Supporting the eXtensible Catalog through Metadata Design and Services

Jennifer Bowen
Co-Principal Investigator, eXtensible Catalog Project
University of Rochester, Rochester, NY
January 8, 2009

The eXtensible Catalog and its Goals

In a recent paper, I presented an overview of issues facing developers of applications that use metadata for library resources, and related these issues directly to the goals of the eXtensible Catalog (XC) Project. In addition to discussing the goals for the XC Project in that paper, I emphasized the importance of four things:

- An understanding of the metadata itself and a commitment to deriving as much value from it as possible
- A vision for the capabilities of future technology
- An understanding of the needs of current (and, where possible, future) library users, and
- A commitment to ensuring that lessons learned in this area inform the development of both future library systems and future metadata standards.

In the year since that paper was written, the eXtensible Catalog team has been using these priorities to guide our work with metadata. We have developed two software toolkits, which as of this writing are being completed in collaboration with XC Partner institutions. We are continuing to develop the remainder of the XC system during our current grant period, and plan to release the software under the open-source Apache license in July 2009.

The eXtensible Catalog (XC) is a unique set of software toolkits that is not directly comparable to either a traditional Integrated Library System (ILS) or a “next-generation” discovery interface that sits on top of an ILS. XC is similar to next-generation discovery interfaces in that it will offer a new user environment (actually more than one) to enable library users to discover library resources, and in that XC is designed to work alongside an ILS’s staff modules (e.g. acquisitions, cataloging, circulation). The XC user environments are similar to other open-source discovery interfaces in that libraries can use them out-of-the-box, or libraries can customize and build upon them to suit their own needs. XC differs from all other end-user discovery solutions, however, in that it will go well beyond providing this discovery layer:

XC will also provide a metadata infrastructure which will facilitate the reuse of metadata in any number of web applications and systems.

The XC Metadata Element Set and the Metadata Services Toolkit

The metadata infrastructure in XC will allow libraries to derive as much benefit as possible from their legacy catalog data, and position them to take steps to move beyond their current dependence on MARC-based systems. It will also facilitate the reuse of legacy catalog data alongside metadata that originates in other schemas.

Our metadata work is in two interrelated areas:

- The design of an element set for XC that can be represented as an XML Schema and also expressed as a Dublin Core Application Profile. The latter will ensure that our work is fully documented in a standard manner and facilitate the later reuse of XC metadata.²

- The development of a software component for XC called the Metadata Services Toolkit (MST), plus a series of metadata services that will work with the MST. These services will prepare existing library metadata to function effectively within a variety of web client environments. The XC Project is developing several services for the MST that will be included with the initial release of the XC software.

This paper focuses upon the work that we have undertaken during the previous year in these two areas. Specifically, it relates the decisions that we have made regarding metadata back to the functional requirements that we have developed for the XC Project itself.³

Use Cases for XC’s Metadata Infrastructure

The eXtensible Catalog’s metadata infrastructure has been designed to facilitate the functionality of the XC software, as described later in this paper. In addition, this infrastructure will support a variety of uses for libraries that go beyond the original purpose of the XC software. Several use cases for XC’s metadata infrastructure reveal potential applications for XC software components in a variety of situations:

- A library consortium implements the XC system to provide an integrated discovery environment for its members’ collections. The consortium uses the XC OAI Toolkit component to harvest

³ Detailed information about the eXtensible Catalog, including links to presentations about XC, is available via the XC project website. “The eXtensible Catalog,” http://www.extensiblecatalog.org/
metadata from each library’s ILS and then uses the XC Metadata Services Toolkit (MST) to process and aggregate the metadata into a single data store using the XC Schema. This XC-Schema data is then harvested and revealed to users using the XC Drupal Toolkit and XC NCIP Toolkit. User-generated metadata captured via the XC Drupal Toolkit is then redistributed for all consortium users.

- A library’s digital development group uses XC’s OAI Toolkit to harvest MARC metadata from its ILS, processes the metadata using the XC MST and then harvests the processed metadata from the MST for reuse within a locally-developed web-based application. The local web-based application has access to the library’s ILS circulation functionality through identifiers in harvested metadata records, which are used in combination with the XC NCIP Toolkit.

- A library systems department uses the XC OAI Toolkit and MST to manage the automatic harvesting and processing of metadata (including updated and deleted metadata) from its ILS into a MARC-based union catalog according to a regular schedule.

- The manager of an institutional repository uses the XC MST to harvest the repository’s metadata and process it so that it can be searched in a web application along with metadata that originated in other metadata schemas from other digital collections.

- A library uses the XC OAI Toolkit to convert its MARC data to MARCXML, and then the XC MST to perform batch-level clean-up of the MARCXML data to correct anomalies in its legacy data. It then converts the processed MARCXML back into raw MARC for loading back into the ILS. By using the MST, the library accomplishes this without the more extensive programming that would otherwise be needed to perform batch changes on its ILS data.

In situations such as these, XC’s metadata infrastructure will provide libraries with a new arena for batch-level metadata enrichment and reuse to supplement existing MARC-based cataloging tools. XC will also offer management tools for metadata in other XML-based schemas, including transformation from one schema to another, normalization of non-standard data, and authority control processing. Other XC metadata services will aggregate metadata from a variety of source repositories and enable metadata records that represent the same resource to be linked. Finally, XC will provide a platform for developing customized metadata services to suit a particular library’s needs.

---

XC’s metadata infrastructure will provide libraries with a new arena for batch-level metadata enrichment and reuse, and will provide a platform for developing customized metadata services to suit a particular library’s needs.

---

4Descriptions of the other XC software components (Drupal Toolkit, NCIP Toolkit, OAI Toolkit, LMS Toolkit and Developer’s Toolkit) can be found on the XC Project website. “The eXtensible Catalog,” http://www.extensiblecatalog.org/
Summary of Functional Requirements for XC metadata

1. Enable metadata to function within the XC Architecture
The architecture that underlies the eXtensible Catalog software will enable libraries to aggregate metadata from a variety of sources, manipulate and enrich that metadata, and then present it to users. These activities require a mechanism for moving metadata from one repository to another that is both standard and sustainable. The standard used within XC is OAI-PMH, the Open Archives Initiative Protocol for Metadata Harvesting.5 OAI-PMH has been adopted widely as the standard for harvesting metadata from digital metadata repositories and has the added advantage of facilitating the management of metadata through its tracking of a metadata record’s provenance in its OAI header.

Inspired by the design of a proposed NSDL Metadata Services Hub,6 XC is designed so that each of its software components (or “toolkits”), will contain its own OAI-harvestable metadata repository, plus an OAI harvester. The OAI-PMH protocol enables reliable synchronizations of XML metadata across a network. This will enable XC components not only to communicate effectively with each other, but also to communicate with non-XC harvesters and repositories that are OAI-PMH compliant. A description of the XC architecture follows below in the discussion of the Metadata Services Toolkit.

Enabling metadata to function in the XC architecture only requires the following:

a. The metadata must be expressed in XML
b. MST services that process metadata and make new metadata records available for harvesting must remain online to enable the XC network to synchronize when source records are updated.
c. Metadata used within the XC end-user environments must be in the XC Schema, as described later in this paper.

2. Enable metadata from a variety of schemas (including MARC) to coexist in the same user environments
In the ITAL paper cited above, Requirement 1 described the importance of acquiring and managing metadata from multiple sources: ILSs, digital repositories, licensed databases, etc. Requirement 2 in that paper specified that XC must handle multiple metadata schemas. The level of complexity needed for XC’s metadata aggregation functions has led us away from using any existing XML schema and toward defining our own schema, which is described later in this paper. Defining a set of data elements and creating an XML Schema to contain those elements for XC has also led us toward including elements from multiple schemas within the same metadata record, even to the point where the same or similar metadata content may coexist in two or more parallel elements within a single XC Schema record. This will enable us to create a rich, consistent metadata schema that we can use to populate XC’s faceted

6 Diane Hillmann, Stuart Sutton, and Jon Phipps, “NSDL Metadata Improvement and Augmentation Services,” 2007. This grant proposal to the National Science Foundation is not available publicly.
user-interface applications. We plan to document exactly which elements in the XC Schema come from which other schemas, and their purpose within the XC Schema, using a Dublin Core Application Profile.\(^7\)

3. **Retain as much richness as possible when migrating from MARC metadata**

Much of the original metadata to be used in XC comes from library cataloging data that currently is coded for use in a MARC environment, and we are striving to bring as much of this data’s richness into the XC environment as possible. This is not an easy task, because much of this metadata is not easily translated into the conventions used in Dublin Core Application Profiles or in the broader web environment. For example, much MARC data exists in coded MARC-specific vocabularies that are not approved for use by the Dublin Core community. Similarly-defined MARC data may also occur in different types of MARC tags, while data that is tagged very similarly within MARC may actually be defined in very different ways. Complicating factors such as these make the reuse of library cataloging data in other environments potentially difficult. Nevertheless, our goal to retain the richness of the original cataloging data underlies our decision not to simply discard a good portion of MARC data when we attempt to combine it with metadata from a less rich schema, such as Dublin Core.

While there will certainly be constraints on what we can do with this metadata in a multi-schema environment, our goal has been to “bring along” as much cataloging data as we can into the XC Schema, with the expectation that many institutions that use XC in the future will want to make use of it. We have made significant progress toward this in the definition of the MARCXML-to-XC Transformation Service, which maps over 100 MARC bibliographic tags to the XC Schema.

4. **Enable a faceted user interface, FRBR-informed resource grouping, authority control, and user-generated metadata within XC**

Because XC will contain its own resource discovery user interfaces, XC metadata must first and foremost support the functionality of its own user interfaces.

XC will provide a faceted user interface to enable users to navigate and narrow their search results or browse a collection. Some categories of metadata lend themselves to being presented as facets because of the relatively consistent nature of the metadata in a record. The XC software will allow libraries to configure which of these facets to display to users, so that a certain facet or facet value can be suppressed when sufficient metadata is not available to populate it. The labels for each facet and the actual definition of the facet will also be configurable, to enable libraries to adjust these to fit the expectations of their users. The list below is an approximation of what may ultimately be displayed to end users, and is not necessarily consistent with the names of the metadata elements described later in this document for use within the XC Schema.

Initially, XC will support facets and facet values based upon the categories of metadata elements listed below:

---

• **Author/Contributor**, with possible facet values for commonly-occurring roles, such as author, composer, performer, etc., when these roles can be determined by the system

• **Format**, using a combination of vocabularies from MARC coded values and the DCMI Type Vocabulary

• **Language** of the content of the resource

• **Date** of the resource

• **Subject**, including facet values for topic, genre, region, and classification scheme (Library of Congress or Dewey Decimal)

• **Type of content**, derived from various MARC coded values

• **Collection or repository** containing the resource

• **Resource availability**, such as whether or not the resource is known to be available online

Much of the XC functionality included within this goal will be supported through XC’s Metadata Services Toolkit and its services. A set of XC normalization service steps will process incoming metadata to prepare it to support the facets defined above. To facilitate FRBR-informed grouping of resources, incoming metadata will be parsed into a “FRBRized” XC Schema record by XC’s transformation services. Other planned XC metadata services will populate XC records with authority record identifiers so that these identifiers can be used by other XC metadata services and within the XC user environments. These services are also described below in conjunction with the XC Metadata Services Toolkit.

5. **Enable reuse of XC metadata by other applications**

Unlike most resource discovery applications that process metadata only for use in one particular application, XC’s Metadata Services Toolkit architecture will enable users to enrich and process metadata for use within any number of systems. This is an important step toward achieving our goal of metadata reuse.

While the design of the XC Schema is optimized for use with XC software, we are also taking appropriate steps to encourage reuse of all or part of the schema. Our goal is to minimize any ambiguity about interpreting XC metadata by following appropriate standards and clearly communicating our metadata practices. We plan to register the XC Schema and any local vocabularies that we develop using the SKOS registry developed for the National Science Digital Library, to ensure that the metadata is understandable both to metadata experts and to machines that process the metadata. This will set the stage for libraries to reuse some aspects of our schema, make improvements to the schema, and transform it to conform to new standards in the future.

---


XC Metadata Element Set

During the past year, the eXtensible Catalog Project Team worked with consultants Diane Hillmann (Syracuse University) and Jon Phipps (Cornell University) to define an application profile and associated XML metadata schema for the XC Project.

In designing an XML schema for the eXtensible Catalog, our goal is to enable the XC functionality described above using the type of metadata that libraries already have available to them, and to do this as efficiently as possible. The XC Schema that has resulted may, at first glance, seem unnecessarily complex and less than theoretically rigorous. It is important to keep in mind, however, that this schema is NOT intended for use by catalogers or metadata experts to create metadata records for new resources. It is intended as a container for metadata that originated in a variety of schemas, to enable that metadata to function within XC and to be shared with other systems (such as faceted-browsing applications using Apache SOLR) in an understandable manner. By “understandable,” we mean understandable to both automated applications and metadata experts, but not necessarily to end-users (at least not in its raw state!). An entire XC Schema record will not be displayed to end-users (just as MARC records are not displayed to users), so its complexity will be hidden behind the scenes.

It is also important to understand that the XC Schema represents a purely practical rather than a theoretical approach to using metadata. While much interesting work is taking place in the library community to develop theoretically consistent metadata frameworks, models and carriers, the process used to develop the XC Schema should not be confused with these start-over-from-scratch-and-do-it-right approaches. In some ways, the XC Schema represents exactly the opposite. For XC, we have attempted to manipulate and make use of the metadata in various schemas that is available to us, and to use this metadata in a way that it can be shared with others. While ours is not a theoretical approach in itself, we believe that the work that we are doing can inform the work of those who are developing more theoretical approaches to the future of library metadata. Our experiences should provide a reality check regarding the feasibility of retrofitting legacy metadata into a new theoretical framework. Much of the effort needed to retrofit library metadata for use in new applications alongside metadata that originated in other schemas is likely to require considerable experimentation. The XC Schema is designed to embrace and facilitate that experimentation.

The XC Schema draws very heavily upon the reuse of metadata elements from standards and schemas that are likely to be encountered by libraries in the near future. One initial challenge with this is that the

---

10 Two examples of these efforts are the RDF model proposed by Martha Yee for her own cataloging rules: Martha Yee, “RDF model document list,” February 18, 2008, http://myee.bol.ucla.edu/rdfmodelintro.html and Karen Coyle’s futureLib wiki that is intended to foster discussion of a “next generation data format” Karen Coyle, “Designing the future--Library Systems and Data Formats,” http://futurelib.pbwiki.com/
library metadata world is entering what will likely be an extended period of transition. *RDA: Resource Description and Access*,¹¹ the new content standard for library metadata, has not yet been published, although much is known about what the RDA Element Set will contain¹² and the complete draft of RDA has recently been made available for constituency review.¹³ Unfortunately, XC’s development timetable has not allowed us to wait for a completed RDA before designing a metadata solution for XC.

In addition to implementing RDA, it is also likely that libraries will continue to use or encounter metadata schemas that have some compatibility with Dublin Core. We have made the best of this situation by creating a cross-schema implementation environment using elements from both the RDA element set and from Dublin Core. The XC Schema attempts to honor the theoretical models for both RDA and Dublin Core, although this is made more complicated because the models behind these standards do not function at the same level, as outlined in the Dublin Core Community’s Singapore Framework.¹⁴ RDA is based upon FRBR¹⁵ and FRAD,¹⁶ which function as content standards to define the realm of data that RDA will address. On the other hand, Dublin Core is based upon the Dublin Core Abstract Model (DCAM),¹⁷ which functions at a deeper level as a domain standard. Our approach has been to attempt to use the underlying data model of DCAM to express the community model described by FRBR/FRAD.¹⁸ In some cases, our functional need for XC to map AACR2/MARC data has resulted in perpetuating some of the idiosyncrasies of that legacy data in newly-defined XC elements (and a structure to contain them) to facilitate problematic metadata mapping.

Our approach to using elements from Dublin Core and RDA, and to defining new XC-specific data elements, is described below.

¹⁸ A joint DCMI/RDA Task Group is working to reconcile aspects of these two data models. See the Task Group’s wikis at [http://dublincore.org/librarieswiki/DataModelMeeting](http://dublincore.org/librarieswiki/DataModelMeeting) and [http://dublincore.org/dcmirdataskgroup/](http://dublincore.org/dcmirdataskgroup/). I am grateful to Karen Coyle for helping me to understand and describe the differences between these data models.
RDA: Resource Description and Access

We have included over 20 RDA elements from the RDA Element Set within the XC Schema. These were chosen specifically to enable us to retain much of the data found in MARC data that simply is not included in Dublin Core. Some examples include:

- frequency (for serials), scale (for cartographic materials), and mode of issuance (serial vs. monograph, etc.). These and the other RDA elements included in the XC Schema represent properties of a resource that libraries expect to be able to express in their catalogs. The XC Schema also includes a subset of RDA roles that represent the most commonly occurring relationships between agents (persons, corporate bodies, families) and library resources.\(^{19}\) Using these roles will facilitate FRBR-based grouping of resources and navigation of result sets for users.

The influence of RDA can also be seen within the XC Schema in its FRBRized structure. The Schema contains levels that correspond to the FRBR Group 1 entities: work, expression, manifestation and item, as well as a level that corresponds to the MARC 21 Holdings Format (which will be discussed further, below). The placement of various elements within a particular level was informed by draft versions of the RDA Element Set and by the Registry of RDA Elements.\(^{20}\) We consider the FRBR placement decisions that we made for the XC Schema to be provisional, and will make any necessary changes to the relationship of elements within the FRBR Group 1 entities once the RDA Element Set has been finalized. Once the RDA metadata standard has been published, we plan to expand significantly the number of RDA elements used within XC.

Dublin Core

The XC Schema includes all terms or properties in the “dcterms” namespace,\(^{21}\) which is roughly equivalent to what was used to be referred to as Qualified Dublin Core. Because the RDA Element Set was still being developed while we were designing the XC Schema, we initially attempted to use many of the DC terms as alternatives to their RDA “equivalents,” such as creator, publisher, etc. This ended up to be an unsatisfactory solution because some of the attributes that we need in the XC Schema are not approved by the Dublin Core community for use with DC terms. We decided to continue to include all DC terms within the XC Schema anyway, in order to facilitate the mapping of incoming DC data into the XC Schema. This will also make it easier for XC to enable other systems to harvest metadata from XC in Simple Dublin Core, which is the minimum for transmitting metadata using the OAI-PMH protocol. To

\(^{19}\) The RDA roles included in the XC Schema are author, composer, compiler, artist, director, editor, illustrator, performer, producer, and translator.

\(^{20}\) “NSDL Registry: Supporting Metadata Interoperability.”

enable the attributes that we need within the XC Schema, we have defined XC-specific elements to parallel some Dublin Core terms, as described below.

In order to use Dublin Core elements within a FRBRized environment such as the XC Schema, we have determined which one (or more) of the four FRBR Group 1 entities that each DC term may describe. As with the RDA FRBR mapping decisions, these Dublin Core FRBR mapping decisions should be considered to be provisional.

**Newly-defined XC elements**

While our goal has been to use declared metadata elements from existing schemas whenever possible within the XC Schema, we have found it necessary to define a number of elements specific to the XC. These XC-specific elements are needed for one or more of the following reasons, to:

- Enable the use of attributes that are not supported by Dublin Core, as mentioned above. One important category of these DC-unsupported attributes will provide direct links between XC Schema records and MARCXML authority records. To enable authority control linking within XC, we have defined parallel XC elements for the following dcterms elements so that the XC elements can contain these attributes:
  - creator
  - relation
  - spatial
  - subject
  - temporal
  - type
  - contributor
  - isPartOf

We have also added a parallel identifier element to the DC element “identifier” for the XC schema that will allow XC to use attributes to designate a variety of identifiers (e.g. IAN, ISMN, Superintendent of Documents (SuDoc) Number) that are routinely recorded in 0XX fields within a MARC record.

- Facilitate additional functionality in the XC User Environments that is not possible using only DC and/or RDA metadata elements. One example is the addition of the role “thesisAdvisor,” which does not currently appear in the draft list of RDA roles. XC’s user research has identified a need in the academic community to track the network of scholarship between graduate students and their professors and mentors, and including the use of the role “thesisAdvisor” is therefore a very desirable enhancement to current resource discovery systems. We have requested that the RDA developers add this to the list of RDA
Fill other gaps within RDA and DC. For example, the RDA element set does not contain an element equivalent to expression title, which we believe will be necessary within XC to enable collocation at the expression level within XC’s user applications. Initially, we will populate this element with the same data used for a resource’s work title because of the difficulty of actually creating a title for an expression. In another example, the addition of the XC element “recordID” will allow XC to make use of identifiers assigned by a source repository and facilitate appropriate management of data from various systems. Administrative metadata such as this is not present in RDA because RDA is not itself a record format.

Provide places for metadata elements from MARC bibliographic and holding records that are otherwise not covered in RDA and DC. In this category are various MARC vocabularies to support facets for the content and carrier of a resource and a variety of XC elements to contain holdings and item metadata. Some of these elements may end up being problematic when we attempt to include them within a Dublin Core Application Profile because they may have been defined in MARC in a way that makes them incompatible with the Dublin Core Abstract Model. MARC Holdings data is particularly problematic in this regard, because it does not describe a well-defined entity. Not only does it contain a mix of metadata elements from a variety of FRBR Group 1 entities (as does the MARC bibliographic format) but it also includes other elements (such as summary holdings) that cannot be mapped directly to any FRBR group 1 entity.

The description set definitions for the XC Schema are currently expressed in tables that include the name (and namespace) of the element, its position within the FRBRized data structure, rationale for inclusion, and the presence of any attributes. This document is available via the XC Website. Examples of XC Schema records and Version 1.0 of the actual XC Schema document will also be made available via the XC website.

---

22This was apparently an intentional omission from both RDA and from the conceptual model, FRAD, (IFLA Working Group on Functional Requirements and Numbering of Authority Records (FRANAR), “Functional Requirements for Authority Data: A Conceptual Model,” http://www.ifla.org/VII/d4/franar-conceptual-model-2ndreview.pdf because of a desire to reduce the redundancy of data for title variants, Email correspondence to the author from Barbara Tillett, December 14, 2008.
Description of an XC Schema Record

A full XC Schema Record consists of linked description sets for related entities that correspond to the FRBR Group 1 entities, Work, Expression, Manifestation and Item, plus an additional level, “Holdings,” which appears between Manifestation and Item in the sequence of description sets. The FRBR-equivalent Work, Expression and Manifestation description sets will contain data that may originate from a MARC 21 Bibliographic record or from a bibliographic-equivalent record in another schema, such as Dublin Core. The FRBR-equivalent Item description set may be populated with item-level metadata from an ILS about a physical library resource (e.g. barcode, circulation status).

The description set for “Holdings” has no equivalent in FRBR. Within the XC Schema, “Holdings” is defined as an equivalent to the MARC 21 Holdings Format to contain data that may be mapped from MARC Holdings records from a library’s ILS or from MARC holdings data contained within a MARC bibliographic record. While “Holdings” does not correspond to a FRBR entity, it has been a useful convention in the MARC environment for grouping multiple items to display to users, especially for runs of serials or multivolume works, which can be presented using a Summary Holdings statement. It is unlikely that “Holdings” will be relevant to metadata from any other incoming schema for XC other than MARC, and perhaps once the library community has moved beyond mapping its MARC data to a FRBRized environment it will no longer be needed within XC.

![Diagram of XC Schema Record](image)

**Figure 1.** XC Schema Record

Each description set within an XC Schema record is defined using the XML element, “xc:entity”, which can contain both a system-supplied XC identifier and a type attribute. The five type attribute values defined for xc:entity are the four FRBR Group 1 entities (work, expression, manifestation, item) plus the
additional value, “holdings”. Each description set may contain all or a subset of the elements defined for use within it in the XC Schema. 25

The definition of each description set within the XC Schema includes one or more elements that allow “up or down” linking to other description sets within an XC Schema record, and which can be assigned and maintained by the XC Aggregation Service described later in this paper. These elements are not required by the XC Schema, but may be essential in an application that stores the FRBR entities within a database structure such as SOLR, which is used within the XC Drupal Toolkit. 26

A valid XC Schema record MUST contain at least one description set representing one of the five defined entities (work, expression, manifestation, holdings, item).

A valid XC Schema record MAY contain:

- All five description sets, one at each level
- More than one description set for a particular level, as long as internal “up or down” links are present to indicate clearly the relationships between the various description sets.

We have deliberately defined the XC Schema with few constraints on the structure of a record, to provide as much versatility as possible. We believe that this versatility will prove useful as we (and other developers) gain experience processing records that represent single FRBR entities and maintaining the appropriate links between these records. For example, the XC Aggregation Service will process XC Schema records that each contain only one FRBR entity, but will create large XC Schema records that represent an entire “bibliographic family”: one work and all of its related expressions, manifestations, etc. These large record groupings will facilitate work-level collocation in the XC user environments. On the other hand, when XC Schema records are harvested by other applications, it may be more useful for an XC metadata service to create XC Schema records that contain a single manifestation and all of its related works, expressions, holdings, etc. This latter practice would enable these XC Schema records to remain compatible with most other metadata schemas, such as MARC, Dublin Core, MODS, etc. and also compatible with the Dublin Core principle of “one to one”: i.e. a description can only describe one resource.

25 The XC Schema does not deal explicitly with the FRBR Group 2 and Group 3 entities because we plan to simply rely upon existing authority records that, in a sense, represent these entities. While this could be considered as only a partial implementation of FRBR, we anticipate that it will still enable us to provide the functionality that we need for the XC system, which is our primary goal. It should also provide a useful and informative interim step toward full implementation of all of the FRBR and FRAD entities.

26 We anticipate that the XC software will rely primarily upon “up” links that link a description set to the level “above” it within an XC Schema record (e.g. to link an expression to a work). However, we have defined both “up and down” links in the XC Schema because they are defined within the RDA Element Set.
XC Metadata Services Toolkit (MST) and Services

About the XC Metadata Services Toolkit

As described on the XC Website, the Metadata Services Toolkit consists of a core application (MST) plus a set of plug-in metadata services, each designed to process metadata and produce new records for a specific, targeted use. When the MST software is released in July 2009, it will include services to aggregate metadata from a variety of sources, transform MARC and Dublin Core metadata, provide authority control, FRBRize metadata, and ultimately prepare metadata to work efficiently in a faceted browsing interface. An API embedded in the MST, along with a developer’s guide, will allow third-party development of additional services that can then be shared with the open source community.

The Metadata Services Toolkit will include a web-based user interface that administrators and catalogers can use to monitor, debug, and configure the data processing steps integral to each of the metadata services. A faceted-browsing user interface will allow catalogers to interact with the data at each step of the processing. The MST will reveal the output of each service as an OAI-PMH repository, which will make it available for harvesting by other XC software components, as well as make it possible for the output to be harvested by other non-XC applications. This will enable the MST to be used with non-XC OAI applications as well as with other XC software.

The MST was formerly known as the Metadata Services Hub, and described as such in the ITAL paper mentioned at the beginning of this document. 27 It retains the previously described “hub” functionality while now offering greater flexibility in configuration. The new MST design allows an institution to install the MST in a number of different ways. The MST can be set up as a hub with separately installed services, or with the “hub and services in one.” These options will allow institutions the maximum number of options to build and host their own individual services using the “hub and services in one” option or, if they choose, to share them across a network. These options are shown below in Figure 2.

The following services will be included with the initial release of the Metadata Services Toolkit, and are described in more detail below:

- Normalization Services (both MARC and Dublin Core)
- Transformation Services (MARC to XC, and DC to XC)
- Authority Control Services
- Aggregation

Each software component of XC (e.g. OAI Toolkit, Drupal Toolkit, LMS Toolkit) will have its own metadata repository containing all of the records that have been created by that

---

component. Likewise, each XC Metadata Service will contain its own metadata repository. Under the second scenario in Figure 2, the MST will contain multiple metadata repositories, one for each metadata service contained within the MST itself.

All metadata that a service outputs will be kept within that service’s embedded repository. When metadata is processed through more than one service, intermediate records will be made available via OAI-PMH at each stage of the processing. For example, when a MARC record is modified in an ILS, the OAI Protocol will enable the XC architecture to propagate that change and reprocess each of the resulting records for each service.

While this multiple metadata redundancy may initially seem to be a drawback of the XC architecture, it is important to remember that data storage costs have become less of a concern. In contrast to the pace of ILS record loads and extracts involving large numbers of records, the harvesting of records by one portion of the XC system from another can be achieved at relatively high speeds. And, because XC’s multiple repositories are created and managed by the system itself (and not by humans), the model is quite low-maintenance.

The redundant storage of metadata in multiple repositories within XC provides important advantages because data can be retrieved at any point in the system and changed before it moves to another point. This provides enormous flexibility for the system by creating the potential for making changes to one
portion of the system without corrupting another, because all metadata can easily be regenerated from
the other repositories in the system. Because many aspects of XC’s use of metadata can be considered
transitional (as described below with regard to the XC Schema), XC is the ideal environment for
exploring what can be accomplished with library metadata in web environments as standards and
functional requirements change over time.

**Normalization Services**

The XC Team has completed designing, coding, and testing a Normalization Service for MARCXML
records that will facilitate XC user interface functionality, especially the display of coded data to users.
This service includes a series of configurable steps to clean up MARC data and transform MARC coded vocabularies into
their textual equivalents using lookup tables that can be displayed to users as facet values. The service normalizes
the following MARC bibliographic data:

- Values from Leader 06, 007 and 008 to enable faceting by format
- Mode of issuance (serial, monograph, etc.) from Leader 07
- Normalization of system control numbers (both internal and external) from 001, 003, and 035 to
  eliminate any incorrectly-formatted control numbers from an ILS
- Details about content, such as “fiction or non-fiction”, whether or not a resource is a thesis or
dissertation, and intended audience, from 008
- Date ranges from 008
- Languages from 008 and 041
- Elimination of data from 020 other than the ISBN, such as price
- Location data, if present in a MARC bibliographic record [Note: this step will also be applied to MARC Holding records]

Other steps in the MARCXML Normalization Service achieve the following results:

- Populate the uniform title element (130 or 240) from the bibliographic record, to facilitate
  FRBRization
- Add role designations for authors and composers for books and scores, respectively
- Deconstruction of Library of Congress Subject Headings to facilitate faceting and authority
  control validation
- Deconstruction of name/title headings to facilitate authority control validation

The MARC Normalization Service has been developed to ensure that the output data from the service is
valid MARCXML by placing normalized data that no longer fits in MARC coded fields into locally defined
9XX fields. This will maximize the usefulness of the service in a variety of situations, as described above
in the XC metadata Use Cases. Libraries can configure the creation of each 9XX field, and also add new
steps to the Normalization Service to achieve additional MARC data cleanup.
The initial definition of the MARCXML Normalization Service is available via the XC Website.\textsuperscript{28} During the next year of the project, the XC Team will also develop a normalization service for Dublin Core records based upon the NSDL “Safe” Transforms.\textsuperscript{29}

**Transformation Services**

The MARCXML-to-XC transformation service, as defined on the XC Website,\textsuperscript{30} parses both MARC bibliographic and holdings data to the FRBRized XC Schema. The specific mappings have been informed by several aspects of the RDA development effort, in particular the distribution of RDA elements among the four FRBR Group 1 entities\textsuperscript{31}, the mapping of RDA elements to MARC\textsuperscript{32}, and an Entity/Relationship diagram that has been developed for RDA.\textsuperscript{33} Our goal has been to model our mapping of MARC data to XC as closely as possible upon the likely way that MARC data may be mapped to RDA’s FRBRized record structure, to facilitate the eventual incorporation of the RDA Element Set into XC.

The XC MARCXML-to-XC Service maps over 100 MARCXML bibliographic fields for cataloging data created using library cataloging standards such as AACR2 to the FRBR-influenced XC Schema. Criteria for selecting MARC fields for mapping to the XC Schema include the following, which are listed roughly in order of the data’s importance to supporting the functionality of the XC System. Those MARC fields that fall into the following categories were given highest priority for mapping to the XC Schema:

1. Enable basic metadata services (e.g. identifiers to enable aggregation)
2. Enable a facet in an XC user interface, as outlined earlier in this paper
3. Facilitate FRBR grouping of search results or authority control functionality
4. Are candidates for indexing in a specialized keyword indexes (author/name, subject, title)
5. Provide text that is likely to be keyword searched by users (e.g. table of contents)
6. Provide information that is not coded, and presumably therefore understandable to users and that will facilitate their identification and selection of resources

The following criteria were used to identify MARC fields that were less likely to be needed within the XC Schema, at least at this time:

\textsuperscript{28} eXtensible Catalog, “XC MARCXML Normalization Service Documentation,” Pre-release Version 0.0, January 7, 2009. \url{http://www.extensiblecatalog.org/MARCXMLNormalizationDocumentation}

\textsuperscript{29} “NSDL "Safe" Transforms,” NSDL Metadata Primer, \url{http://docushare.lib.rochester.edu/docushare/dsweb/Get/Document-32324/safeXform.html}

\textsuperscript{30} eXtensible Catalog, “MARCXML to XC Schema Transformation Service Documentation,” Pre-release Version 0.0, January 7, 2009. \url{http://www.extensiblecatalog.org/XCMARXMLTransformationDocumentation}

\textsuperscript{31} Joint Steering Committee for Development of RDA, “RDA Element Analysis.” \url{http://www.collectionscanada.gc.ca/jsc/working2.html#rda-element}

\textsuperscript{32} “RDA | Constituency Review.” The MARC mappings are in Appendices D and E.

\textsuperscript{33} This document is not publicly available.
1. Data which does not serve #1-6 above and which is coded in a manner that would require additional metadata normalization to make it understandable to users, e.g. 043 (Geographic Area Code), 047 (Form of Musical Composition Code)
2. Information used primarily for metadata specialists but which users don’t need for resource discovery, e.g. 040 (Cataloging Source)
3. Fields that would require a metadata expert to manipulate using an ILS cataloging client or metadata editor in order to get them in a form that XC can use, e.g. 306 (Playing Time) with multiple values.
4. Fields that received an extremely low score in William Moen and Shawne Miksa’s study of the frequency of use of MARC fields, unless some other compelling reason existed for including this data within XC at this time.\(^\text{34}\)

We have also defined a Dublin Core-to-XC Transformation Service that will be coded and tested within the coming months. Because the XC Schema already contains all terms in the dcterms namespace, the DC-to-XC transformation service will simply map each DC term to the same term in the appropriate FRBR section of an XC Schema record, without mapping the terms to different metadata elements. The architecture of the MST will easily accommodate other specific transformation services that may be developed in the future to crosswalk from other XML schemas to the XC Schema, or to other XML schemas for use in other non-XC applications.

**Authority Control Services**

Within the next eight months, we plan to enable a basic authority control linking service for XC. Our initial service will create and maintain links between MARCXML bibliographic and authority records. The service will match MARCXML bibliographic records against local copies of MARCXML authority records from the Library of Congress (LC) authority files and populate the MARCXML bibliographic records with the recently-defined control subfield 0 (zero) containing the matching authority record control number. When these records are transformed to the XC Schema, the authority record control numbers will be set as attributes within the XML document. Both the authority-identifier-populated MARCXML records and the XC records with authority record attributes can then be used to enrich various user applications, including XC’s discovery applications.

One difficulty that we have encountered is using control subfield zero in cases where a heading in a MARC record contains within it multiple headings that could each be validated on their own. Name/title headings are one example of this, where the name portion of the heading may validate on its own. Another example is the various segments of a pre-coordinated Library of Congress Subject Heading string, where the initial heading and subdivisions may also validate separately. To begin to address these situations, we have defined steps within the XC MARC Normalization Service to split these

\(^{34}\) “MARC Content Designation Utilization » Results of Frequency Counts Analysis,”

http://www.mcdu.unt.edu/?p=43.
headings for validation purposes using locally defined 9XX fields in MARCXML, while also preserving the precoordinated LCSH string in the metadata record. This will enable the XC service to capture the authority record control numbers for each valid portion of a heading. After this authority linking is accomplished, however, much work will remain to explore the potential usefulness of authority links to portions of a heading, and to refine the matching process.

Ultimately, we plan to design a fully-functional authority validation service based upon code developed for the XC Project during Phase 1 of the XC Project by a group of senior software engineering students at the Rochester Institute of Technology (RIT). The RIT-developed code was able to ingest Dublin Core as well as MARC records and match names against local copies of LC authority records. It then presented matching results to a cataloger or metadata specialist based upon the probability that questionable matches were accurate. The XC development team will incorporate this RIT-developed software into XC and also write a service that will match XC Schema records to LC authority files. This last functionality will enable authority control functionality to be applied to metadata that originates from any component of the XC System.

The basic XC Authority Control Service will provide a framework for our longer-term vision to enable the validation of a variety of controlled vocabularies against existing registries and authority files and across a variety of schemas. Providing an initial linking of bibliographic and authority records through the XC Authority Control Service using system-independent identifiers is an important advancement over linking via string matching or system-specific identifiers, as is often done within an ILS. We hope that, by experimenting with these linking techniques, the XC Project will set the stage for additional identifier-based record linking in the future by demonstrating the advantages of this type of linking and also revealing possible difficulties with its implementation. 35

**MST Functionality and Aggregation Services**

**Record Matching and De-duping**

XC’s MST software will include functionality to manage the flow of records through the system, such as adding new records and removing deleted records. When the MST identifies an updated record or a new record that matches on an existing record (as described below), it will allow libraries to decide how to treat these records by offering the following configuration options:

1. Choosing one record to keep while discarding the other. The record to keep can be selected based upon such factors as provenance, number of elements in the record, or total record length.
2. Creating a merged record. This more-sophisticated XC functionality will compare the content of the metadata in each record.
3. Flagging the records for human review

---

35 For example, it would be possible to link directly to another authority file, such as the “Virtual International Authority File,” assuming that unique identifiers are available for each entity represented in that file.

http://www.oclc.org/research/projects/viaf/
The XC Aggregation Services will serve a variety of functions related to maintaining the relationships between records in the XC System. These relationships are of the following two types:

**Equivalence**
One component of the Aggregation Service will compare metadata records based upon the presence of a defined hierarchy of identifiers. Theoretically, this equivalence matching could take place at each FRBR level, using appropriate identifiers for that level. Unfortunately, the current lack of adequate identifiers for specific instances of the FRBR Group 1 entities (especially for specific works and expressions) is a significant problem. To enable maximum flexibility for solving this problem in the future, the XC aggregation service will allow users to change the hierarchy of the service’s matching sequence, to match on additional existing identifiers when the situation warrants it (e.g. matching on SuDoc numbers for a collection of government documents) or to add new identifiers as they are defined in the future.

The current configuration of the Equivalence matching service will focus upon matching at both the manifestation and work entity levels within XC schema records. At the manifestation level, the service will use both system-specific identifiers (such as LCCN and OCLC number) and standard identifiers (such as ISBN and ISSN) and then take one of the actions described above to avoid duplicate manifestation-level data within the system. At the work level, the equivalence service will use identifiers for work authority records supplied by the XC Authority Control Service to identify those XC work records that match on the same authority record. The presence of these work authority links will facilitate the collocation of resources that represent the same work.

**FRBR Group 1 Entity Linking**
The Aggregation Service will assign and insert unique identifiers ("up links") between the various description sets in an XC Schema record (Work, Expression, Manifestation, Holdings, Item) so that the description sets may be processed independently while retaining the relationships between the sets. For XC Schema records derived from ILS MARC records, these links will initially be made using existing identifiers at the bibliographic, holdings, and item levels. When links between MARC records change within an ILS (e.g. when a MARC Holdings record is relinked to a different MARC Bibliographic record), the MST will automatically relink the resulting XC records. This will enable XC’s user interface clients - and potentially other SOLR-based applications as well - to make use of records that represent a single FRBR Group 1 entity. This technology therefore has the potential to facilitate implementation of a relational/object-oriented database structure using the FRBR Group 1 entities, as described in Scenario 1 of the RDA Implementation Scenarios.37

---

36 While some of these authority records actually represent expressions rather than works, there is no reliable identifier within MARC authority record to identify which is which. This will likely be problematic for XC.
Next Steps for XC Metadata

Between now and the release of the XC Software in July 2009, we will continue to refine the definitions of the current XC Normalization and Transformation Services, and develop the XC Aggregation and Authority Control services. As this work progresses, we will make additional changes to these services and to the definition of the XC Schema as needed to support the functionality of the XC software.

We will also be working on completing the following tasks:

- Define additional XC Schema elements to contain user-generated metadata, informed by recently-completed XC user research activities
- Define additional XC Schema elements to incorporate metadata for licensed content, in particular for journal citation data
- Release Version 1.0 of the XC Metadata Schema via the XC Website
- Formally register all XC elements and vocabularies within a new XC namespace, using the NSDL metadata registry.
- Complete and publish the XC Application Profile, potentially using the Singapore Framework as defined by the Dublin Core community (currently in draft).  

Looking Beyond our Next Steps

The eXtensible Catalog Project provides a platform for enriching and transforming MARC metadata to make it usable in a variety of web environments. Because libraries have a huge investment in creating this metadata, it is imperative that we find ways to reuse as much of it as possible. The XC MARC Normalization and Transformation Services provide an excellent starting point for libraries to engage in this work. Additional work remains to make MARC Holdings data usable in a FRBRized environment, especially in cases where libraries have used Holdings data as a way to enable a single MARC bibliographic record to describe more than one version of a resource (the “single record technique,”), thus further complicating the mapping of such data. However, the work that we have accomplished to date in this area can already inform other efforts underway to design a new container for library metadata that may eventually replace MARC.

XC’s architecture, which enables “born MARC” metadata to be combined easily with other metadata schemas, should prove useful in learning how to optimize an environment for users when the metadata within the system is not totally consistent. Additional work to incorporate user-generated metadata and

---

metadata for licensed content (such as journal citations) is needed, as is additional user research to ensure that interfaces that present this variety of metadata to users are understandable and usable.

The XC Metadata Services Toolkit provides a platform that can also be used for experimentation and testing of new metadata standards and schemas. It can therefore become an invaluable tool for libraries as they become accustomed to these new standards, especially RDA. Because the XC Schema incorporates a subset of RDA, libraries can gain practical experience with the new standard once the XC software is released. We also hope that the lessons that we learn from mapping MARC data to our hybrid schema will inform the developers of RDA to alert them to possible RDA implementation issues.

Once RDA is ready to be implemented and MARC-to-RDA mappings have been optimized, we plan to enable XC to ingest RDA data and improve the XC Schema to be even more compatible with RDA than it already is. Should an XML schema for RDA be developed, the XC Metadata Services Toolkit will be well suited to handling RDA data with the development of additional MST services. Once RDA is adopted, developers of library software will need to learn how to manipulate linked records that represent FRBR entities and how to maintain the relationships between those entities when the records are updated. The MST, and in particular the MST’s Aggregation Service, can provide a way to accomplish this.

The library metadata environment is entering a period that will be characterized by significant change and uncertainty, and the eXtensible Catalog Project will provide a variety of useful tools to help the library community make informed decisions about the future. The XC Project Team welcomes feedback on our work and suggestions for future activities and partnerships related to metadata. Please contact us via the project website at www.eXtensibleCatalog.org.

Acknowledgments

Phase 1 of the eXtensible Catalog Project was funded through a grant from the Andrew W. Mellon Foundation and through contributions from XC’s partner institutions. My thanks to the following colleagues and partners for their comments on drafts of this paper: David Lindahl, Judi Briden, Eric Osisek, Diane Hillmann, Karen Coyle, Barbara Tillett, Jonathan Rochkind, K.G. Schneider, and Eric Lease Morgan.