bibliotek-o: a BIBFRAME Extension Ontology

Following the release of BIBFRAME 2.0 (BF2) in April 2016, the LD4 Ontology Group assessed changes to the ontology in order to understand implications for implementation in the LD4L Labs tooling and for the LD4P BIBFRAME Ontology Extensions. This assessment included review of every term in BF (approximately 180 classes and 190 properties). Through this process, the group determined that composing narrative recommendations alongside OWL files implementing those recommendations would facilitate community feedback and influence the evolution of BIBFRAME. As such, the group created bibliotek-o, an ontology framework extending BF2 to handle broader aspects of bibliographic description and implementing alternative patterns for select areas of BF2.

Reusing BIBFRAME as its core, bibliotek-o is not implementable without BIBFRAME; however, bibliotek-o reuses properties and classes from already established ontologies, in addition to minting its own terms where the developers determined a lack of adequate terms for reuse. The 110 recommended changes to BF2 are delineated in a bibliotek-o to BIBFRAME Deviation CSV. In this document, we provide a concise reason alongside determination regarding whether the issue reflects a structural or non-structural change in BF2. Additionally, we composed eight documents discussing larger modeling patterns whereby we believed that BF2 could benefit from deeper analysis; these documents are summarized on the bibliotek-o : an overview wiki page.

While the bibliotek-o ontology is not intended for long-term usage or production-level implementation, w e firmly believe that decisions as important as the widespread adoption of bibliographic da ta models should be community-driven based on extensive consideration and experimentation. bibliotek-o represents LD4's experimentation in this space. bibliotek-o was not designed as a competitor to BIBFRAME; by demonstrating extended and alternative patterns, we aimed to encourage community feedback on potential solutions to areas that the Ontology Group believed could benefit BIBFRAME modeling.

Work on bibliotek-o occurred between April 2016 and December 2016, during which time the team included a member of the LC Network Development and MARC Standards Office; further, the group engaged in extensive communication with the BIBFRAME architects, based at the Library of Congress, throughout the process. Since this time, Library of Congress introduced a GitHub repository to facilitate feedback to BF2; however, this was created more than a year following the conclusion of bibliotek-o development.

The process of developing bibliotek-o led to revisions to BF2. Notably, the Library of Congress defined OWL constraints in the ontology; minted a number of new properties, including the creation of inverses for existing properties; and minted new subclasses for existing BF2 classes.

Documentation:

- Website: https://bibliotek-o.org/
- OWL file: http://bibliotek-o.org/ontology.owl
    - Human-readable OWL file: https://bibliotek-o.org/1.1/ontology.html
- GitHub repository: https://github.com/ld4l-labs/bibliotek-o
- Wiki: bibliotek-o
- Overview of the patterns in bibliotek-o: bibliotek-o : an overview

Alternative Ontology Patterns:

- Activities
    - bibliotek-o defines a general Activity pattern that facilitates explicit roles through subclassing of the bib:Activity class, which links Agents to BIBFRAME core classes (Works, Instances and Items). This pattern eliminates the distinction between provisions and contributions, thus simplifying both querying and potential extension for additional relationships between agents and bibliographic resources. This design pattern is adopted in other ontologies, such as the Schema.org Action class and the CIDOC CRM Activity class.
- Content Accessibility
    - In February 2017, BIBFRAME changed bf:contentAccessibility from a datatype property into an object property, following the Content Accessibility recommendation finalized by the Ontology Group in December 2016. bibliotek-o extends this revised bf:contentAccessibility /bf:ContentAccessibility model by making a distinction between Accessibility Hazards and Accessibility Features; bib:AccessibilityHazard pertains to aspects of the resource that cause issues with accessibility (e.g., flashing lights) whereas bib:AccessibilityFeature pertains to aspects of the resource that facilitate accessibility (e.g., Braille). Further, bibliotek-o mints specific subClasses of bib:AccessibilityHazard and bib:AccessibilityFeature.This model closely aligns with the schema.org model for content accessibility and better facilitates usage of library resources than having a general class without differentiation between hazards and features; schema.org's model was not directly reusable due to the lack of classes for these concepts.
- Content Type, Carrier Type and Media Type
    - 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 the above property/class patterns and also through subclassing bf:Work, bf:Instance and bf:Item. Explicitly creating multiple patterns for the same concept fails to capture semantic commonalities and complicates queries; 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 commits to modeling content types, carrier types and media types through subclasses of Work, Instance, and Item, alongside type assertions on individual resources. This pattern can be interpreted as RDA implementation because it expresses content/carrier/media information about library resources and thus supports cataloging according to the RDA content standard.
- Legacy Literals
    - Broadly, bibliotek-o defines 'legacy literals' to mean any existing textual metadata being migrated into RDF that does not conform to the desired model and/or application profile, and represents data as unstructured, unparsed, and unnormalized string literals. The authors of BIBFRAME are legitimately concerned with preserving these legacy literals, and have defined a broad range of datatype properties to do so. Given bibliotek-o's preference for structured, machine actionable data over note-like strings, it instead defines object properties and classes where appropriate, to provide semantically rich models and promote the future creation of structured metadata. In order to preserve legacy metadata, it defines a custom datatype legacySourceData to flag this data for future cleanup, thus achieveing both goals of an expressive data model and preservation of legacy literals.
- Notes and Annotations
    - BIBFRAME prefers the use of specified properties with expected domains of BIBFRAME core class (Work, Instance, Item) for many note types. Rather than relating a note directly to a BIBFRAME core class, bibliotek-o uses the Web Annotation (OA) model for many of these note types (e.g.: bf:credits, bf:custodialHistory, bf:historyOfWork, bf:natureOfContent, bf: preferredCitation , bf:summary, bf:review, bf: systemRequirements, and bf:tableOfContents). This deviation stems from a concern about directly attributing these data to the bibliographic resource; generally, these notes types are not necessarily intrinsically related to the resource but are asserted by the cataloger. By using the OA model, one can provide a target (i.e., the resource being described) and a motivation (e.g., summarizing) for

the note, meanwhile providing an intermediate node to separate the note from data intrinsically connected to the resource. Rather than simple text strings, the OA model provides richer expressivity, such as annotation bodies that are resources, along with specification of type and format; making multiple atomic but related annotations on a specific resource; and ascribing purposes and motivations to annotations.

- Relations
  - bibliotek-o uses a combination of BIBFRAME and RDA Unconstrained (RDAU) relationship properties. As a framework with BIBFRAME as its core, bibliotek-o retains many BIBFRAME properties (e.g.: bf:relatedTo subproperties: bf:expressionOf, bf:hasExpression, bf:otherEdition, bf:referencedBy, and bf:references); however, bibliotek-o preferences RDAU properties in place of many BIBFRAME properties (e.g.: bf:hasEquivalent, bf:hasDerivative, bf:derivativeOf, bf:hasReproduction, bf:originalVersion, bf:originalVersionOf, bf:otherPhysicalFormat, bf:reproductionOf, bf:translation and bf:translationOf); further, bibliotek-o uses RDAU properties for derivative relationships (rdau:P60250, its inverse and subproperties). The use of RDAU properties is primarily due to their more granular nature than what is afforded in BIBFRAME, as well as being a well established descriptive standard in the library community.
- Titles
  - With the addition of several constructs and patterns to the BIBFRAME model, bibliotek-o introduces a more expressive and accurate title model, including the representation of a main or primary title; modeling title parts as objects rather than literals in order to express nuanced relationships among these elements; and defining a controlled vocabulary of title origins such as binding, cover, spine, and so on, rather than using free-form literals.

Design Principles:

- External Ontology Reuse
- OWL versus RDFS
- bibliotek-o and RDA
- Object versus Datatype Properties

Presentations (included on LD4L Labs Communications and Outreach):

- 2016-10 - "Ontology Assessment and Extension: A Case Study on LD4L and BIBFRAME", Jason Kovari and Steven Folsom DCMI, October 2016
- 2017-04 - "Ontology Topic Area", Steven Folsom and Jason Kovari LD4P/LD4L Community Meeting, April 2017
- 2017-10-27 - "Towards a BIBFRAME Implementation: the bibliotek-o Framework", Steven Folsom, Jason Kovari, and Rebecca Younes. DCMI: International Conference on Dublin Core and Metadata Applications (Washington, D.C.).
- 2017-12-05 - "The bibliotek-o Framework: Principles, Patterns, and a Process for Community Engagement" (video), Steven Folsom, Jason Kovari and Rebecca Younes. SWIB 2017, Hamburg, Germany