Towards a BIBFRAME Implementation: The bibliotek-o Framework

Jason Kovari
Cornell University, USA
jak473@cornell.edu

Steven Folsom
Cornell University, USA
sf433@cornell.edu

Rebecca Younes
Cornell University, USA
rebecca.younes@cornell.edu

Abstract

bibliotek-o is a framework for modeling bibliographic metadata as linked data, consisting of the BIBFRAME ontology at its core. This paper presents the background and motivation behind the bibliotek-o framework, including an overview of the model, ontology principles and best practices guiding its development, a description of aligned tooling under development, and a report on the project's status and outputs. A small sample of discrete ontology design patterns in which bibliotek-o deviates from BIBFRAME is provided to demonstrate motivations and modeling principles. Our goal is to illustrate the strengths of BIBFRAME, while suggesting areas where BIBFRAME should evolve to a more streamlined and expressive model, such as in the treatment of Activities and Content/Carrier/Media Types. We aim to encourage feedback and community engagement in ongoing development of the framework outlined in this paper.

Keywords: bibliotek-o; BIBFRAME; bibliographic metadata; data modeling; linked data; ontology development

1. Introduction

bibliotek-o is a framework for modeling bibliographic metadata as linked data based on the BIBFRAME ontology (http://id.loc.gov/ontologies/bibframe.rdf), consisting of the BIBFRAME ontology at its core; the bibliotek-o ontology, which both extends and provides alternative models to BIBFRAME; defined fragments of external ontologies, both within and outside the bibliographic domain; and an application profile specifying the recommended implementation of these ontologies. Our aim is to illustrate the strengths of BIBFRAME, while suggesting areas where BIBFRAME should evolve to both simplify the model and be more expressive.

A joint effort of the Andrew W. Mellon Foundation funded Linked Data for Libraries Labs (LD4L Labs: http://ld4l.org/ld4l-labs/) and Linked Data for Production (LD4P: http://ld4p.org) projects, this work represents significant effort by a large group of colleagues from Columbia, Cornell, Harvard, Princeton and Stanford Universities as well as the Library of Congress (hereafter "LD4 Ontology Group"). LD4L Labs and LD4P are complementary efforts in support of tool development, RDF metadata production, community engagement and ontology development in the library realm.

The broader library community should determine implementation and evolution of BIBFRAME through experimentation and analysis, facilitated by transparent processes; bibliotek-o represents the LD4L Labs and LD4P approach to this analysis. bibliotek-o is not intended to replace or compete with BIBFRAME. Instead, it seeks to expand and provide proofs-of-concept for alternative modeling patterns to BIBFRAME in select modeling areas where the LD4 Ontology Group recommends more expressive and/or streamlined models without losing semantics.

Note that both BIBFRAME and bibliotek-o represent "core" bibliographic descriptive practice. Within the LD4P project, community-based domain ontology extensions are being developed concurrently with bibliotek-o development; these efforts at times inform and are informed by bibliotek-o, but represent a separate development stream.

https://doi.org/10.23106/dcmi.952137885

This paper presents the background and motivation behind the bibliotek-o framework recommendation, the ontology principles and best practices guiding its development, an overview of the model, a description of tooling under development in support of analysis and testing of the model, and a report on the project's status and outputs. To demonstrate motivations and modeling principles, a small sample of discrete ontology design patterns in which bibliotek-o deviates from BIBFRAME is provided.

2. Background and Motivation

Principles behind bibliotek-o were introduced to the DCMI community during the 2016 conference in Copenhagen, DK. Focusing on modeling principles and select extension efforts, the presentation by Folsom and Kovari (2016) provided a very high-level overview of ontology efforts within the space of LD4L Labs and LD4P. Since then, efforts related to bibliographic data modeling have progressed substantially into a framework including multiple ontologies alongside an in-development application profile.

Development on bibliotek-o began with an assessment of BIBFRAME 2.0, focusing on the question of alignment with recommendations written by Rob Sanderson (2015) in review of BIBFRAME 1.0. Concurrent to Sanderson's development of the 2015 report, the Linked Data for Libraries (2014-2016) Ontology Group deemed BIBFRAME 1.0 insufficient for the description of library resources, see: https://ld4l.org/ld4l-2014/overview; as a workaround, the group developed a temporary ontology, modeled and implemented solely for achieving the goals of the LD4L 2014-2016 project and illustrating the concrete implementation of its recommendations. With the start of LD4L Labs and LD4P in 2016, the newly formed LD4 Ontology Group wished to deprecate the LD4L 2014-2016 ontology, hoping to implement BIBFRAME 2.0 wholesale under the assumption of full alignment with Sanderson (2015) and other recommendations. This was not the case.

During the alignment analysis, the LD4 Ontology Group noted significant improvements over BIBFRAME 1.0; however, there remained areas of "core" description not provisioned in BIBFRAME 2.0, as well as modeling decisions made by the BIBFRAME architects with which the LD4 Ontology Group disagreed. Thus began the development of the bibliotek-o framework. At the framework's core is BIBFRAME; bibliotek-o builds upon BIBFRAME and cannot be implemented without select BIBFRAME patterns, alongside a number of other external ontology fragments ("target" ontologies).

The goal of the LD4 Ontology Group is to provide an ontology as the basis for an RDF cataloging tool and a MARC-to-RDF converter, and to use the resulting RDF instance data for analysis and for testing hypotheses. We hope that bibliotek-o will provide for a richer model to represent bibliographic data than using BIBFRAME alone; however, the purpose of this development is to provide the ability to analyze both BIBFRAME and the bibliotek-o framework to determine whether either provides models that adequately balance cataloging use cases and users' discovery needs. The metrics for evaluation of either framework remain undeveloped.

The LD4 Ontology Group wishes to engage the community in development of bibliotek-o as a method of analyzing BIBFRAME. To do so, we demonstrate alternative ontology design patterns that we believe more closely align with linked data principles; further, we believe that these patterns yield more accurate and queryable models than the corresponding BIBFRAME patterns. Through this development, we aim to foster a dialogue with the community pertaining to alternative models for consideration as BIBFRAME evolves in future versions. If the community decides bibliotek-o modeling more successfully addresses cataloging use cases and user querying expectations, we hope that bibliotek-o's extensions and alternative design patterns will converge with BIBFRAME in future releases. Although Library of Congress is represented on the LD4 Ontology Group and was deeply involved in discussions around bibliotek-o development, there is no defined plan regarding convergence of BIBFRAME and bibliotek-o.

https://doi.org/10.23106/dcmi.952137885

The LD4 Ontology Group does not anticipate supporting bibliotek-o in perpetuity; instead, our aim is to encourage community discussion around bibliographic modeling questions and deprecate bibliotek-o as the community decides upon modeling patterns. During this period of transition, LD4P partners intend to create linked data for use by the community using the bibliotek-o model as well as the BIBFRAME model. This will assure the community that our data model is safe to implement and our instance data is safe to consume even in the near term if so desired.

It is important to note that not all LD4P members are using the bibliotek-o framework. Library of Congress and Stanford University are building their LD4P projects from BIBFRAME without the extension and alternative patterns provisioned as part of bibliotek-o. This further affords assessment of the two models and resultant data; in addition, this divergence encourages future investigation of the effects of multiple "core" ontologies and frameworks in both cooperative cataloging and discovery environments, an issue that will absolutely arise in a shift to linked data production and consumption. As a collaboration, LD4L/LD4P expect to share data with each other regardless of the models we use. This underscores the need to consider strategies to account for the fact that there are multiple existing core ontologies already in use in the wider library community (e.g., BIBFRAME, RDA Elements, Schema.org, CIDOC CRM).

3. Development Process

The bibliotek-o ontology underwent a formal development process in April-December 2016. Self-selected representatives from each institution, including Library of Congress, formed an ontology development group to bring proposals back to the full ontology group.

The development subgroup began with a detailed alignment of BIBFRAME to the LD4L 2014-2016 ontology; this revealed areas where BIBFRAME 2.0 had not implemented those recommendations, as well as areas that neither ontology modeled adequately. After prioritizing these areas (time and resource constraints required that many would have to be left for future work), the group engaged in frequent and regular analysis and discussions that resulted in a series of recommendations documenting new requests to Library of Congress for BIBFRAME modifications, including motivations, design pattern diagrams, and complete enumeration of affected terms.

Throughout this process, our default stance was to ask Library of Congress for changes in BIBFRAME before implementing in the bibliotek-o ontology. These requests included deprecating BIBFRAME classes and predicates and adopting those from more established ontologies, addition of semantics via OWL axioms, new modeling patterns that we deemed more expressive, and miscellaneous small changes and corrections. BIBFRAME evolved in some substantial ways in response to these requests: a few properties were changed from datatype to object properties (e.g. bf:projection, bf:contentAccessibility); new classes were added to correspond with (as domains or ranges of) new object properties; inverse properties were declared for most BIBFRAME object properties; properties were defined as symmetric where appropriate; domains and ranges of some properties were removed to broaden their applicability; and the BIBFRAME agent classes were declared subclasses of the corresponding FOAF properties. There were also a number of smaller changes made in response to the LD4 Ontology Group recommendations.

Where Library of Congress chose not to follow LD4 Ontology Group recommendations, the group made decisions about whether to defer to the existing BIBFRAME implementation or to proceed with an implementation in its own namespace. We took the former course for design patterns that we deemed not sufficiently important to warrant deviation from BIBFRAME, and the latter for those we considered of crucial importance; these resulted in the bibliotek-o ontology.

https://doi.org/10.23106/dcmi.952137885

4. Design Principles

bibliotek-o differs from BIBFRAME in both modeling principles and modeling patterns that derive from those principles. The principles guiding the development of bibliotek-o include:

- Reuse and align with existing external vocabularies to promote data exchange and interoperability.
- Conversely, define terms broadly enough for reuse by external data sources.
- Use OWL axioms and RDF constructs such as domain and range in moderation to provide expressivity without overly constraining the ontology and the data it can model.
- Prefer object properties, structured data, and controlled vocabularies over unstructured literals.
- Prefer the simplest, most streamlined model capable of faithfully representing the data; adopt a single method of expressing a relationship or attribute in order to minimize query paths.
- Prefer atomic over composite data representation, e.g. bf:MovingImage and bf:Cartography instead of a class ex:CartographicMovingImage, which is an RDA content types.

Differences in modeling patterns are described in the following sections.

5. Overview of the bibliotek-o Framework

Figure 1 provides a high-level visualization of the bibliotek-o framework.

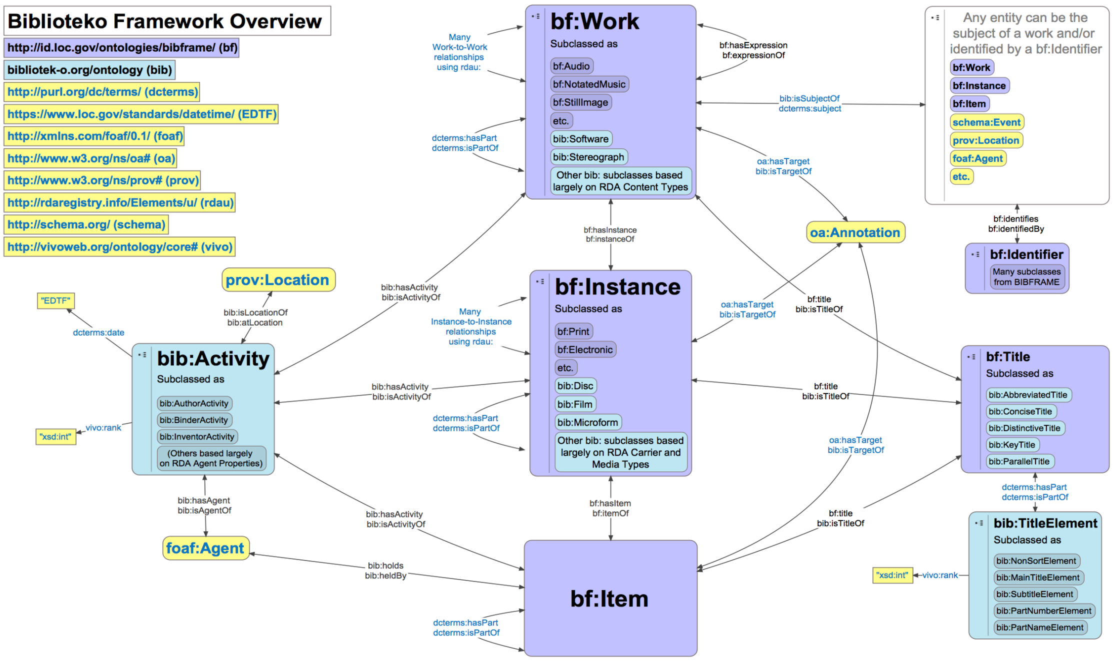

FIG. 1. bibliotek-o Framework Overview

As stated earlier, BIBFRAME is the base ontology within the bibliotek-o framework; the bibliotek-o ontology and other target ontology fragments provide additional semantics for core bibliographic descriptive practice. While these ontologies expand well beyond what could be considered a "core description" or "core record", "core" in this context means provisioning for general cataloging descriptive practice. Neither BIBFRAME nor bibliotek-o are intended to sufficiently provision for specialized cataloging practice. For instance, neither ontology would

https://doi.org/10.23106/dcmi.952137885

enable a rare materials cataloger to sufficiently describe resources without either substantial loss of granularity or omission of crucial data points.

Again, one foundational principle of bibliotek-o is the reuse of existing classes and properties where semantics align and there has been substantial community development and adoption; this is outlined in the Reuse and Alignment Principle by LD4L Labs / LD4P Ontology Group (2016). For instance, dcterms:subject has deep usage in the library domain; we thus chose to reuse the dcterms predicate rather than implementing bf:subject. BIBFRAME does not reuse existing ontology fragments or patterns, instead opting to mint new terms and in a limited set of cases assert subclass relationships to external classes. The reasons for this are beyond the scope of this paper; see Library of Congress (2017).

Other high-level areas of divergence between BIBFRAME and bibliotek-o are described in design pattern documents housed in the bibliotek-o wiki (https://wiki.duraspace.org/x/H5TBB); these patterns do not represent a full discussion of the points of deviation, which is currently in development and will be available at the same location once complete. Examples of these divergent patterns are provided in the following section.

Because bibliotek-o extends beyond BIBFRAME and in many cases adopts richer and more expressive models, transformation from bibliotek-o to BIBFRAME is lossy. That said, part of this effort is to determine whether more granular modeling with precise semantics has a positive effect on data querying and thus user discovery (though the current LD4L Labs and LD4P projects do not include a customized discovery environment). As data in both BIBFRAME and bibliotek-o is made available and consumed by both ourselves and others, we can evaluate whether bibliotek-o's streamlined and more resource-centric model meets our needs.

6. Legacy Data and Object vs. Datatype Properties

A significant challenge for an ontology representing bibliographic metadata is to bridge the competing demands to both migrate the existing highly detailed and nuanced data, and prepare for a future of original cataloging in RDF that captures data in meaningful and useful ways with a real-world orientation.

While BIBFRAME is often oriented towards preserving existing MARC data in its current format as string literals, the LD4 Ontology Group has emphasized modeling for the future without loss of legacy data. bibliotek-o thus replaces many BIBFRAME datatype properties with object properties, using the former only for data which is truly unstructured by nature, such as bf:responsibilityStatement and bf:provisionActivityStatement (though this goal has not been fully realized in the current version of bibliotek-o due to time constraints). Where tools do not currently exist to meaningfully parse and structure the legacy data, the bibliotek-o framework recommends using object properties and creating resources to hold this data in a generic datatype property such as rdf:value, and defines a custom datatype to flag such data for future processing. This approach prevents distorting the model in order to accommodate unstructured data, while nevertheless preserving that data for future integration into the model.

7. Modeling Patterns

The sections below demonstrate select design patterns where bibliotek-o offers alternative modeling from BIBFRAME and one pattern where we demonstrate an extension pattern for BIBFRAME. The LD4 Ontology Group believes that these patterns provision for better descriptive practice. These patterns, and others, are documented in significantly more detail in the bibliotek-o wiki.

https://doi.org/10.23106/dcmi.952137885

7.1 Activities

BIBFRAME makes a distinction between Provision activities (Distribution, Manufacture, Publication and Production) and a Contribution activity, the former applying to Instances and the latter to Works. While the LD4 Ontology Group understood the reasoning behind this distinction, we anticipated that the profiling of these classes would be closely aligned and that the sharp division between Instance- and Work-related activities might not be fully sustainable. We believe that the bifurcation is unnecessary and results in overly complex query paths, which should be tested once datasets are available.

As a result, we defined a general Activity pattern that provisions for explicit roles through subclassing of the bib:Activity class which links Agents to Works, Instances and Items, eliminating the distinction between provisions and contributions. This pattern also allows potential extension to Activity relationships with other types of resources, such as events. This basic design pattern is adopted in other ontologies, such as the Schema.org Action class (http://schema.org/Action) and the CIDOC CRM Activity class (http://www.cidoc-crm.org/Entity/e7-activity/version-6.2). While there was an attempt to reuse related terms from these other models, the LD4 Ontology Group agreed to mint new bibliotek-o terms until we had more experience with the pattern's semantics and our related use cases. Alignment with related external models was identified as future work. The diagram below provides a visualization of the bibliotek-o model:

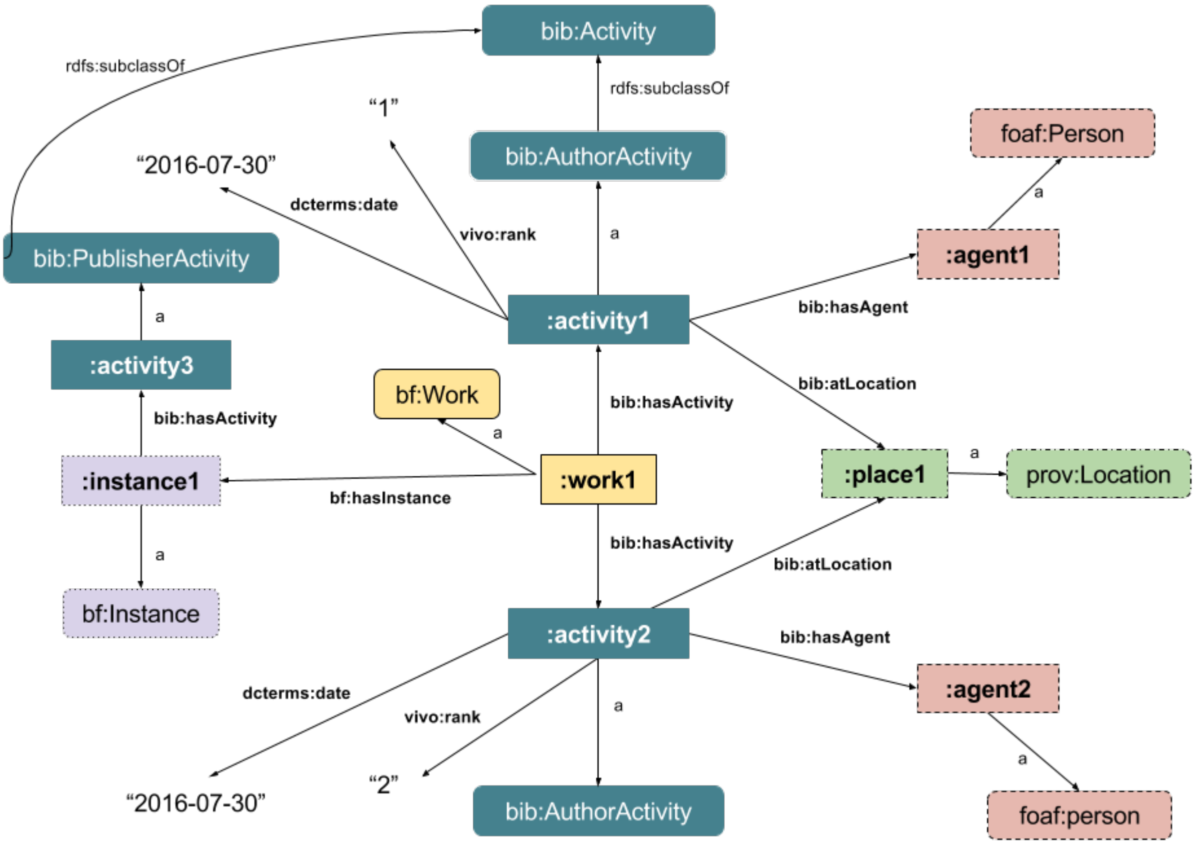

FIG. 2. bibliotek-o Activity Model

https://doi.org/10.23106/dcmi.952137885

7.2 Content/carrier/media

BIBFRAME asserts content types, carrier types and media types by establishing bf:content/bf:Content, bf:carrier/bf:Carrier, bf:media/bf:Media patterns. This modeling creates two potential means for stating the same thing: through subclassing bf:Work, bf:Instance and bf:Item, or through the property/class pattern referenced above. The LD4 Ontology Group believes that this should be avoided, as it demands that consumers of this data perform more complex queries to identify all things of a certain type (see the principles enumerated above); further, it diverges from the standard linked data practice of using rdf:type to declare that a resource is a particular kind of thing.

Rather than using the BIBFRAME multi-path pattern, bibliotek-o models content types, carrier types and media types as rdf:type assertions directly on the resource. After some testing through the creation and use of data according to defined subclasses, we may find the need to reconsider the class hierarchies and related definitions.

Library cataloging practice has a long history of capturing content type, carrier type and media type. Capturing types corresponding to content and carrier directly on Works, Instances and Items through the use of rdf:type can still be interpreted as an RDA implementation pattern because it captures content/carrier/media information about library resources; thus, this pattern does follow the RDA content standard. Note, as shown in the example below, entities can have multiple type assertions.

The following diagram visualizes the bibliotek-o design pattern:

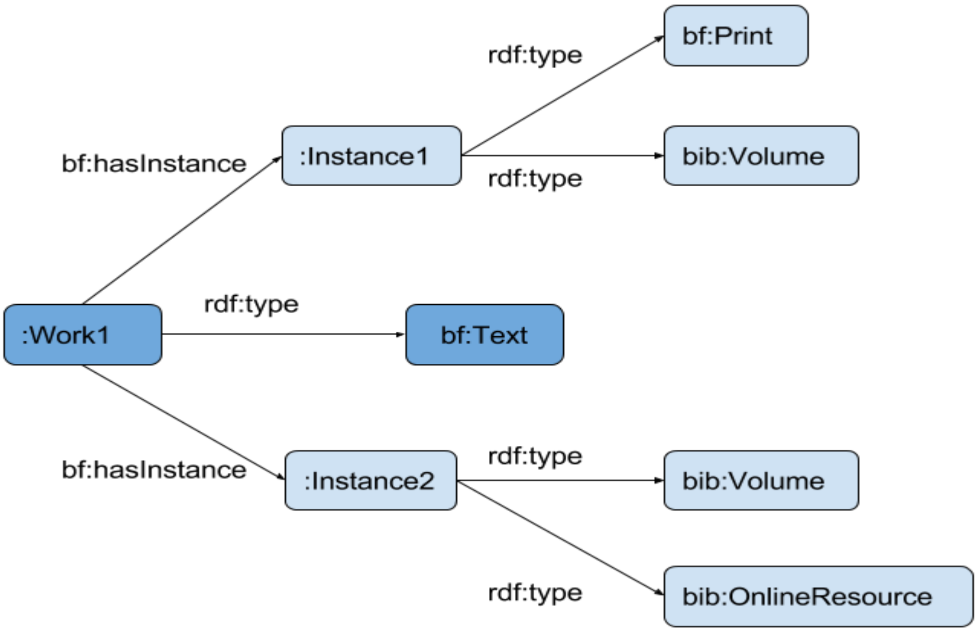

FIG. 3. bibliotek-o Content/Carrier/Media Model

8. Status and Implementation

As of the final submission of this paper (September 2017), the following technical outputs have been achieved:

- Publication of version 1.0 of the bibliotek-o ontology:
    - OWL file: http://bibliotek-o.org/ontology.owl
    - Human-readable documentation: http://bibliotek-o.org/ontology.html
    - GitHub repository: https://github.com/ld4l-labs/bibliotek-o/tree/v1.0.1

https://doi.org/10.23106/dcmi.952137885

- Specification of fragments from other ontologies, including the core BIBFRAME ontology, that are part of the bibliotek-o framework
- Recommendation documents for bibliotek-o principles and design patterns, published: https://wiki.duraspace.org/x/H5TBB

The following technical outputs are in progress, and are intended to facilitate implementation of the bibliotek-o framework in tooling (see section on Tooling):

- An application profile specifying expected use of classes and properties, to ensure consistent application of terms across data sets
- A mapping from MARC to bibliotek-o
- A listing of all BIBFRAME classes and properties not used in bibliotek-o

9. Tooling

Based on the bibliotek-o framework, tools for both original RDF metadata creation and conversion of existing metadata are currently under construction to facilitate further analysis of and experimentation with the framework.

In a joint effort between LD4L Labs and LD4P to begin creating RDF natively within the bibliotek-o framework, we have begun the development of VitroLib (https://github.com/ld4l-labs/vitrolib), an ontology-based cataloging editor that enables manual cataloging and is built on Vitro (https://github.com/vivo-project/Vitro), a generalized semantic web ontology and instance editor with customizable browsing. As part of this work, usability testing is being performed with catalogers to better understand how VitroLib can be customized to conform to cataloger needs and expectations. Considerable work is being devoted to ensuring that VitroLib cataloging forms not only allow for lookups of locally created data, but also provision for linking to external data in much the way that we reuse bibliographic records and authority data in current cataloging workflows. As ontology extensions become available from the LD4P domain projects, they will be incorporated into VitroLib implementations to achieve richer descriptions of particular domains. Additionally, we will begin testing customization of CEDAR (https://metadatacenter.org/tools-training/cedar-metadata-tools) for bibliotek-o during Fall 2017, customization will be based on the same in-development application profile as VitroLib.

The converter is an open source, flexible, community-extensible tool for the conversion of conventionally-formatted bibliographic metadata to RDF and linked data (https://github.com/ld4l-labs/bib2lod). In its primary implementation, it will convert core bibliographic metadata in MARC to RDF expressed in the bibliotek-o framework, based on mappings under development by the LD4 Ontology Group. It is architected for extensibility, allowing implementations that accommodate both other metadata input formats (e.g. FGDC (http://www.fgdc.gov/metadata/fgdc-std-001-1998.dtd) and CSV) and output formats that extend the bibliotek-o framework, such as the LD4P domain ontologies; in fact, an extension converting FGDC data to bibliotek-o plus the cartographic extension has already been developed. The converter is capable of converting catalog metadata at scale, allowing analysis and evaluation of large amounts of linked data built on the bibliotek-o framework.

10. Community Engagement

The LD4 Ontology Group encourages feedback on any fragments and modeling patterns represented in bibliotek-o. Documented in GitHub (https://github.com/ld4l-labs/bibliotek-o), we encourage the submission of GitHub issues to raise concerns about particular classes, properties and patterns; and pull requests to suggest actual changes to the ontology file and related documentation.

https://doi.org/10.23106/dcmi.952137885

While in-depth ontology development for bibliotek-o concluded December 2016, development of extensions and application profiles is intended to continue, based on both LD4P domain extension work and community input. The bibliotek-o versioning protocols and change and release management process have been published in the GitHub repository (https://github.com/ld4l-labs/bibliotek-o/); these processes will ensure transparency of ontology development moving forward.

11. Select Outstanding Issues and Future Work

While we cannot address all outstanding questions adequately in this paper, there remain a number of issues in the bibliotek-o framework as well as in BIBFRAME that the LD4 Ontology Group would like to begin addressing with the community. These include, though are not limited to:

- Should BIBFRAME, bibliotek-o or other ontologies be made more modular? Is there a benefit to the broader (library and non-library) community if definable ontology patterns are managed and hosted in namespaces distinct from the core ontologies? For example, some of the LD4P domain extension ontology groups have surfaced common needs for provenance modeling; rather than including a provenance model in any one extension, should it be included in a BIBFRAME or bibliotek-o core, or hosted in a namespace of its own so that it can be implemented independently of a core bibliographic ontology?
- The LD4 Ontology Group identified a number of BIBFRAME models (e.g., Awards, Administrative Metadata, Degrees, Form/Genre, Serials and Multi-parts, etc.) which require further work before submitting recommendations to BIBFRAME or implementing in bibliotek-o. We do not regard the data model as complete without the inclusion of these models.
- In addition to alternative modeling patterns, the LD4 Ontology Group has considered areas of data representation that BIBFRAME, as well as traditional data formats and schema, have not addressed. One area of extensive exploration was Attribution: the complex web of relationships (over 20 uses cases were identified) that may occur between a bibliographic resource and attributed agents, such as: pseudonyms, ghostwriters, name changes, deliberate misattribution, etc.; note: relationships between identities and agents were not within the scope of this research. While existing metadata does not express such relationships, we view it as a fruitful area for future efforts to expand beyond existing descriptive domains.
- Methods and metrics for analysis and evaluation of the bibliotek-o framework based on the RDF generated during the project by original cataloging and bulk conversion, as well as comparison with BIBFRAME implementation data.

Acknowledgements

This work is supported by the Andrew W. Mellon Foundation. The authors would like to thank the Mellon Foundation as well as the numerous colleagues who contributed to this development from Columbia, Cornell, Harvard, Princeton and Stanford Universities as well as the Library of Congress.

References

Folsom, Steven, and Jason Kovari. (2016). Ontology Assessment and Extension: a case study from LD4L and BIBFRAME. Retrieved May 25, 2017, from http://dcevents.dublincore.org/IntConf/dc-2016/paper/view/433/505.

LD4L Labs / LD4P Ontology Group. (2016). Reuse and Alignment Principle. Retrieved May 26, 2017, from https://github.com/LD4L Labs/bibliotek-o/blob/develop/doc/principles/bibliotek-o_principle_reuse_201612.md.

Library of Congress. (2017). BIBFRAME frequently asked questions. Retrieved May 26, 2017, from

https://doi.org/10.23106/dcmi.952137885

https://www.loc.gov/bibframe/faqs/#q07.

Sanderson, Robert. (2015). Analysis of the BIBFRAME Ontology for Linked Data Best Practices. Retrieved May 25, 2017, from https://goo.gl/KRiuTt.

https://doi.org/10.23106/dcmi.952137885