190](https://doi.org/10.1.1.108.7190).
 25. Tim Berners-Lee, "Why RDF Model Is Different from the XML Model," Semantic Web Road map, September 1998, <http://www.w3.org/DesignIssues/RDF-XML.html>.
 26. See Jeff Mixter, "Using a Common Model: Mapping VRA Core 4.0 Into an RDF Ontology," *Journal of Library Metadata* 14, no. 1 (January 2014): 1–23, doi:[10.1080/19386389.2014.891890](https://doi.org/10.1080/19386389.2014.891890).
 27. The document currently labeled "How to Convert Version 3.0 to Version 4.0" contains a recommendation for a minimum set of elements for "meaningful retrieval" in VRA Core: http://www.loc.gov/standards/vracore/convert_v3-v4.pdf.
 28. Mixter, "Using a Common Model," 2.
 29. "VRA Core RDF Ontology Available for Review," Visual Resources Association, October 7, 2015, <http://vraweb.org/vra-core-rdf-ontology-available-for-review/>.
 30. "Bibliographic Framework Initiative," Library of Congress, <https://www.loc.gov/bibframe/>.
 31. See "MARC to BIBFRAME transformation tools" at "Tools" BIBFRAME, <http://bibframe.org/tools/>.
 32. "Why a single namespace for the BIBFRAME vocabulary?" Library of Congress, BIBFRAME Frequently Asked Questions, <https://www.loc.gov/bibframe/faqs/#q06>.
 33. "PBCore 2.1," Public Broadcasting Metadata Dictionary Project, <http://pbcore.org>.
 34. "WGBH," Hydra Community Partners, <http://projecthydra.org/community-2-2/partners-and-more/wgbh/>.
 35. "Metadata specifications," European Broadcasting Union, <https://tech.ebu.ch/MetadataEbuCore>.
 36. See notes from PBCore Hackathon Part 2, which occurred in June 2015 showing an element-by-element analysis of PBCore against EBUCore. "PBCore Hackathon Part 2," June 15, 2015, <https://docs.google.com/document/d/1pWDfYIzHpjCn5RWJ1fioweXg5RIrXuDxCWkBQ5BMIA/>.
 37. "Join us for the PBCore Sub-Committee Meeting at AMIA!" Public Broadcasting Metadata Dictionary Project Blog, November 11, 2015, <http://pbcore.org/join-us-for-the-pbcore-sub-committee-meeting-at-amia/>.

-
38. "MODS and RDF Descriptive Metadata Subgroup," last modified March 19, 2016, <https://wiki.duraspace.org/display/hydra/MODS+and+RDF+Descriptive+Metadata+Subgroup>
 39. "MODS RDF Ontology," Library of Congress, <https://www.loc.gov/standards/mods/modsrdf/>.
 40. "MODS and RDF Descriptive Metadata Subgroup," last modified March 19, 2016, <https://wiki.duraspace.org/display/hydra/MODS+and+RDF+Descriptive+Metadata+Subgroup>
 41. "Avalon Media System," <http://www.avalonmediasystem.org>.
 42. See Silvia B. Southwick, "A Guide for Transforming Digital Collections Metadata into Linked Data Using Open Source Technologies," *Journal of Library Metadata* 15, no. 1 (March 2015): 1–35, <http://dx.doi.org/10.1080/19386389.2015.1007009>.
 43. The URL for information is a blog with no links to a data set (<https://www.library.unlv.edu/linked-data>) and the collection site seems to still be based on CONTENTdm (<http://digital.library.unlv.edu/collections>).
 44. "Oregon Digital," <http://oregondigital.org>.
 45. See Karen Estlund and Tom Johnson, "Link It or Don't Use It: Transitioning Metadata to Linked Data in Hydra," July 2013, <http://ir.library.oregonstate.edu/xmlui/handle/1957/44856>, accessed from ScholarsArchive@OSU.
 46. "ERA: Education & Research Archive," <https://era.library.ualberta.ca>.
 47. Farnel, "Metadata at a Crossroads."
 48. https://docs.google.com/spreadsheets/d/1hSd6kf4ABm-m8VtYNYqfjGtiZG7bLJQ3fWRbF_nVoIw/edit#gid=1362636241.
 49. The substantially revised data model is not available online yet, but the following shows some of the progress toward an RDF data model: "Overview of DAMs Metadata Workflow," UC San Diego, May 21, 2014, <https://tpot.ucsd.edu/metadata-services/mas/data-workflow.html>; "DAMS4 Data Dictionary," <https://htmlpreview.github.io/?https://github.com/ucsdlib/dams/master/ontology/docs/data-dictionary.html>, retrieved from GitHub.
 50. See the Apache Jena SPARQL Tutorial for an example of complex RDF with sample queries against that complexity. "SPARQL Tutorial - Data Formats," The Apache Software Foundation, https://jena.apache.org/tutorials/sparql_data.html.