Ontology-Based Metadata:

Transforming the MARC Legacy

Peter C. Weinstein
University of Michigan Digital Library
Artificial Intelligence Laboratories
1101 Beal Ave.
Ann Arbor, MI 48109-2110, USA
Telephone: 734-763-9074
E-mail: peterw@eecs.umich.edu

ABSTRACT

We propose a new catalog based on a formal ontological model of bibliographic relations. A hierarchy of five central concepts describes the creation of work. Each kind of relation between works occurs at a particular level in the hierarchy. Related works share data at some level of the hierarchy, yielding a tree structure that reduces redundant representation of shared attributes.

To show that ontology-based metadata is practical, we generated a knowledge base of metadata from a sample of MARC records. We implemented the ontology in description logic (Loom), mapped Machine Readable Cataloging (MARC) attributes and values to the ontology, and loaded the data into Loom with all values treated as separate instances. We then unified matching instances, and deduced relations between works. This process thus converts relationships implicit in MARC into explicit relations that are easy to utilize with computers.

Our web interface permits browsing by navigating relations between works. Ontology-based metadata can also support user inquiry and digital-library operation in other important ways.

KEYWORDS: Metadata, ontology, bibliographic relations, catalog structure.

INTRODUCTION

We propose a new catalog structure that explicitly models relationships between works. We use formal ontology with description logic to represent the model (Section 2 defines "ontology"). This approach is less well-developed than technologies such as relational databases, but has advantages that may prove invaluable. Namely, ontology facilitates computational reasoning about the data. We used reasoning to generate a knowledge base of metadata starting with ordinary MARC [16] records. In MARC, information about bibliographic relations is hidden in redundant data (values common to the related works) and in the meaning of natural language notes. In the knowledge base, bibliographic relations are explicit, labeled links amenable to manipulation by computers. This paper describes our ontology, the generation of ontology-based metadata, and an interface available on the Web for browsing the knowledge base.[1]

This research is challenging because of the complexity of bibliographic relations, and the immensity of the task of recreating metadata in a new and relatively elaborate format. Works may derive, add, describe, continue, be a part of, or copy other works in numerous ways [17, 19]. Frequently many such relations apply, so that a family of works may realize a complex web of relations. Meanwhile, the flood of new information is overwhelming the professional cataloging community. Clearly, it is imperative that any new system fully leverage the tremendous investment in MARC.

Formal ontology is well-suited for modeling bibliographic relations. Concepts are defined in terms of other concepts, forming webs of relations with structural complexity sufficient to capture the richness of the bibliographic domain. As always when building models, however, we are imposing order on chaos. The goal is to simplify reality while retaining fidelity.

Formal ontology's precision and declarative encoding also support the computational reasoning needed to make the creation of ontology-based metadata practical. Exactly which capabilities are available depends on the language chosen to implement the ontology and knowledge base. There are many options, with complex tradeoffs between the ability to reason in various ways, and the efficiency and scalability of the system. Thus, the choice of representation system will follow from the vision articulated for the creation and

[1] http://www.umich.edu/~peterw/Ontology/Beethoven/demo.html.

use of the new catalog (Section 5 discusses this issue further). To implement our metadata knowledge base, we use Loom [14], a description logic system in the KL-ONE family [24].

Ontology-based metadata can provide the following important benefits:

- Very precise queries. For example, consider an amateur musician who needs to learn a part. She will be able to pose a query equivalent to: "Find me a score for Beethoven's Fifth concerto, arranged for cello, with a recorded performance that is also available". A similar query could be posed without specifying the particular concerto. The traditional distinction between access and descriptive metadata is blurred, because all ontology-based data can be used for both purposes. Attributes designated for controlled vocabulary, however, will continue to be particularly useful for access.

- Explicit paths from vague to precise queries. A vague query identifies values desired for one or two attributes. In ontologies, attributes are defined by their relations to other attributes, providing numerous intuitively natural sequences for prompting users to articulate their requirements. Frequently, query refinement will occur by starting at the top of the work hierarchy (see Section 2) and progressing downwards. Queries can also be refined by requesting increasingly specific versions of known attributes, or by adding values for attributes associated with the known values.

- Integrating catalogs. The same kind of reasoning used to merge work descriptions when generating our knowledge base can be applied to integrate multiple catalogs. Indeed, given a work derived from a predecessor, we look for the predecessor in our collection; if we do not find it, we generate new metadata including what we know about the predecessor. These descriptions are essentially waiting to be fully instantiated by integration with other catalogs that include the preceding works. (See Section 5 for issues related to scaling-up).

- Per-service calculation of license needs. License requirements are a function of the kind of service provided, the work involved, and the licensing agreement that covers the work. The ontology's model of relations between works will provide both a language for expressing license agreements, and a means for computing fees associated with services at runtime. Much of this work remains to be done, but see Weinstein and Birmingham [22] on the use of ontologies to support agent selection of services.

Our approach has advantages over other proposals for new catalog structures. One idea, with many variations, is to extend MARC with fields for explicit links to other works. MARC, however, is highly redundant both within and especially between records [13]. Redundant data reduces the efficiency of storage and updating. The most serious threat for digital libraries, however, is the prospect of overwhelming users with numerous versions of very similar works [19]. MARC also has an archaic file organization. It is possible to convert MARC's idiosyncratic record structure, field codes, subfield codes, and value-conditioning "indicators" to more standard formats. The need to do so, however, impedes utilization of advances in mainstream database technology.

Several proposals explore using relational databases to store the new catalogs [5, 10]. These systems are fast and reliable for massive quantities of data. Unfortunately, relational technology is best suited for applications such as banking, with large quantities of highly standardized data. The complexity of bibliographic data subverts normalization [5]. Normalization is the process whereby data is divided into a separate table for each domain entity, indexed by a combination of dimensional values that provides a unique key for each record. Bibliographic data requires many descriptive dimensions, partial description at multiple levels of granularity (with any combination of dimensions), and viewing from many perspectives (access by different sequences of dimensional values). Relational design forces a commitment to particular combinations and orderings of dimensions. In complex domains, the result is a proliferation of tables that destroys efficiency and maintainability.

Object-oriented databases are appropriate for complex domains. Heaney [8] has proposed an object-oriented catalog with a model similar to our own. Actually, ontology is object-oriented—with a class structure that is declaratively encoded. In typical object-oriented systems, the class structure is encoded procedurally (in a programming language rather than in a data structure). Information that is procedurally encoded is not accessible to reasoning. For example, in standard C++ there is no built-in way to determine the type of a given object. Of course, developers can add methods to return object type—and superclasses, subclasses, and so on. If this is done in a well-principled way, then the developers are heading back down the road to description logic, or other representation systems used to reason with ontologies.

The remainder of this paper is structured as follows. Section 2 presents the University of Michigan Digital Library (UMDL)[1] ontology of bibliographic relations. Section 3 describes the generation of a metadata knowledge base from a sample of US MARC on works by or about Beethoven. Section 4 describes an interface, available on the Web, for browsing the Beethoven knowledge base. Section 5 discusses several important outstanding issues. Section 6 concludes.

2 AN ONTOLOGY OF BIBLIOGRAPHIC RELATIONS

This section first explains the idea of "ontology", and distinguishes "formal ontology" from other approaches. It then describes the UMDL ontology of bibliographic relations for digital-library content. A more detailed

description of the process of developing the UMDL ontologies and their use in the UMDL is available in Weinstein and Alloway [20]. Concept descriptions in stylized natural language, and definitions in Loom, are also available on the Web.[2]

2.1 Formal Ontology

In artificial intelligence, an "ontology" is a set of vocabulary definitions that expresses a community's consensus knowledge about a domain. This knowledge is meant to be stable over time, and reused to solve multiple problems. Note that this concrete and utilitarian approach is quite different from "ontology" in philosophy, which concerns the abstract nature of reality apart from human endeavor.

"Formal ontologies" define vocabulary with logic. The exact syntax and semantics depends on the representation language (see Genesereth and Fikes [4] on KIF, and Borgida and Patel-Schneider [2] for description logics). Syntactically, these languages are roughly equivalent to first-order logic in expressiveness, but differ by their focus on inheritance relations. These include subsumption, when one concept is a "kind-of" another, and instantiation, when an instance "is-a" member of a concept.

Formal-ontology concept definitions are usually construed, more or less explicitly, as frames. Definitions include a name, a set of relations to other concepts, and a natural-language description that serves strictly as documentation. In slightly different contexts, relations may alternatively be called "roles", "slots", or "dimensions". Concepts that restrict relation ranges are "attributes", and objects that instantiate relation ranges are "fillers". Restrictions can be placed on fillers, such as the number of fillers required, or constraints in relation to other fillers. An instantiation of a concept may be partial, with some roles filled, and others waiting for fillers. Subconcepts and instances of a concept inherit the concept's relations; subconcepts can be specialized from parent concepts by adding relations, or by further restricting filler values.

Semantically, instances denote objects, concepts denote sets of objects, and relations denote sets of tuples of objects. A concept (C) subsumes another (D) if every instance of the latter is always also the former: C —| D if and only if for every model M, $D^M \subseteq C^M$. (In logic, a "model" is a particular mapping from symbols to a universe of objects). There is debate in the description-logic community as to whether an extensional or intensional semantics is more desirable [23]. For ontologies, a Fregean semantics is appropriate [7]: The intensional meaning of a concept is a function that maps from models to the concept's extension, Ω_C: $M \rightarrow C^M$. Thus, ontologies can be understood as nets of constraints that restrict the set of possible models.

[2] http://www.umich.edu/~peterw/Ontology/ontology.html.

Another way to understand the nature of formal ontologies is to compare them to less formal structures. Consider the following spectrum of structures, used to define vocabulary, that vary by degree of formality. On the informal end are subject-classification hierarchies such as Dewey and Library of Congress. These structures are trees of unadorned symbols, where the relation between symbols, often called "is-a", is actually an imprecise commingling of subsumption and instantiation.

Moving to the middle of the spectrum, we find "informal ontologies", such as WordNet [15]. These structures have other kinds of relations besides "is-a", although "is-a" links are still the most important. Concepts can also have multiple parents: Informal ontologies are not trees, but directed graphs. Furthermore, each concept has a dictionary-style natural-language description, and this description provides the authoritative meaning of the term.

On the formal end of the spectrum are ontologies such as those in the Ontolingua library at Stanford [3]. Here, the variety and precision of relations is elaborated to the point where the entire meaning of central concepts is captured by their relations to other concepts. Concepts usually have natural language descriptions, but these are considered to be documentation only. By analogy, imagine studying a C program. You read the comments first to learn how the program works, but keep in mind that the documentation may be wrong—it is the code that executes.

Gruber's widely-quoted definition of ontology makes sense for formal ontologies, but is frequently misapplied to informal ontologies. "An ontology is an explicit specification of a conceptualization" [6]. In logic, a conceptualization identifies the objects and relations that exist in a logical universe. Outside of the context of logic, what does this definition tell us?

2.2 The UMDL Content Ontology

The backbone of the UMDL ontology of digital-library content is a hierarchy of concepts that loosely models the creation of work (Table 1). This hierarchy revises, extends, and clarifies the proposal by IFLA [10]. The creative process starts with an idea. As long as it is abstract, we call the work a CONCEPTION (we use capitals to identify a concept defined in the ontology). After the content of the work has been specified, it is an EXPRESSION. When the EXPRESSION is published, it becomes a MANIFESTATION. The physical embodiment of the MANIFESTATION is a MATERIALIZATION; if digital, the MATERIALIZATION is a DIGITIZATION. A particular copy of a MATERIALIZATION is an INSTANCE (not to be confused with a description-logic "instance" of a concept). We believe that our work hierarchy is consistent with informal definitions of work by Yee [25], and others in the library community. Note that "work" itself is not a defined concept, but an ambiguous natural-language word that is very useful for conversation about fully or partially instantiated objects in the work hierarchy.

Table 1: The work hierarchy

Concept	Description	Example
CONCEPTION	A concept, plan, or design for work.	An idea for a story.
EXPRESSION	A conception with specified content.	A manuscript for a novel.
MANIFESTATION	An expression packaged in a publishing format.	A published edition of the novel.
MATERIALIZATION, DIGITIZATION	A manifestation embodied in a physical format.	The novel coded in SGML.
INSTANCE	A copy of a digitization.	A particular file on a disk.

More precisely, each level of the work hierarchy is defined by the attributes associated with it. Figure 1 provides an overview. A CONCEPTION is identified by its UNIFORM-TITLE, CREATOR, and also SEQUENTIAL-NUMBER for numbered musical works. An EXPRESSION has a "genre": a natural-language term that we use to refer to an area of the ontology, described just below. A MANIFESTATION has a PUBLISHING-FORMAT, which is divided into RECORDED for BOOKs, JOURNALs, etc.; and LIVE for performances, ongoing scientific observations, and so on. A MATERIALIZATION has a DIGITAL-FORMAT, which includes TEXT, IMAGE, VIDEO, and other kinds of formats. Furthermore, every level of the work hierarchy is associated with CREATORs constrained to be of certain types. For example, on the EXPRESSION level there are EDITORs, TRANSLATORs, and so on.

Figure 2 illustrates genre, the most complex part of the ontology. Unlabeled relations denote subsumption, the underlined concepts in rectangles are attributes of the concept at the top of the rectangle, and the concepts in brackets are kinds-of the underlined concepts. Whereas FICTION and NON-FICTION are disjoint, a work may be one or more of SYMBOLIC, SOUND, and VISUAL.

Other proposals for new catalog structures have work hierarchies with two [13], three [8], or four [10] levels (see Velluci [19] for an overview). Ours has five. There is no advantage to having fewer levels, however, as long as every level adds information to every work. The associated attributes need to be attached somewhere.

Families of related works have a simple and elegant structure that is amenable to efficient maintenance and manipulation by computers. Namely, these structures are trees, as illustrated in Figure 3. CONCEPTIONs can have multiple EXPRESSIONs, but an EXPRESSION can have only one CONCEPTION; and so on for each level down to INSTANCE. Full metadata for a particular copy of a work includes a single (description logic) instance at each level.

Each kind of relation between works occurs at a particular level of the work hierarchy. For example, sequels are between CONCEPTIONs, translation is from one EXPRESSION to another, reprinting is at the MANIFESTATION level, reformatting involves DIGITIZATIONs, and copying is between INSTANCEs. Similar constraints apply to relations that add, continue, or describe other works rather than derive from them. Whole-part relations are different (see Section 5). In general, we call relationships describing the creation of work *ontogenic relations*.

Services that operate on higher levels of the work hierarchy involve rights that are relatively profound (and may therefore require licenses that are relatively expensive). For example, borrowing a copy of a book is a service on the INSTANCE level. A digital library might reformat a document from Microsoft Word to ClarisWorks: This would be on the DIGITIZATION level. Customized news service is on the MANIFESTATION level. Collaborative editing is on the EXPRESSION level, and licensing fictional characters would be on the CONCEPTION level.

The relationship between the levels of the work hierarchy is somewhat trickier than is immediately apparent, however. We want the lower levels to inherit from the higher. So, an INSTANCE "is-a" MATERIALIZATION, a MANIFESTATION, and so on. Thus, an INSTANCE has a UNIFORM-TITLE that can be retrieved directly without navigating all the way up the work hierarchy. We also, however, want sibling instances to share data at inherited levels of the hierarchy. In object-oriented systems, instantiating an object allocates separate memory for the members of every inherited class. Fortunately, in description logic we can have this cake and eat it too. Each concept in the work hierarchy is defined with both "kind-of" and "extends" relations to the concept above. The inherited value is then constrained to be the "same-as" the value associated by the "extends" relation. Currently, Loom does not properly enforce these "same-as" constraints, but we achieve equivalent results for our limited application by enforcing them procedurally.

This discussion covers only the core of the UMDL ontology for digital-library content. At the time of writing, the ontology includes 183 concepts, not including concepts built into Loom. Of these, however, all but 16 central concepts are primitive. In other words, most of the definitions do not have relations sufficient to justify typing an instance as a member of the concept unless it is explicitly asserted to be a member. Although the UMDL is a large project, the ontology was developed by a small group of people (see the acknowledgments below).

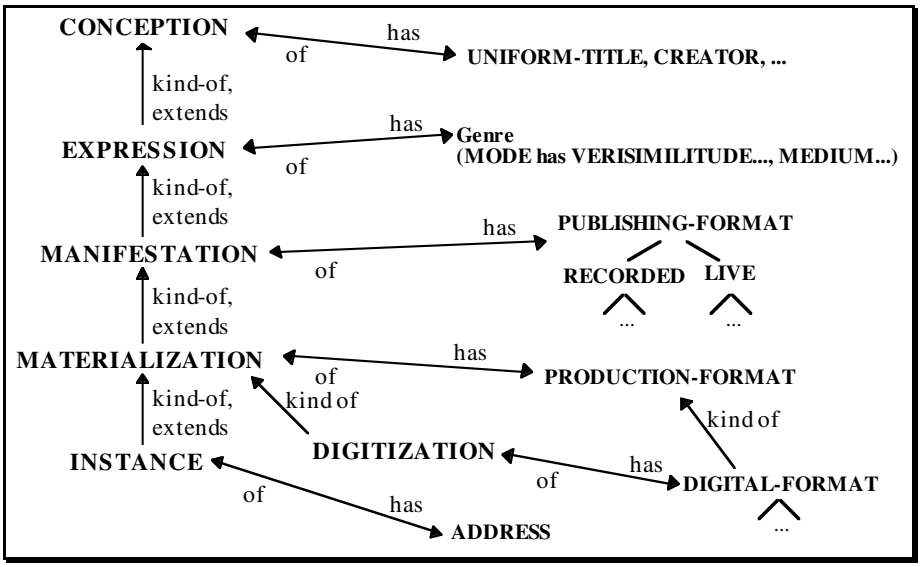

Figure 1: Attributes associated with the work hierarchy

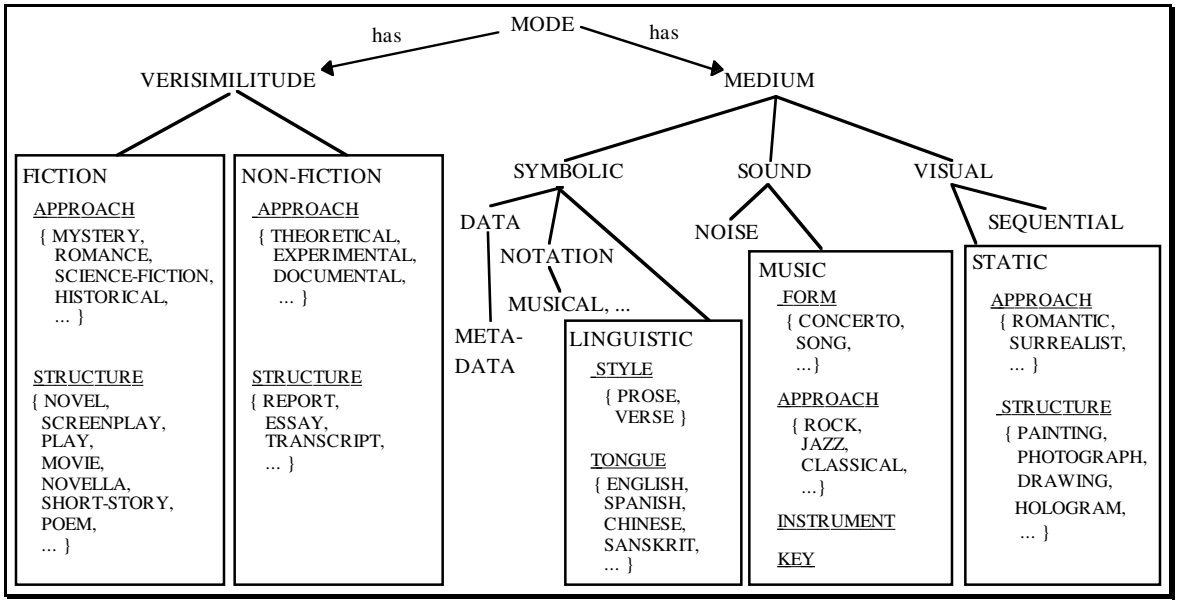

Figure 2: The structure of genre

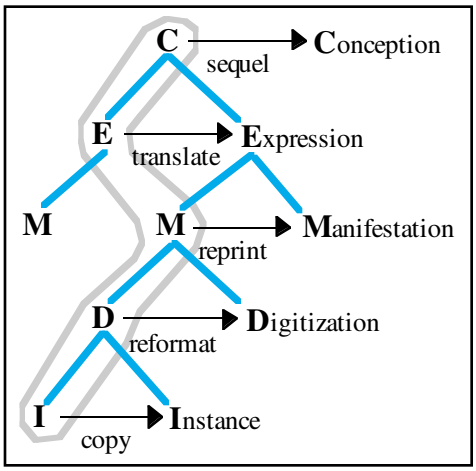

Figure 3: A work with five derivation relations

3 GENERATING A KNOWLEDGE BASE OF METADATA

We started to generate a knowledge base of metadata by requesting a sample of 500 US MARC records from the University of Michigan library. All of the records are for works either by or about Beethoven. The idea was to provide data with many works related to others in the sample, and also a variety of publishing formats and genres. The actual sample contains 493 records, out of a total of almost five thousand in the university library related to Beethoven.

Generating the knowledge base involves mapping data from MARC to the ontology, and reasoning about the data to identify relations between works. More specifically, we generate the knowledge base in four steps:

1. Convert binary MARC data to text tagged with ontology concepts.

2. Extract coded attributes and values from natural-language comments in the tagged text.

3. Convert the tagged text to Loom assertions. At this stage every value is treated as a distinct object.

4. Reason about the data to establish explicit relations between works.

We convert MARC fields and values to tagged text with a set of four control files that describe the MARC format and its mapping to the UMDL ontology. The primary control file maps each selected MARC field and code to one or more ontology concepts. Some mappings are conditioned on the MARC value, to permit a single field to be mapped to multiple ontology concepts. The second control file establishes priorities for situations in which multiple MARC fields map to the same ontology concept. The third control file identifies the location of codes within the MARC fields, contingent on MARC record type and bibliographic level. The fourth control file includes information for each code, including its length and the location of the code-value table.

For the Beethoven knowledge base, the extraction of coded attributes and values from MARC notes was done by a human—but in a rapid, mechanical way that could also be achieved computationally with a lexicon and some natural-language processing. For example, the 245c value "compiled and edited by H. C. Robbins Landon" translates into "RE H. C. Robbins Landon" ("RE" is a keystroke-saving code for "role, editor"). We spent an average of only forty seconds per record on this transcription. With less than 500 records, it was simply easier to do this by hand.

Converting the tagged text to Loom assertions (TELL statements) is trivial, since we postpone all reasoning until the data is in Loom. Every MARC record is assigned an ITEM-NUMBER associated with a MATERIALIZATION (we are dealing only with metadata and so do not include the INSTANCE level). A fifth control file provides the paths that link each of the destination ontology concepts to MATERIALIZATION. For each step along a path, a separate Loom instance is created, with the MARC value (if there is one) linked to the instance of the destination concept. Thus, after execution of the TELL statements, the knowledge base contains metadata where every value assertion is separate. For example, every ITEM-NUMBER has its own MATERIALIZATION, MANIFESTATION, EXPRESSION and CONCEPTION. Furthermore, single MATERIALIZATIONs are frequently linked to multiple instances of intermediate concepts. For example, a concerto would start with two instances each of MODE and MUSIC to separately connect its MUSICAL-KEY and MUSICAL-INSTRUMENT to its EXPRESSION.

Stage 4 is by far the most interesting. Here, we reason about relations implicit in the MARC data to make them explicit. Our treatment, however, can only be considered a crude first pass. There is no end to the effort that might be invested in developing increasingly refined rules to glean additional information. See Tillet [18] for an interesting review of how relationships between works are represented in past and current catalogs.

Frequently, partial or imprecise data suggests relationships that cannot be confirmed. In these cases we assert relations that are considered to be "tentative". By definition, "certain" relations that are not true are errors; false "tentative" relations are merely false. Instances are merged only if their common identity is certain. Otherwise, a "tentatively-the-same" relation is asserted to link them.

We are liberal with "tentative" relations, but conservative with "certain" relations. Thus, many tentative relations are incorrect. The result of any deduction that uses a tentative relation is also tentative.

Our processing includes the following steps:

a) Merge multiple instances of intermediate concepts with unification. The unification algorithm uses filler-cardinality constraints that are part of the ontology definitions. If an instance's asserted type is subsumed by another's, and multiple values are permitted, then the instances are merged.

b) Identify and merge shared CONCEPTIONs. CONCEPTIONs with the same UNIFORM-TITLE, CREATOR, and also SEQUENTIAL-NUMBER for numbered musical works are considered to be the same. When the UNIFORM-TITLE is missing, we use the MANIFESTATION's TITLE instead. Differences in punctuation or case are ignored.

c) Identify and merge EXPRESSIONs that have the same CONCEPTION, if all genre and associated CREATOR values match, or if a manifestation-level certain derivation is recognized.

d) Identify and merge MANIFESTATIONs that have the same EXPRESSION, if the PUBLISHER and

PUBLICATION-DATE match, and other PUBLISHING-FORMAT values of one instance are a subset of those of the other instance. Also merge MANIFESTATIONs if a materialization-level certain derivation is recognized.

e) Identify CONCEPTIONs that are "tentatively-the-same" using the same rule described in (b) above, but ignoring missing CREATOR names, and using a permissive matching algorithm for comparing titles.

f) Identify EXPRESSIONs that are tentatively-the-same as in (c), with either matching genre or CREATORs, but not both.

g) Identify MANIFESTATIONs that are tentatively-the-same as in (d), but rather than a shared EXPRESSION, require only EXPRESSIONs that are tentatively-the-same, or a shared CONCEPTION.

h) Establish relations to preceding works based on MARC notes describing derivation relations. If no predecessor is found in the work-family, generate metadata that describes what is known about the predecessor by sharing all levels of the work hierarchy above the level of the derivation.

   There are 19 types of relations transcribed from the MARC notes, mapped to eight relations in the ontology. Values transcribed from the notes of the successor work are compared to ordinary values describing candidate predecessor works. For each type of relation, a subset of the following attributes are compared: author, editor, language, title, publisher, city, date, and publisher-item-number. "Points" are awarded for a match for each value, depending on the relation. A match to a candidate predecessor is considered certain or tentative if points are accumulated above respective thresholds.

i) Establish certain and tentative reproduction relations on the MATERIALIZATION level, triggered by a FACSIMILE production format or an "original date" in the MARC publication-date code.

We generated the Beethoven knowledge base on a Sun SparcStation 20. Stages 1 and 3 are written in C++, and completed in a few seconds. Stage 4 is written in Lisp using Loom, and required several hours. The binary MARC data required 0.5 MB of disk. The Lisp image of the resulting knowledge base requires 35 MB (this is not a space-efficient format), but loads for execution in half a minute.

Table 2 summarizes the results. After initial loading of the data there is one instance on each level of the work hierarchy for each MARC record. The "Merged" column shows the results of steps (a) through (d), above. Steps (e) through (g) do not affect the number of work instances. The "New" column shows the number of predecessors generated to represent works outside of the collection in steps (h) and (i). The "All" column includes empty instances at the lower levels, generated to complete the work family of new predecessor instances created at higher levels.

Table 2: Number of work-hierarchy instances

	Raw	Merged	New	All
Conceptions	493	286	1	287
Expressions	493	473	128	602
Manifestations	493	490	26	645
Materializations	493	493	58	706

4 BROWSING THE KNOWLEDGE BASE

We implemented a Java interface, available on the Web, to help library researchers develop a feeling for the structure of ontology-based metadata (see footnote 1). The initial window lets the user select a work by identifying a type restriction from a list of genre and publishing format attributes: autobiographies, concertos, sound recordings, and so on. The user can also specify a keyword to be included in one of several fields, including creator, title, topic, and others such as editor.

The second window supports browsing work families by navigating relations that connect the works. All metadata associated with the selected "current" work is displayed in a text area on the left side of the window. In the middle are the relations that link that work to others. Pick a relation, then select from the list of works related in that way. All metadata for the related work is then displayed on the right side of the window. The related work can be made the current work, and the process repeats.

Users in various institutional contexts may potentially contribute to the knowledge base while they are using it. To suggest this possibility, our demonstration interface provides a button that lets a user "confirm" a tentative relationship between two displayed works. We consider "confirmed" to be an intermediate level of certainty, between "tentative" and "certain".

5 DISCUSSION

This section places our current research in perspective by discussing work that remains to be done. The issues considered are ordered by increasing scope and complexity.

First, the ontology contains many attributes that do not have corresponding MARC data. Many of them could be deduced, however, from other values. For example, we might deduce BOOK given a BIOGRAPHY without a SOUND or VISUAL medium. This kind of reasoning could be most vigorously pursued with default (non-monotonic) logic. Monotonic reasoning can also work, however, if applied conservatively.

Second, we have not yet implemented Whole-Part relations. Metadata for musical works will benefit greatly from Whole-Part relations, since these works often include multiple pieces that can also be considered works themselves. Figure 4 illustrates our design for Whole-Part

relations with a hypothetical scenario involving a textbook for teaching C++. The second edition adds a diskette containing sample programs. CONTAINS relations are established at each level of the work hierarchy, starting at the first level for which the contained work is considered to be independent relative to the original work. Links at each level are required to avoid ambiguity about the contents of the original work if, subsequently, new works are created that are derived from the contained work.

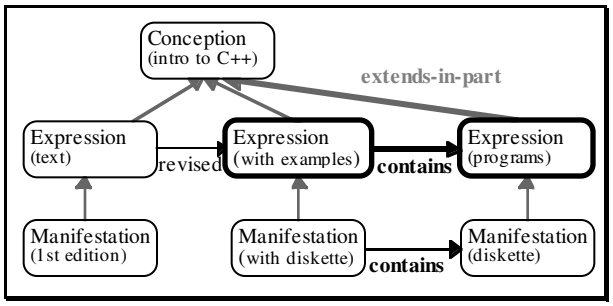

Figure 4: Adding a Part to a Whole

The ontological model changes the nature of the venerable debate about when a work requires its own "main entry" [25]. Here, a subjective decision may be required to determine the level of the work hierarchy where the new work deserves an independent instance not shared with the original work. For example, in Figure 4, if the programs already existed as examples in the original text, then they would share the EXPRESSION with the whole work, but would have a separate MANIFESTATION. Instead, the figure is drawn with a separate EXPRESSION, consistent with a judgment that the sample programs were first created for the second edition. The ontology refines the issue by distinguishing among five levels of work description rather than between two. The ontology also changes the practical context of the decision: To some extent, the choice is determined by the attributes associated with each level. For example, if the successor work does not share its genre with the predecessor, then it cannot share its EXPRESSION either.

Third, a thorough solution to the "versioning problem" will require experimenting with the implementation and use of the knowledge base. The versioning problem derives from the ease of modifying many electronic documents, which results in a proliferation of very similar works. A new version creates redundant metadata, depending on the type of modification. If a new version requires a new DIGITIZATION, then very little redundancy in metadata occurs; but a minor change in the content of the document requires a new EXPRESSION, and all of the lower levels of the work hierarchy are duplicated. The consequences of this redundancy are not dire. Redundancy makes updates difficult, but most bibliographic data do not change frequently. Uncontrolled redundancy (as opposed to intentional replication) wastes disk space, but storage requirements for text are inconsequential compared to multimedia. Nevertheless, to minimize software complexity it might be desirable to implement version-controlled metadata instances: i.e., store changes from one version to another, rather than replicating information that stays the same.

Note that the work hierarchy imposes a dimensional ordering that we could avoid with description logic, thus reducing redundant metadata. For example, Figure 5 defines EXPRESSION, MANIFESTATION and DIGITIZATION as complementary subconcepts of CONCEPTION; INSTANCE inherits from all of them. In this model, a new EXPRESSION creates a new INSTANCE that shares the old MANIFESTATION and DIGITIZATION. We chose a work hierarchy instead of a less-structured model, however, because we believe it will be more useful for supporting user inquiry.

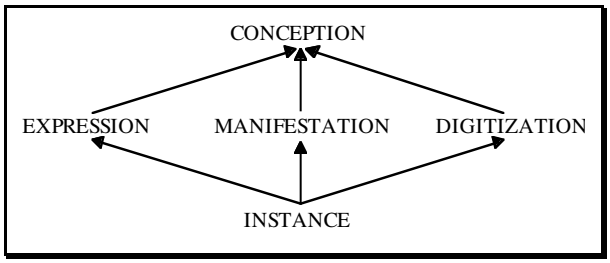

Figure 5: An alternative structure for work

The most important issue related to versioning is the need to protect users: Users should see one work with many similar versions, not many similar versions of the same work. Typically, user interfaces will be responsible for summarizing metadata. Documents that change in an on-going manner can be stored in a version-controlled form, and described with a single EXPRESSION, MANIFESTATION, and MATERIALIZATION, associated with growing lists of versions, publications, and productions, respectively.

Fourth, before undertaking the production of ontology-based metadata on a large scale, it will be desirable to think very carefully about the objectives of the system, and the representation that could best support those objectives. The fundamental obstacle is the tradeoff between expressiveness (the ability to model complex relationships) and tractability (the efficiency of computations). Both expressiveness and tractability have many facets, making the selection of representation a very difficult decision that is highly sensitive to the definition of system objectives.

For generating ontology-based metadata, description logic is too expressive in some ways, and not enough in other ways. Description logic's greatest asset is its ability to do "automatic classification": the algorithmic placing of new concepts into their proper location in the subsumption taxonomy based on the structure of their definition. We did not use automatic classification to generate the knowledge base of metadata, nor is it likely to be important to support end-user queries. On the other hand, a formal way to reason about uncertainty would be very useful, and uncertainty is not handled well by description logic.

Deductive object-oriented databases seek to combine logic

with support for object-oriented structures. In general, these systems have less inferential power than description logic, but are good at deducing new individuals and relations [11], and might therefore be appropriate for a library catalog. This is a relatively new area of research, however, and no consensus on the best approach has yet emerged [12]. We also need to worry about uncertainty. Three categories of systems have been developed to reason with uncertainty, using probability, possibility, and non-monotonic logic, respectively (see Hunter [9] for a good overview). Unfortunately, none is entirely satisfactory. For now, some sort of ad-hoc compromise (such as adding "tentative" relations to description logic) is unavoidable.

Fifth, although we dedicated many hours of discussion to developing the central concepts, in many areas our ontology remains provisional. A more thorough ontology will facilitate internationalization. Global acquiescence to a single ontology, however, may not be necessary. We are currently pursuing research to support communication between computational agents despite local variation in ontologically defined terminology [21].

Sixth, we need studies of users engaged in inquiry to prove the utility of ontology-based metadata. This will require interfaces that more fully exploit the ability of the knowledge base to support queries. The interface now available on the Web (see Section 4) is designed for researchers, not end-users.

Finally, our big-picture goal is a knowledge base of information resources that is very large, well organized, and sustainable. The professional cataloging community is already hard-pressed to keep up with the flood of new information. Hence, we envision "knowledge sharing" among many kinds of users—publishers, researchers, students, and so on—that contribute to the knowledge base as they use it. Computational reasoning can be used to facilitate, control, and apply user contributions. Correspondingly, the role of the professional cataloging community will change, to become increasingly focused on quality control. Clearly, this development will require institutional as well as technical change.

6 SUMMARY

This paper presents a formal ontology of bibliographic relations. A hierarchy of five central concepts loosely models the creation of work. Metadata for fully instantiated works includes instances at each of these levels. Families of related works have a tree structure. Ontogenic relations—derivation, addition, sequential continuation, secondary description, and so on—are each constrained to connect instances at a particular level of the work hierarchy.

We believe that our ontology has great clarity compared to the IFLA proposal, from which we borrowed most heavily. Instead of pages of often confusing text, we define each level of the work hierarchy by precise association with attributes that are also ontologically defined.

To test our ontology and demonstrate feasibility, we generated a knowledge base of metadata from a sample of MARC records. This process is fully automatic, except for the extraction of attribute/value pairs from notes in a way that could be automated with a lexicon and some natural-language processing. The most interesting part of the generation process uses reasoning to identify and merge shared work instances, and to make implicit ontogenic relations explicit.

The resulting knowledge base differs from ordinary catalog metadata in several fundamental ways. The knowledge base includes works not held in the collection. Relations are considered to be "tentative" as well as "certain". Most importantly, bibliographic relations are explicit, and can thus be used to facilitate user inquiry and digital-library operation.

7 ACKNOWLEDGMENTS

Gene Alloway and Judy Ahronheim contributed substantially to the development of the UMDL ontology for digital-library content. Bill Birmingham provided valuable advice and encouragement. Eric Glover provided sound technical insight when it was most needed. This work was funded by the NSF/ARPA/NASA Digital Library Initiative under grant CERA IRI-9411287.

8 REFERENCES

1. W. P. Birmingham, K. M. Drabenstott, C. O. Frost, A. J. Warner, and K. Willis, "The University of Michigan Digital Library: This is Not Your Father's Library," presented at Digital Library '94, 1994.

2. A. Borgida and P. F. Patel-Schneider, "A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Language," *Journal of Artificial Intelligence Research*, vol. 1, pp. 277-308, 1994.

3. A. Farquhar, R. Fikes, and J. Rice, "The Ontolingua Server: a Tool for Collaborative Ontology Construction," Computer Science Department, Stanford University, Palo Alto, California, Technical Report KSL-TR-96-26, 1996.

4. M. R. Genesereth and R. Fikes, "Knowledge Interchange Format Version 3.0 Reference Manual," Computer Science Department, Stanford University, Palo Alto, California, Technical Report Logic-92-1, 1992.

5. R. Green, "The Design of a Relational Database for Large-Scale Bibliographic Retrieval," *Information Technology and Libraries*, vol. 15, pp. 207-221, 1996.

6. T. R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," Knowledge Systems Laboratory, Stanford University KSL-93-04, 1993.

7. N. Guarino, "Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration," in *(To appear)*, M. T. Pazienza, Ed.: Springer Verlag, 1997.

8. M. Heaney, "Object-Oriented Cataloging," *Information Technology and Libraries*, vol. 14, pp. 135-153, 1995.

9. A. Hunter, *Uncertainty in Information Systems*. London: McGraw-Hill, 1996.

10. International Federation of Library Associations (IFLA), "Study on the Functional Requirements for Bibliographic Records," Frankfurt am Main, Germany: Deutsche Bibliothek, 1996.

11. P.-T. Kandzia and C. Schlepphorst, "DOOD and DL - Do We Need an Integration?," presented at Intelligent Access to Heterogeneous Information Workshop at KRDB-97, Athens, Greece, 1997.

12. M. Kifer, "Deductive and Object Data Languages: A Quest for Integration," presented at Fourth Intl. Conf. on Deductive and Object-Oriented Databases, Singapore, 1995.

13. G. H. Leazer, "A Conceptual Plan for the Description and Control of Bibliographic Works," in *School of Library Service*. New York: Columbia University, 1993.

14. R. M. MacGregor, "The Evolving Technology of Classificiation-based Knowledge Representation Systems," in *Principles of Semantic Networks*, J. F. Sowa, Ed. San Mateo, California: Morgan Kauffman, 1991, pp. 385-400.

15. G. A. Miller, "WORDNET: An On-Line Lexical Database," *International Journal of Lexicography*, vol. 3-4, pp. 235-312, 1990.

16. Network Development and MARC Standards Office, "USMARC format for bibliographic data: including guidelines for content designation," Library of Congress, Cataloging Distribution Service, Washington D.C., USA 1994-.

17. B. B. Tillet, "Bibliographic Relationships: Toward a Conceptual Structure of Bibliographic Information used in Cataloging," in *Graduate School of Library and Information Science*: University of California, 1987.

18. B. B. Tillet, "The History of Linking Devices," *Library Resources and Technical Services*, vol. 36, pp. 23-31, 1992.

19. S. L. Vellucci, "Bibliographic Relationships," presented at International Conference on the Principles and Future Development of AACR, Toronto, Canada, 1997.

20. P. Weinstein and G. Alloway, "Seed Ontologies: growing digital libraries as distributed, intelligent systems," presented at Second ACM International Conference on Digital Libraries, Philadelphia, Pennsylvania, USA, 1997.

21. P. Weinstein and W. P. Birmingham, "Matching Requests for Agent Services with Differentiated Vocabulary", student abstract presented at the Fourteenth National Conference on Artificial Intelligence (AAAI-97), Providence, Rhode Island, 1997.

22. P. Weinstein and W. P. Birmingham, "Service Classification in a Proto-Organic Society of Agents," presented at Workshop on Artificial Intelligence for Digital Libraries, 15th International Joint Conference on Artificial Intelligence (IJCAI-97), 1997.

23. W. A. Woods, "Understanding Subsumption and Taxonomy: A Framework for Progress," in *Principles of Semantic Networks*, J. F. Sowa, Ed. San Mateo, California: Morgan Kaufmann Publishers, 1991, pp. 45-94.

24. W. A. Woods and J. G. Schmolze, "The KL-ONE Family," in *Semantic Networks in Artificial Intelligence*, F. Lehmann, Ed.: Pergamon Press, 1992.

25. M. M. Yee, "What is a Work? Part 1: The User and the Objects of the Catalog," *Cataloging and Classification Quarterly*, vol. 19, pp. 9-28, 1994.