Journal of Library Metadata, 13:290–309, 2013
Published with license by Taylor & Francis
ISSN: 1938-6389 print / 1937-5034 online
DOI: 10.1080/19386389.2013.828551

Routledge
Taylor & Francis Group

Moving Our Data to the Semantic Web: Leveraging a Content Management System to Create the Linked Open Library

KERI THOMPSON and JOEL RICHARD
Smithsonian Libraries, Smithsonian Institution, Washington, DC, USA

While migrating their website and digital library content to the Drupal content management system, the Smithsonian Libraries saw an opportunity to not only improve the management and presentation of their content, but also to make it available for reuse by its own site and others by publishing it as linked open data[1] (LOD). Leveraging the core functionality of Drupal 7 to produce RDF, it embarked on two projects. The first will publish bibliographic data taken from the library catalog as part of its digitization program and present it as RDFa. The second will create LOD from a much-cited botanical reference work.

KEYWORDS Drupal, libraries, linked data, linked open data, Semantic Web

The Smithsonian Libraries is the world's largest museum library system, with 22 physical locations, more than 1.9 million collection items, and a website that receives more than 1.2 million unique visitors per year. Its primary focus is supporting the research of the scientists, curators, and museum specialists who work at the Smithsonian, whose fields of study range from anthropology and American art to the history of technology and zoology. The system has, however, always had a robust Web presence that targets a much broader audience and seeks to make its collections available to researchers

This paper assumes that the reader will have some familiarity with the concepts of linked data as described by Heath and Bizer (2011) and a general awareness of the terminology and concepts behind RDF, triplestores, and RDFa (RDF in attributes), as described by Manola and Miller (2004).

Address correspondence to Keri Thompson, Smithsonian Libraries, P.O. Box 37012, MRC 154, Washington, DC, 20013-7012, USA. E-mail: thompsonk@si.edu

outside the institution—hobbyists and collectors, educators, and the general public.

Though the Smithsonian Libraries has had a Web presence since 1996, and has been digitizing collections and making them available online since 1998, both digitization and Web development were being done without the benefit of either a content management system or a collections management system. When the pace of digitization increased twenty-fold in 2008, we realized that we could no longer manage without one or both. After careful consideration, in 2011 we began to migrate the Libraries' extensive Web site to the Drupal 7 content management system. Our aim was not only to better manage the content and presentation of the website, but also to develop a framework that would enable us to publish and manage the nearly 4,000 digital library objects that were stored in a variety of systems. We chose Drupal 7 as our content management system for a number of reasons, including its flexibility, open development model, and increasing adoption by cultural heritage organizations. The decision to experiment with publishing linked open data (LOD), however, was catalyzed by the fact that Drupal 7 core includes native functionality to publish linked data.

DRUPAL AND LINKED DATA

Drupal support for linked data began in 2008 with the release of the RDF CCK (Content Construction Kit) module for Drupal 6 (Corlosquet, Delbru, Clark, Polleres, & Decker, 2009). When Drupal 7 was released, some of the functionality that was developed to work with RDF in Drupal 6 was integrated into the Drupal 7 core, particularly the ability to publish RDFa 1.0. Since then, numerous modules and extensions to handle specific functions with data modeled in RDF have been developed, including modules that enable publication of RDFa 1.1 Lite, which is more fully supported by the major search engines and Facebook.[2] In addition to RDFa, Drupal 7 has functionality that extends core RDF support by providing extra APIs and the ability to publish in additional serialization formats such as RDF/XML, NTuples, or Turtle. The data that is published in RDF is not, however, stored in triples, but in a traditional database. The data is then output in the serialization of your choice, using the vocabulary mappings you specify. Drupal also has the ability (via a module) to act as a SPARQL end-point,[3] so others can query and reuse data, and the RDF proxy module, which enables you to connect your data to other RDF data sources and keep them in synch. For more background on Drupal's implementation of RDF and for generic-use cases and tutorials, the Slideshare presentation "How to Build Linked Data Sites with Drupal and RDFa" (Corlosquet, Clark, & Passant, 2010) is recommended, as is the two-part overview "Semantic Web, Linked Data and Drupal" (Clark, 2011; Corlosquet & Clark, 2011).

Two Projects, One Platform

The opportunity to model and publish our digital collections, particularly our digitized texts, in Drupal 7 prompted us to think not only of what functionality we had been lacking and how to create it, but to think ahead to what functionalities we would want in the future. We had long wished for the ability to more easily reuse our own data, and be able to enrich our data programmatically by connecting it to other data sources in the Smithsonian and on the Web. LOD offered the promise of a method to achieve both goals, and our digitized texts provided what we felt was a perfectly sized data set with which to do a LOD pilot project. This test data set included descriptive metadata for the 4,000 digital objects and links to the objects themselves.

In addition to our digitized texts, we had another project that we quickly realized would also benefit from being published as LOD. This project, Taxonomic Literature II online (TL-2), is comprised of the digitization and online presentation of a standard reference work published by the International Association for Plant Taxonomy. TL-2 is a guide to the literature of systematic botany published between 1735 and 1940, which provides biographical and bibliographic data, including suggested standard abbreviations for authors' names and short titles that can be used by plant taxonomists when citing publications or species. By giving the name abbreviations and bibliographic numbers stable URIs and publishing them using RDF, we would be able to give the plant taxonomy community an additional way to access and reuse this valuable data.

Though the methodology for realizing each project differed slightly, the goals and guiding principles for implementation were the same for both. Our initial goals would be to produce at least 4-star linked data[4](Berners-Lee, 2010), following best practices that include reusing common vocabularies as much as possible, making all data available with an open license, and clearly publishing data rights along with the data (Linked Data Cookbook, 2011). Because the most difficult and time consuming part of creating 5-star linked data is actually creating the links, we are leaving the fifth star for the second phase of our projects. We will, however, discuss our approaches and the tools and methods we are exploring to create links.

In each project the basic steps would be the same:

1. create a data model, including URIs
2. migrate existing data to Drupal
3. choose ontologies and assign vocabulary terms to data elements
4. publish the data
5. create links to external sources (phase two)

Below, we will walk through the process for publishing linked open data for both projects. In the first section, we will discuss how we extracted MARC and non-MARC metadata from our Integrated Library System (ILS),

mapped that metadata to RDF vocabularies, and published it as LOD for the "Online Books" section of our Digital Library. In the second section of this paper, we will discuss the process we used to publish data from TL-2 as LOD on our website. Difficulties that arose, and lessons learned from both projects will be covered at the end of the paper.

LOD FOR ONLINE BOOKS

The Smithsonian Libraries has been digitizing books and putting them online for 15 years, starting with hand-linked HTML pages and PDFs and progressing to databased page images and structured files stored on the Internet Archive. Historically, each digital text object was displayed with its own unique features, navigation, and styling. When the decision was made to migrate our website to a content management system, it gave us the opportunity to create a dedicated space on the site to serve up all our digitized books in one consistent interface. Extending existing tools and data created as part of the digitization process, we have developed a way to display and index the collections by periodically ingesting and reusing descriptive metadata from our ILS, combined with additional descriptive and structural metadata, to make our full-text collections available on the website in a useful way.

Online Books: Making the Data Sausage

Our data model for presentation of LOD in the digital library is heavily dependent on our existing digitization process. In order to manage our digitization workflow and create all the necessary files to make a functional online book, we have developed two tools: a workflow database and Macaw,[5] a locally developed Web-based tool for creating page-level descriptive and item-level structural data. The workflow database contains item-level data about a scanned object, such as barcode, location, and volume and issue numbers, which are periodically extracted from our ILS and then loaded into a separate database. The creation of this "shadow catalog" was necessary because, unlike title-level data that we can extract via Z39.50, we are not able to easily harvest and reuse item level data from our catalog. The workflow database also contains administrative metadata entered by staff such as copyright information, the date it was scanned, and identifiers for the item at Internet Archive. All of our digitized books at this time are sent to Internet Archive, which serves as free storage, a publicly accessible repository, and a staging area.

Our second tool, Macaw, is used to create the necessary structural data for publishing a scanned book online; it also collects and packages all the descriptive, administrative, and structural data and then submits that package

to the Internet Archive. Macaw queries the workflow database for item-level data and harvests the corresponding MARC bibliographic record for that item from the ILS using Z39.50. Macaw then transforms that MARC record to MARCXML and creates additional XML files from the structural and administrative data for submission to the Internet Archive. Internet Archive reuses the item and title-level descriptive data, including the MARCXML record, for display and indexing in archive.org and in Open Library.[6] The structural data are used to create the "page turning" book display.

To ingest the books data created for Internet Archive into Drupal, we initially planned to use existing modules such as the Feeds Importer. However, it quickly became clear that existing tools would not be suitable given the nature of our data, the fact that it is stored in multiple places and because existing tools simply took too long to import. We also did not want to rely too heavily on library-specific protocols, such as Z39.50, in order to make our importer reusable for projects with data originating outside our ILS. Our alternative to using the Feeds Importer and Z39.50 was a two-step process: first, create a command-line PHP script to preprocess and collect the data into an XML import file that would then be passed to Drupal via the Web interface. In the second step, Drupal would receive the XML file through a custom-built "SIL Books Import" module.

The "SIL Books Import" module harvests the metadata files staged at Internet Archive and uses a Z39.50 query to the ILS to capture any updated descriptive data added since the time of scanning. Once the data are collected, the module also harvests a cover thumbnail for each item, again from the Internet Archive. After having harvested all the required files, a script combines all of the data necessary to identify vocabulary terms, series/serial relationships and volumes, sample cover images, and all metadata to be displayed on our Online Books into an XML file that can be handed off to Drupal for quick import.

The actual import process was crafted to not overwrite existing information, which means that the one XML file can always contain all of the volumes that are meant to be online. The file can be imported multiple times without harm to the website. Because digitization of our collections is ongoing and we would be periodically updating the collection of digitized texts in our Digital Library, we included a separate routine to update existing records from the XML file. Various other small routines were created on an ad-hoc basis to correct errors or false assumptions made during the early stages of development. For example, the original way we found and extracted cover image thumbnails was a multi-step process, which we were later able to streamline when we discovered a more efficient way to download single images. In general, most of these early assumptions were predicated on our knowledge of Internet Archive and/or how the ILS delivered data through Z39.50 and affected the creation of the XML file rather than the import process. Most of the corrections could be made in the PHP script followed

by an iteration of the update routine, which did not impact data that had already been imported.

Online Books: The Data Model

The Online Books section of the digital library is not meant to replicate the functionality of the ILS. It is also not the repository of record for our collections metadata—that remains our ILS. As such, it contains only a subset of the bibliographic information we have for each item sufficient to enable common functions including searching, citing, and indexing by search engines. It is this basic data that we are making available using RDFa on each page.

The data model we developed for the Online Books is heavily informed by our digitization practices. Each discrete "item" being digitized is (typically, but not always) defined by its physical extent at the time of scanning. For example, one monographic volume is one item, as is one bound journal volume, regardless of how many intellectual volumes, issues, and so on, that a bound journal volume may contain. Unlike the relational model of item and title records in the ILS, each item record can stand alone as a basic (but not full) bibliographic record and contains data from the MARC title level record, including data from the 1XX, 245, 260, and 6XX fields as well as the OCLC number and unique record number ("bib number") from the ILS that are used to create links to the original title data. Item records also contain "piece" specific information such as barcode, volume and issue number, or publication date ranges. In our data model, though we focus on the "item," we still need Web pages ("nodes" in Drupal) and, therefore, records that collocate serials and series items for browsing. These serial parent records, which exist only in Drupal, include the same descriptive data as the item records (for display and indexing) as well as linking IDs for all the serial "children" and links to preceding and succeeding titles taken from the MARC 780 and 785. On the surface, this approach may seem inefficient, compared to a proper relational data model, particularly given that the majority of our data are originating in a relational database, the ILS, with very different structures. A data model that focuses on the "item" and that duplicates data so that each item record can effectively stand alone as a descriptive record has two advantages. Practically, it makes importing and exporting data from our various systems more straightforward, which happens regularly, as digitization is ongoing. It also works better with Drupal's general data model, which assumes each "node" or record contains one complete intellectual object.

The item metadata are stored in a Drupal node with the content type "book." In Drupal, similar nodes are kept together in a "content type," which is logically equivalent to a single database table. Besides the book content type, we also created a separate content type for people or corporate

bodies that we called "author." This was done for several reasons, including the assumption that in the future, after creating links to biographical data sources, we would be able to create descriptive records for people or institutions that were richer than those found in our ILS. The author content type could also be reused in other data sets published in the Digital Library, such as for botanists in TL-2 or for companies listed in our inventory of trade catalogs. We assume that in the future, when we are able to create or import additional issue, article, or chapter metadata, we will create additional content types and relate them to the appropriate book and author nodes.

To create a simple, citation-like record from which data could be easily linked and reused, some of the data that we took from our ILS was modified. For example, Library of Congress Subject Headings (LCSH) stored in MARC 650 were deconstructed by subfield, so 650 $z was stored in a field for geographic data, and 650 $y was stored in a separate field for chronological data. Similarly, publisher information in MARC 260 was also broken apart by subfield. This will, we hope, make it easier down the road to connect to linked data sources for geographic and temporal data, respectively.

Part of creating a data model that will work for linked data is designing a usable and stable URI[7] system for each digital object. When constructing URIs, it is common to include data that helps place the content in context, such as dates or years (common in most blog software) or nesting content into directories that mirror organizational structure, or include file extensions (.php, .html.). Creating good URIs for linked data, however, involves ensuring that the URIs do not contain additional contextual information that may change if the resource is moved to a different platform or technology. We did, however, want to have some semantic indicators within the URI string, simply because it is helpful for humans. Most of our digital books are also stored at the Internet Archive, which uses the unique Internet Archive ID as part of the URL. Instead of minting new unique identifiers for each item, we thought that using either the Internet Archive identifier or an ARK[8] would work well as unique identifiers for each object. Though ARKs are definitely unique and persistent, the Internet Archive identifiers are useful for humans in that they contain some elements of the title, author, and publication year or volume number. Authors, the other piece of data in our scheme, which would receive a URI, didn't already have an identifier, so one was created on ingest that uses the full name of the author as found in the authority controlled MARC 100 or 110. Since there is a strong possibility that at some point we will have two authors with the same exact name, we rely on the fact that Drupal will not automatically create two URIs that point to the same node. Instead, it will append a numerical suffix to ensure that the URI is unique. We find the convenience of automatic URI generation to be an acceptable trade-off for the loss of semantic specificity. Drupal's "Redirect" module also has the capability to automatically create a redirect when the URI to a node

is changed, which ensures that no piece of content is "lost" due to a URI change.

Online Books: Ontology Selection and Data Mapping

After creation and ingest of the data, the next step in getting them published as LOD is to select vocabularies and ontologies and apply them to the data on our page. One of the great, and also slightly troubling, things about RDFa and RDF is that anyone can publish a vocabulary. Specialist vocabularies exist for anything from botanical taxonomy to used cars. When choosing a vocabulary we looked for "5-star" vocabularies (Vatant, 2012), that is, vocabularies that are published at a stable and open URI, are well documented with both machine- and human-readable files, and link to and reuse other vocabularies rather than re-inventing terms. We also looked for vocabularies that were widely used, since following common practices promotes interoperability among disparate data sets. There are downsides, however, to implementing common vocabularies that are not very semantically specific, which can hamper rather than promote interoperability. For example, what is the Dublin Core "description" concept (dc:description) when applied to a book? Is it an abstract? Is it a book review? Happily it's not necessary to apply only one vocabulary or term to a given piece of data. We assume that we will continue to update and add mappings to our data over time to improve the semantic specificity, and Drupal makes this process relatively painless.

We started our ontology mapping by doing an environmental scan, looking at other institutions who have published their bibliographic data as linked data (Figure 1). We also looked at some of the potential data sources we would be linking to, to see how we could best harmonize with their model. Data from the British Library (British Library, 2012), Bibliotheque Nationale de France (Bibliothèque nationale de France, n.d.), Deutsche Nationalbibliothek (Deutsche Nationalbibliothek, 2012), and Biblioteca Nacional de España (BNE) as well as OCLC WorldCat and Europeana were analyzed. The data models of BNE and BnF follow the Functional Requirements for Bibliographic Records (FRBR) model. BNE's MARC21 to RDF/OWL mappings are very thorough and well documented[9] but because they primarily use the FRBR vocabulary, were not easily applicable to our project. We are not implementing a data model, or using any vocabularies, following the FRBR model simply because our existing data is not currently mapped to FRBR. The process of "FRBRizing" the bibliographic data is outside the scope of our LOD project, but if the Libraries do move to a FRBR model for ILS data in the future, we will also strongly consider adopting a similar model for Online Books.

Unlike the European national libraries, OCLC has chosen to use schema.org vocabularies, commonly used for HTML microdata[10] markup, to describe their linked data. Like Dublin Core, schema.org vocabularies are widely used on the Web, are simple, and are supported by all the major

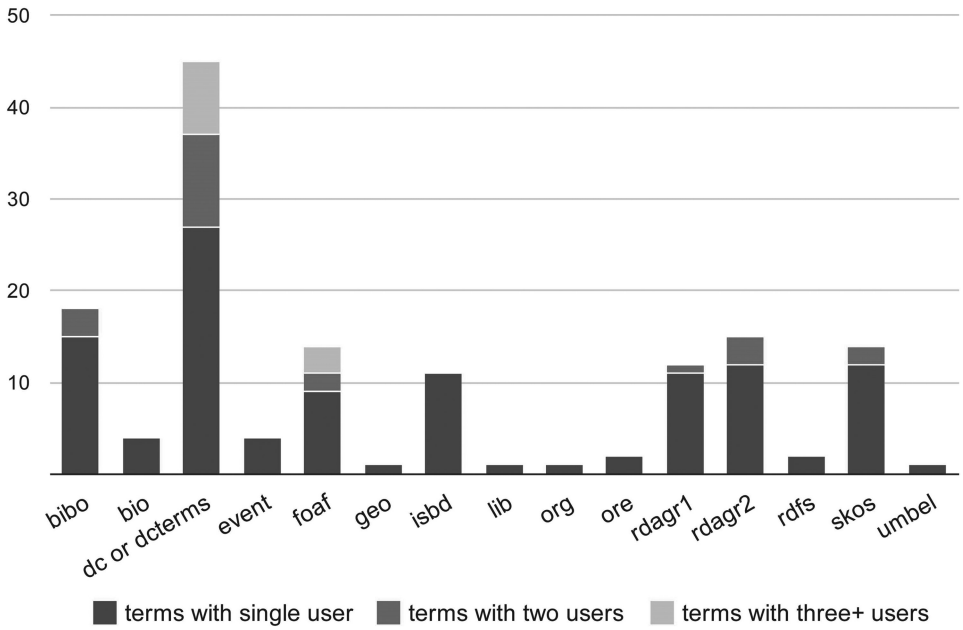

FIGURE 1 Appearance of terms from common vocabularies (Color figure available online.) This chart reflects the use of terms from common (non-institutionally specific) ontologies as found in the data models or term mappings from the British Library, Bibliothèque nationale de France, Deutsche Nationalbibliotek, Europeana and VIAF. 154 total terms were found for use in title, subject and author descriptions. 85 terms were used by more than one institution, with the majority of shared terms coming from the Dublin Core or Dublin Core Terms namespaces. Some vocabularies—specifically BIO, EVENT and GEO—are used only by the British Library.

search engines. As such, it should work well in applications that aggregate data from a variety of sources. We considered using OCLC's schema.org vocabulary, but because the library specific schema.org terms they are using are not yet officially approved by schema.org, they don't meet our criteria for a "5-star vocabulary." Since it's not difficult to add namespaces and apply vocabulary terms to data in Drupal, we've chosen to wait and add any schema.org library-specific terms when they are formally published.[11]

Though not library-specific, we also looked closely at the Europeana Data Model (Europeana, 2012) and all the ontologies used within EDM. The Europeana model is a sophisticated model based partially on CIDOC-CRM.[12] It is meant to accommodate many types of cultural heritage materials, from books and archival papers to paintings, sculptures, and videos—providing cross-domain integration while maintaining the specificity of individual domain standards. It enables the distinction between the real object, whether born digital or analog, and its representation on a Web page and between the properties of the original object and its metadata. Such a robust model that is designed for use with both library and museum data was very

Author Data Fields	Predicates
Full Name	dc:title, foaf:person
Titles	dc:bibliographicResource prop
First Name	foaf:givenName, foaf:firstName
Last Name	foaf:familyName,foaf:family_name
CorporateName	foaf:Organization
SameAs	owl:sameAs

Book Data Fields	Predicates
Author(s)	dc:contributor rel
Author (person)	foaf:person dc:contributor property
Author (corporate)	foaf:organization dc:contributor property
AuthorPerson sameAs	foaf:person dc:contributor rel
AuthorCorporate sameAs	foaf:organization dc:contributor rel
Full Title	dc:title, property ; skos:prefLabel
Cover Image	foaf:depiction
Serial Parent Title	dc:isPartOf rel
Series Title	dc:isPartOf rel
Next Title	dc:isReplacedBy rel
Previous Title	dc:replaces rel
Serial Volumes	dc:hasPart rel
Abstract	dc:description
LCSH	dc:LCSH rel, skos:inScheme rel id.loc.gov;
Subject	dc:subject prop
Date Term	dc:subject prop, dc:coverage, dc:temporal
Geographic Name	dc:subject prop, dc:coverage, dc:spatial
Copyright	dc:license prop
CC sameAs	owl:sameAs rel (CC license)
Metadata Copright	dc:license prop
Metadata CC sameAs	link to CC license as appropriate
Item Type	dc:text
Language	dc:language property
Volume	bibo:volume prop
Date of Publication	dc:copyrightDate prop
Place of Publication	dc:spatial prop
Place of Publication sameAs	dc:spatial rel
Publisher	dc:publisher prop
Year of Publication	dc:date prop
Year of Publication (end)	dc:date prop
DSpace Identifier	dc:identifier prop , bibo:handle
DSpace Identifier sameAs	dc:identifier rel , bibo:handle rel
Catalog Barcode	dc:identifier prop
Barcode sameAs	dc:identifier rel
Catalog ID	dc:identifier prop
Catalog ID sameAs	dc:identifier rel
Internet Archive	dc:identifier prop
Internet Archive sameAs	dc:identifier rel
ISBN	bibo:isbn13, bibo:isbn10 property
ISSN	bibo:issn prop
OCLC number	bibo:oclcnum, dc:identifier prop
OCLC number sameAs	owl:sameAs rel

FIGURE 2 Predicates used in Smithsonian Libraries data model.

attractive given some of our proposed use cases, however it was difficult for us to conceptualize how this data model could be replicated with Drupal and RDFa, plus we had already constructed a simpler data model based on our digitization process. Instead of changing our data model, we focused on the ontologies used within the EDM.

After reluctantly ruling out both RDA vocabularies and the library-specific schema.org vocabularies because they do not yet meet our "5 star vocabulary" criteria, and based on the frequency of use in Europeana and European national libraries as well as VIAF, we chose to primarily use Qualified Dublin Core, along with FOAF and BIBO terms (Figure 2). As such, we fell back on the standard MARC21 to Dublin Core data mappings. Because there is much we don't know about how we and others intend to reuse the data, and because ontologies and Web standards are constantly evolving, we plan to revisit all those choices, along with the data model itself, sometime next year. In the meantime, we are closely monitoring the development of the RDA and schema.org vocabularies, and as soon as they meet the "5 star" criteria, particularly in regards to the adoption of hash/slash URIs and stable namespaces for terms, we plan to reconsider their use. As the publication of LOD by other institutions grows, we will also be re-evaluating our data model as it relates to interoperability with other data sets. It is likely that additional data elements and additional vocabularies will be added in the future to enable novel reuse of others' data or reuse of our data by others.

Books Online: Publishing LOD

After selecting the vocabularies and mapping them to our data, the next steps are fairly straightforward thanks to Drupal. There are three things to do in order to publish the data: enable the RDF modules, specify the namespaces of the chosen vocabularies, and apply the mapped terms to the published data.

In Drupal 7, you must first enable the RDF module, and then the External RDF Vocabulary Importer, RDFx and RDF UI, and RESTful Web services modules. Under the RDF publishing settings you can review the default existing vocabulary mappings Drupal has provided for selected fields in your nodes and taxonomies and you can, on the namespaces tab, specify additional vocabularies that you want to use. A useful list of many available vocabularies and their namespaces can be found at the Linked Open Vocabularies[13] website. After adding namespaces, Drupal will automatically use the values found in the RDF document at the namespace URI to help auto-fill the field mappings for the data elements as you assign them.

Books Online: Linking

The next, most important, and arguably the most time consuming step in making "5-star" LOD is creating links to other data sources. We have not yet begun to systematically create these links, but when we do, we will identify and store the URIs for the corresponding data and then publish them along with our data using the owl:sameAs predicate.

In order to discover the external URIs, we plan to export the data from Drupal and use a tool such as LODRefine[14] to query those data sets with SPARQL end-points. Of the data sources we want to link to, id.loc.gov and DBPedia have SPARQL end-points and VIAF currently does not. The first goal for linking our online books will be to create links from our author fields to VIAF using LODRefine, possibly leveraging additional data from our ILS' authority records. If we do not have unambiguous strings, that is, identifiers to match on, a human will need to disambiguate them before importing the URIs into Drupal. Besides LODRefine, we are also looking at the Karma data integration tool[15] developed at the Information Sciences Institute, University of Southern California, to create links. One interesting feature of Karma is that it has machine-learning capabilities that can improve the matching algorithm, given a little upfront work from humans.

After linking to VIAF, we hope to try to link author data to DBPedia, but given the relative obscurity of many of the authors in our data set at this time, we don't expect to create many links. In addition to creating author links, we plan to also create links to id.loc.gov for the LCSH and possibly for MARC country and language codes. If we are confident in the links we've created, we hope to at some point display data from those remote data sources,

particularly biographical information from DBPedia and VIAF, alongside our data, pulling it in on the fly using the SPARQL Views module.

TAXONOMIC LITERATURE II: LINKED DATA FOR AND ABOUT BOTANISTS

Though the majority of the Libraries' data is contained in our ILS, much of that data is duplicated elsewhere—in other libraries and in WorldCat. Though it is valuable to publish this data as LOD, we also have special data sets such as uncataloged collections inventories and full-text documents that are unique to our institution. Making these datasets available as linked open data, particularly 5-star linked data, is where libraries can really bring added value and richness to the web of data. One such valuable dataset that we have is Taxonomic Literature II.

Taxonomic Literature II, formally known as *Taxonomic Literature: A Selective Guide to Botanical Publications and Collections With Dates, Commentaries and Types* (Stafleu, 1976) and often abbreviated as TL-2, includes information on botanists and their publications from 1753 to 1940. TL-2 is the premier publication of the International Association for Plant Taxonomy (IAPT), and we have been able to digitize it with their generous cooperation and support. The rights granted to us by IAPT also include permission to provide data from TL-2 openly on the Web, as the sole, authoritative source for that data online.

TL-2 is organized on two levels, first by author, and then by their publications. Each author entry contains a unique abbreviation for the author that is used throughout TL-2 and is commonly used by botanists in their research when citing other botanists' work or major publications. The information for each of the authors' publications includes a unique number for the publication within the TL-2 corpus, the full title, a shorter title used in the indexes of TL-2, publication information, and sometimes a unique abbreviation for the publication based on the title of the volume.

TL-2 provides a variety of information about a botanist that may include a brief biography, institutions or organizations where they worked, herbaria containing their collected plant samples, citations to further information about the botanist, and their publications listed in chronological order (Figure 3). A common use of TL-2 is to learn more about a botanist when their name or one of their publications is encountered during routine research. The information in TL-2 provides context, additional publications, or reveals some of the history of an author's career. In some cases, TL-2 can provide references to handwriting samples or, in rare cases, references to their images on postage stamps.

Because of its use of unique identifiers and structured entries, TL-2 is essentially a database published as a series of 15 print volumes including two

Darwin, Charles Robert (1809-1882), British evolutionary biologist. (*Darwin*).

HERBARIUM and TYPES: plants collected on the voyage of the Beagle (1831-1836): CGE, K, MANCH, MO, P, TCD (alg.).
Ref.: IH 1: 152.

NOTE: The literature on Darwin and his work is immense. No attempt is made here even to present a review of the most important biographies and studies. In a work on taxonomic literature, however, Darwin should have his place, even if only symbolic, because of his immense influence also on the development of ideas in taxonomy leading to the development of modern evolutionary biology. For a list of the main recent literature on Darwin see Smit p. 904-913.
Main bibliography: R. B. Freeman, The works of Charles Darwin. An annotated bibliographical handlist. London 1965, x, 81 p.
Autobiographies and letters: Francis Darwin, Life and letters of Charles Darwin, London 1887, 3 vols. *and* Charles Darwin, his life told in an autobiographical chapter and in a selected series of his published letters, London 1892.
Biography: e.g. G. R. de Beer, Charles Darwin, evolution by natural selection, London 1963; also as Charles Darwin, a scientific biography, New York 1965 and in DSB 3: 565-577. 1971. (bibl.)
Ideas: e.g. M. T. Ghiselin, The triumph of the Darwinian method, Berkeley and Los Angeles 1969.

HANDWRITING: e.g. in Francis Darwin, The autobiography of Charles Darwin, Dover ed. 1958.

EPONYMY: *Darwiniella* Spegazzini (1887); *Darwiniothamnus* G. Harling (1962).
Note: *Darwinia* Rafinesque (1817) and *Darwinia* Rudge (1815) are dedicated to Erasmus Darwin (1731-1802), English physician, grandfather of Charles Darwin.

POSTAGE STAMPS: Poland 20 gr. (1959) yv. 997; Roumania 55 b. (1959) yv. 1621; Soviet Union 40 k. (1959) yv. 2144; Czechoslovakia 3k. (1959) yv. 1045; DDR 10 p. (1958) yv 349; Ecuador 20 c. (1936) yv. 335.

1313. *On the origin of species* by means of natural selection, or the preservation of favoured races in the struggle for life. London (John Murray) 1859. Oct. (*Origin sp.*)
Publ.: 24 Nov 1859, p. [i]-ix, [1]-502. – *Facsimile* edition with an introduction by Ernst Mayr, Cambridge, Mass. 1964. – See Freeman 112-207, 409-441 for all editions.

FIGURE 3 Sample page from TL-2 (Color figure available online.)

cross-referenced indices. As such, it was a natural choice to digitize and place on the Web as LOD. However, beginning with a print publication presented its own challenges for conversion to an electronic, structured form.

Tl-2: Digitizing

Our process for converting TL-2 to a digital form began with scanning each page of each volume and their indices. This was completed in January 2011 and the images were uploaded to the Internet Archive as per our usual scanning process. The Internet Archive provides basic OCR text conversion using the ABBYY OCR engine. Following that, we hired a contractor to correct, rekey, and mark up the OCR text of the scanned pages to produce

two sets of files: the corrected OCR text in a structure that directly mirrors the page and volume structure of the books and an XML file containing a limited set of parsed data suitable for import into a database. Data was parsed based on the existing database-like structure of TL-2, with author biographical information in one set of fields and publication information in another set. Once we identified and corrected residual errors, we imported to a database that we exposed via a proprietary search tool on the Web in order to fill an immediate business need while we planned the move to LOD. This was the first version of TL-2 Online.[16] This version provides a simple search engine and a display of results modeled on the physical structure of the TL-2 volumes with the search results linked directly to the pages of TL-2. Additionally the site provides downloadable text and XML files for the data of TL-2.

Tl-2: Import and Conversion

The conversion to LOD began with identifying those data elements that were both most suitable for exposing as linked data and readily available or easily parsed. These included the authors' names, birth and death years, biographical summary, and TL-2 abbreviation. For the publications this includes the TL-2 publication number, full title, short title, publisher, city of publication, date of publication, and, when available, the TL-2 title abbreviations. Additionally, although we are not presenting them as LOD, we chose to include references to the physical volume and page information for each author and publication in TL-2.

Based on our earlier experience with Online Books, we realized that getting data into Drupal would require development of a custom module to handle the import. Standard import tools, such as the Feeds module, proved to be too slow for importing approximately 47,000 nodes. It became necessary to write our own import script to regulate the amount of activity in the database. The new TL-2 import has been designed to be very careful about how and when data are saved or updated in the database. Although this took a considerable investment in time, we were able to increase the performance of the import from several hours to about one. Import speed is also dependent on outside factors, such as server and network configuration, and we continue to work with our institutional IT staff on addressing them. The main purpose of this time investment was to ensure that future imports were also speedy as we continue to improve the TL-2 data set.

Because we already had a simple TL-2 site that was created quickly to meet immediate user and grant requirements, our new LOD enriched, Drupal based TL-2 site needed to replicate that functionality, including a custom search tool and a page viewer to see search results in the context of the original scanned page. For convenience, the search functionality was replicated in the same Drupal module we developed for import. Early on we

recognized that the data model we created allows us to present information in a more cohesive form than the original site. For example, search results in the original TL-2 Online were displayed separately if results came from different volumes of TL-2. In our new Drupal based search engine, the data does not follow the structure of the physical volumes and instead displays the results in a more cohesive manner, as one would expect from a single data set.

Each of the search results, be it an author or a publication, takes the user to a page for that author or publication. These pages contain HTML content with linked data embedded using RDFa in the HTML attributes of the page. The URI for the page is our definitive, persistent, and authoritative identifier for the author or publication as it relates to Tl-2. This identifier can be used by other linked data websites to link to our content. Although the user and the web browser see HTML content, the same URL can be used by a remote system or computer program to request the linked data content in a serialized, machine-readable format such as RDF/XML, NTuples, or Turtle. The choice of format is determined by who is asking for the content. That is, a Web browser asks for and receives HTML content; whereas computer code can ask for and receive the same content in XML.

Tl-2: Ontology Development and Linking

Selecting vocabularies for this initial version of TL-2 was straightforward, given that we were often able to follow the vocabulary mappings previously developed for Online Books. The major difference between TL-2 and Online Books is that TL-2 contains unique data, author abbreviation and book number that requires the creation of a new ontology specific to TL-2. We realized early on that we would need to create a small vocabulary to define these data elements and publish it online to both define TL-2 and to enable others to reuse TL-2's data. The vocabulary consists of two types of identifiers and three predicates. The types of identifiers are "Author" and "Title." The Author type has one predicate, "authorAbbreviation" and the Title type has two, "titleAbbreviation" and "tl2Number." The vocabulary is currently published at http://library.si.edu/sites/default/files/rdf/tl2.rdf.

At this point in the process, we had "4-star data," that is, data presented in linked data format, while still lacking links to other resources on the Web. The final step in this phase of converting TL-2 to LOD was to find and link to sources on the Web for all of the authors and publications in TL-2. Because of the large number of both authors and titles, and because we assumed that getting accurate one-to-one matches, particularly for authors, would be difficult, we decided to start with a smaller dataset that would have fewer ambiguous matches. Each botanist has some biographical information about them, and, for some, this includes information about the herbaria where the

specimens collected during the course of his or her studies and expeditions have been deposited. The abbreviations of these herbaria are authority-controlled, well-defined, and are easily parsed with some simple scripting tools. With the analytical help of a student intern from the University of Maryland's iSchool, we were able to work out an algorithm to parse, verify, and generate URIs for all herbaria listed in TL-2. The only real challenge we encountered was the letter "A," which is often used to start a sentence but is also the abbreviation used for the herbarium of the Arnold Arboretum at Harvard University. Once we overcame this challenge, we had links for the many thousands of herbaria references in TL-2.

We now felt confident that we could begin to identify links from TL-2 authors to LOD sources on the Web. We began by using LODRefine,[17] which is an open-source program based on OpenRefine (formerly known as Google Refine.) LODRefine and OpenRefine are both geared towards cleaning up and transforming messy data to other formats, but LODRefine is additionally designed to identify potential links to SPARQL-enabled LOD sources. To find links, we created a comma-separated values (CSV) file of the authors' information (full name, first name, last name, birth year, and death year) and used LODRefine to attempt to identify the authors at the Virtual International Authority File (VIAF). This was mostly a trial and error process, and our results were not encouraging: we only had a 5% to 10% match rate on the sample set that we started with. It's worth noting that in the 5% to 10% of names that got a match, we were confident that we had found an exact match. Most cases, however, resulted in no matches at all.

A second attempt was made to link our TL-2 authors to VIAF records, this time using a custom Perl script and the same CSV file to query the VIAF Web-based API. This effort yielded slightly better results with more entries found; however, only a small fraction of these were correct matches. Other matches resulted in multiple hits for the same name, and it was not immediately clear which was the correct entry. Some (human) effort was required to search through the list and identify the correct entry. It was clear that matching author names is a more complicated task that needs to leverage multiple avenues of effort including using LODRefine, custom scripts, and additional data sources such as Wikipedia and DBPedia, as well as possibly crowdsourcing the disambiguation of both botanist names and publications.

CONCLUSIONS, LESSONS LEARNED, AND FUTURE WORK

Our foray into LOD in the context of publishing books online and in presenting a unique data set online has been a learning experience both in terms of implementation in the Drupal ecosystem and learning the details of LOD. We dove in headfirst and though we did do basic research on linked data

principles and standards, we found that prototyping, revising, and experimenting in our actual development environment was time-consuming but essential.

Lessons Learned and Problems Encountered

Somewhat obviously, regardless of what system or software you intend to use to create and publish linked data, the importance of the data model cannot be overemphasized. Be prepared to revise it before you publish, or consider developing it iteratively. Unless you have a very narrow and defined used case, ensure that the model you end up with is extensible. Choose a small data set on which to test your model and assumptions if possible.

We started with the assumption that flattening our data and storing it in nodes would be ideal, but, as we have found working with Drupal on other projects, there are many ways to implement a given functionality and it's often difficult to know which one will give the best result until you have actually tried it. Since we are still learning about both Drupal and linked data, we imagine that as time goes on we may have to either modify our implementation to provide additional functionality or reassess it completely. Because our data is stored in databases and XML files, as well as available as RDFa through our website, there are many options for exporting and manipulating it if we do need to migrate it to another software platform or alter the data model.

The biggest surprise came after publishing our Online Books data set as LOD in a development environment. We unhappily discovered that Drupal's implementation of our data model rendered the RDF predicates inaccurate for some of our data. Our initial model assumed that a value stored in a field within the node could be expressed in both HTML and RDF as either a link ("relationship") or as a value ("property"). The Books module was built to create a URL from a single field of data when displaying that field on the web page. For example, OCLC number, stored as a value, is converted on-the-fly into a link to WorldCat when the HTML page is loaded. Drupal was aware only that the data was a value ("property") and, therefore, though it was displaying as a link, incorrectly assigned the "property" predicate to the link rather than "relationship." In order to get the correct predicate to display, we would either have to completely rewrite the Books Import module, or create duplicate fields for some pieces of data—one field to store the data as a value and one to store the link. Since rewriting the Import module would have been time and resource intensive, we chose the latter. Our data model then needed to be updated with the additional fields, and the data reimported. This was not an onerous task, but time could have been saved with more thorough planning and testing up front.

One minor issue in the otherwise very convenient namespace management functionality is that when you add a vocabulary namespace to the RDF

module, you cannot edit or delete it without a great deal of effort. For our Online Books implementation, we may try to add a feature in our Books Import module to get around this, but we hope that this problem will be corrected in the next release of either the RDF module or in Drupal 8.

Perhaps the biggest drawback of the RDF module at this time is that it requires strict namespace declaration—all namespaces must use "hash" or "slash" URIs.[18] The current RDA vocabularies' namespaces do not follow this standard, which is why we have not used any terms from those vocabularies in our initial implementation.

With respect to the "fifth star" of the LOD quality scale, we found that the act of identifying accurate links in other data sources on the Web is the most challenging and time-consuming aspect. When you have few or no common identifiers, creating the links is a daunting task, but we believe the benefit to this effort is a richer, more usable dataset.

Looking Ahead

After successful creation of links for at least a majority of our Online Books, we hope to expand the creation of LOD to the rest of our Digital Library content including videos, exhibitions, and image collections. More importantly, we hope to partner with other units at the Smithsonian to begin linking collections data from museums and archives to relevant library collections and datasets; for example, linking object information from the museums' collections management systems to exhibition catalogs or other publications in which those objects have appeared. Another major goal of our publication of LOD is to reuse our data internally, for instance, creating online exhibitions linking together illustrations and text from books, photo collections, and databases or linking botanist bibliographies in TL-2 to books in the Digital Library.

Publishing TL-2 as LOD is still very much a work in progress. So far we have addressed author names, author publications, and herbaria, but there are other elements that can be linked if we are successful in extracting and identifying them on the Web. Some examples include books, publications, and articles that exist within the blocks of narrative content; publications that identify handwriting samples or detailed biographical information; and institutions where the botanists worked or studied. We would also like to connect data from TL-2 to another project we are involved in—the Biodiversity Heritage Library (BHL).[19] BHL contains an enormous number of botanical publications with millions of species names. It will be challenging to try to connect authors in TL-2 to BHL publications or species in BHL publications back to authors in TL-2. We expect to continue refining the data from TL-2 for at least the next two years. Ultimately we hope that TL-2 Online becomes a hub for botanical citation on the Web, eventually including more recent material.

NOTES

1. Linked open data, abbreviated LOD, is simply linked data released under an open license. These terms will be used somewhat interchangeably in this paper, with linked data being the more general term.

2. Support for and use of RDFa Lite 1.1 was declared by Facebook and the major search engines around the time of the announcement of schema.org. However Google does index and use RDFa 1.0 (Sporny, 2012).

3. A SPARQL end-point is a service that follows the SPARQL protocol for RDF and enables users (most commonly other computers rather than humans) to query a data set expressed in RDF using the SPARQL language. Query results are typically returned in one or more machine-readable formats.

4. Five-star linked data includes data that (a) is available on the Web at stable URIs, (b) is available as machine-readable structured data, (c) is in a nonproprietary format, (d) uses open standards such as RDF and SPARQL, and (e) links to other people's data. The linking of your data to other data is the "fifth star."

5. https://library.si.edu/departments/web-services/macaw

6. http://openlibrary.org

7. A URI (Uniform Resource Identifier) is used to identify a Web resource. A URL (Uniform Resource Locator) is a specific representation of a URI. For example a single resource with one URI may be represented as either HTML or XML depending on the URL provided.

8. http://en.wikipedia.org/wiki/Archival_Resource_Key

9. http://bne.linkeddata.es/mapping-marc21/

10. Microdata is a specification for applying semantic markup to information in HTML pages that can be used by search engines and browsers to provide an enhanced browsing experience.

11. The W3C community to extend schema.org to better represent bibliographic information (http://www.w3.org/community/schemabibex/) is as of this writing actively working on recommendations. We assume that their recommendations will eventually be approved by schema.org, at which time we will reevaluate inclusion of that vocabulary in our implementation.

12. The International Council of Museums CIDOC-CRM conceptual reference model for describing cultural heritage information (http://www.cidoc-crm.org/).

13. http://lov.okfn.org/dataset/lov/index.html

14. http://code.zemanta.com/sparkica/

15. http://www.isi.edu/integration/karma/

16. http://www.sil.si.edu/digitalcollections/tl-2/

17. http://code.zemanta.com/sparkica/

18. http://www.w3.org/TR/swbp-vocab-pub/#recipe1

19. http://biodiversitylibrary.org

REFERENCES

Berners-Lee, T. (2010). Linked Data. *Linked Data*. Retrieved from http://www.w3.org/DesignIssues/LinkedData.html

Bibliothèque nationale de France. (n.d.). Semantic Web and Data Model [Bibliothèque nationale de France]. *Semantic Web and Data Model (data.bnf.fr)*. Retrieved from http://data.bnf.fr/semanticweb-en

British Library. (2012, August). *British Library Data Model—Book*. Retrieved from http://www.bl.uk/bibliographic/pdfs/bldatamodelbook.pdf

Clark, L. (2011, April 12). *The Semantic Web, Linked Data and Drupal, Part 1: Expose your data using RDF*. Retrieved from https://www.ibm.com/developerworks/web/library/wa-rdf/

Corlosquet, S., & Clark, L. (2011, May 3). *The Semantic Web, Linked Data and Drupal, Part 2: Combine linked datasets with Drupal 7 and SPARQL Views*. Retrieved from http://www.ibm.com/developerworks/library/wa-datasets/

Corlosquet, S., Clark, L., & Passant, A. (2010, June 25). How to build Linked Data sites with Drupal 7 and RDFa [PowerPoint slides]. Presented at the SemTech 2010, San Francisco, CA. Retrieved from http://www.slideshare.net/scorlosquet/how-to-build-linked-data-sites-with-drupal-7-and-rdfa?src=related_normal&rel=4796732

Corlosquet, S., Delbru, R., Clark, T., Polleres, A., & Decker, S. (2009). Produce and consume Linked Data with Drupal! In A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, K. Thirunarayan (Eds.), *The Semantic Web—ISWC 2009: The 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25–29, 2009* (pp. 763–778). Berlin, Germany: Springer. doi: 10.1007/978-3-642-04930-9_48

Deutsche Nationalbibliothek. (2012). *Der Linked Data Service der Deutschen Nationalbibliothek: Modellierung der Titeldaten. Deutsche Nationalbibliothek (Leipzig, Frankfurt am Main) 2012*. Retrieved from http://www.dnb.de/SharedDocs/Downloads/DE/DNB/service/linkedDataModellierungTiteldaten.pdf?__blob=publicationFile

Europeana. (2012, February 24). *Definition of the Europeana Data Model elements Version 5.2.3, 24/02/2012*. Retrieved from http://pro.europeana.eu/documents/900548/bb6b51df-ad11-4a78-8d8a-44cc41810f22

Heath, T., & Bizer, C. (2011). *Linked Data: Evolving the Web into a global data space* (1.0 ed.). Retrieved from http://linkeddatabook.com/editions/1.0/

Linked Data Cookbook: Ingredients for high quality Linked Data. (2011, December). In *Linked Data Cookbook—Government Linked Data (GLD) Working Group wiki*. Retrieved from http://www.w3.org/2011/gld/wiki/Linked_Data_Cookbook#Ingredients_for_High_Quality_Linked_Data

Manola, F., & Miller, E. (2004, February 10). *RDF Primer W3C Recommendation 10 February 2004*. Retrieved from http://www.w3.org/TR/rdf-primer/

Sporny, M. (2012, February 14). *Google Indexing RDFa 1.0 + schema.org Markup [Web log post]*. Retrieved from http://manu.sporny.org/2012/google-indexing-schema-rdfa/

Stafleu, F. A. (1976). *Taxonomic literature: A selective guide to botanical publications and collections with dates, commentaries and types* (2nd ed.). Utrecht, Netherlands: Bohn, Scheltema & Holkema.

Vatant, B. (2012, February 10). *The wheel and the hub: Is your linked data vocabulary 5-star?* [Web log post]. Retrieved from http://bvatant.blogspot.com/2012/02/is-your-linked-data-vocabulary-5-star_9588.html