

The Code4Lib Journal

ISSN 1940-5758

Issue 1, 2007-12-17

Table of Contents

Editorial Introduction — Issue 1.....	2
by Jonathan Rochkind, Coordinating Editor, Issue 1	
Beyond OPAC 2.0: Library Catalog as Versatile Discovery Platform.....	5
By Tito Sierra, Joseph Ryan, and Markus Wust	
Facet-Based Search and Navigation With LCSH: Problems and Opportunities.....	15
by Kelley McGrath	
The Rutgers Workflow Management System: Migrating a Digital Object Management Utility to Open Source.....	33
By Grace Agnew & Yang Yu	
Communicat: The Next Generation Catalog That Almost Was.....	48
by Ross Singer	
Connecting the Real to the Representational: Historical Demographic Data in the Town of Pullman, 1880-1940.....	56
by Andrew H. Bullen	
BOOK REVIEW: The Success of Open Source by Steven Weber.....	72
Weber, S. (2004).The Success of Open Source. Harvard University Press. ISBN: 0674012925 (COinS)	
COLUMN: 700 Dollars and a Dream : Take a Chance on Koha, There's Very Little to Lose.....	76
by BWS Johnson	

Editorial Introduction — Issue 1

by Jonathan Rochkind, Coordinating Editor, Issue 1

This is a decisive time for libraries. In the changing social and technological environment, libraries must adapt to fulfill their missions and satisfy their users. Library technology is acutely involved in this adaptation. Digital services, content and tools have become a part of nearly every aspect of library operations. The “digital library” is here—if you work in a library, you probably work in a digital library.

This mission of this journal is to cover “the intersection of libraries, technology, and the future.” We plan to provide practical information to help the library community envision and achieve our technological future, to bring libraries’ tradition of collaboration to bear on new challenges. We want the digital libraries of today to be transformed into the digital libraries of tomorrow, providing quality information while meeting new and changing needs. Rapid transformation has risks, but maintaining the status quo brings its own, greater, risks. Libraries must take a leading role beside their vendors in the technological innovation that must accompany this needed transformation.

The Code4Lib Community

One locus of pragmatic innovation has been the Code4Lib community. Inspired in part by the social dynamics of distributed open source projects, Code4Lib is an informal online social and professional network of library technologists, embodying values of transparency, cooperation, and pragmatic problem solving. The Code4Lib infrastructure includes a listserv, an irc channel, a blog aggregator, and an annual conference [1]. Code4Lib is a dynamic community which fosters collaboration and encourages the sharing of skills and ideas [2,3,4]. But paradoxically, this amorphous informality can make it hard for someone new to the field—or wanting to take a new look at the field—to find a comfortable entry point to the community and the resources it has to offer.

We hope the Code4Lib Journal can manifest the values that have been successful for the Code4Lib community, while providing increased access to the collective knowledge and experience held throughout our diverse professional networks and local organizations, increasing cross-pollination and collaboration among library technology innovators—and helping more people and organizations become innovators.

This Journal is an Experiment

The Code4Lib Journal project aspires to balance a variety of sometimes competing goals. We want to provide quality articles providing useful information and discussion on bringing library technology into the future. We want every article to be a useful intervention into our communities of practice. We value readability over formality, and hope to meet high standards for quality and utility. We’d like articles to have the technical detail for reproducibility, while still being accessible to readers at varying levels of technical expertise. At the same time we want to ensure an easy process for authors, letting authors share their important work and ideas with as few barriers as we can get away with. The Journal is intentionally edited rather than refereed, and we try to contribute editing advice to help authors improve their articles without aggravation. We are committed to the Journal’s free online availability, to increase its visibility and impact in addition to its accessibility. We want the immediacy of a blog, the usefulness of a professional conference, the reliable quality of a good scholarly journal, and the participatory nature of our online communities, all in one easy to read and easy to produce package.

And we are trying to accomplish all of that on a shoestring, with an all-volunteer editorial committee sharing management and editorial responsibilities in an informal, open, and pragmatic way as per the Code4Lib ethic. Our Coordinating Editor will rotate with every issue; I’ll soon be passing the baton to Eric Lease Morgan.

The Code4Lib Journal project is in that sense much like some of the technology projects many of us work on in our daily lives, balancing competing values and priorities with limited resources. And we’ve tackled this project the way we do

those projects, with a ‘can do’ spirit and an agile development approach—in other words, we’re making it up as we go along.

So how is the experiment working out? We think we’ve got a great first issue. This is due to the great work of our authors, and of the Editorial Committee. I am not alone among the Editorial Committee in discovering that inventing a journal—even one solely online which is intended to be relatively informal and agile—is more work than I personally expected. All of our authors and editorial staff deserve to be proud of what we’ve produced together through hard work [5]. But ultimately only the judgments and actions of you, our readers, can measure our success. If you think this first issue is evidence of a worthwhile endeavor, you can contribute to its future success.

How Can You Help?

You can read our articles, suggest them to others, and continue the discussions in your blogs, listservs, and in comments attached to the articles themselves. We want every article here to be part of an ongoing conversation towards cooperative innovation among libraries.

You can submit articles to us, and when you run into a colleague with an interesting project or idea, you can suggest that they submit articles to us. We’re happy to accept articles and proposals at any time; proposals for our third issue are due by Friday March 14th. We welcome anyone interested in participating in the operation of the journal to join our public discussion list for journal business [6]. At some point in the future, we will solicit more official members of the Editorial Committee, too.

We hope that this journal can be one more contribution to the developing culture of collaboration around library technology, and we welcome you to join in our experiment.

Code4Lib Journal Founding Editorial Committee

- [Carol Bean](#), North County Regional Library, (Palm Beach County Library System)
- [Jonathan Brinley](#), Ball State University Libraries
- [Edward Corrado](#), The College of New Jersey, corrado@tcnj.edu
- [Tom Keays](#), Syracuse University Library
- [Emily Lynema](#), North Carolina State University Libraries
- [Eric Lease Morgan](#), University Libraries of Notre Dame
- Ron Peterson, University of Delaware Library (ronp@udel.edu)
- [Jonathan Rochkind](#), Johns Hopkins Libraries
- [Jodi Schneider](#), Amherst College Library & Graduate School of Library and Information Science at UIUC
- [Ken Varnum](#), University of Michigan

Notes

[1] <http://www.code4lib.org>

[2] Barrera, Antonio, Parmit Chilana, Kevin Clarke and Michael Giarlo (2007). 2007 Code4Lib Conference Report. *Library Hi Tech News* 24(6). pp. 4-7. <http://eprints.rclis.org/archive/00011670/>

[3] Frumkin, Jeremy and Dan Chudnov (2006). Code4Lib 2006. *Ariadne* Issue 47, April 2006. <http://www.ariadne.ac.uk/issue47/code4lib-2006-rpt/>

[4] Chudnov, Daniel (2007). code4libcon Shows What a Participatory Conference Looks Like. *Computers in Libraries* 27(5), May 2007. pp. 37-40. ([COinS](#))

[5] Special thanks to Jonathan Brinley for providing the nuts-and-bolts web management that many of us wanted to leave at our day jobs. He deserves the credit for the clean look and useful functionality of the site.

[6] <http://groups.google.com/group/c4lj-discuss/>

Beyond OPAC 2.0: Library Catalog as Versatile Discovery Platform

By [Tito Sierra](#), [Joseph Ryan](#), and [Markus Wust](#)

Introduction

Many libraries, with the goal of modernizing their web presence, are racing to deploy a “next generation catalog.” Next generation catalog applications typically offer a mix of these features: faceted navigation, keyword searching, relevance-ranked search results, “did you mean?”-style search revisions, item recommendations, RSS feeds, and mechanisms to collect and display user feedback [[1\(COinS\)](#)]. These “OPAC 2.0” efforts to replace or upgrade legacy OPACs with more powerful alternatives will no doubt improve the overall catalog experience for many library users. Unfortunately, the gains from these efforts are limited because a single catalog application cannot be optimized for all library users and uses.

Consider, for example, the two basic types of searches a user might perform in a library catalog: known-item and exploratory. In a known-item search the user typically has a specific item in mind; the goal is to acquire information about this item as quickly as possible. In an exploratory search the user has a topic in mind and the goal is to acquire a list of items related to the topic. An interface optimized for a known-item search would likely take advantage of the information the user knows about the item, thus emphasizing bibliographic metadata such as the item’s title or author. Optimizing for an exploratory search would likely emphasize descriptive metadata that items share in common, such as subject headings or user-defined tags. It would be challenging, if not impossible, to optimize a single library catalog application for both of these common use cases.

This conflict begs the question: why should the discovery of library collections be limited to a single catalog application? Given the resources required to keep library catalog data accurate and up-to-date, libraries ought to explore methods for integrating this data into relevant external applications. Lorcan Dempsey, of OCLC, refers to this process as “lifting out the catalog discovery experience.” He writes:

As we work to aggregate supply... so we must work to place these resources where they will best meet user needs. In this process, discovery of the catalogued collection will be increasingly disembedded, or lifted out, from the ILS system, and re-embedded in a variety of other contexts — and potentially changed in the process. [[2](#)]

Realizing Dempsey’s vision requires increased flexibility in how we provide programming interfaces to our catalog data. To enable creative use of catalog data and improve interoperability with external systems, we need to think beyond the application-specific enhancements that characterize current OPAC 2.0 efforts. One strategy is to think about our catalogs as a platform that can support many discovery applications, not just the OPAC. We believe this approach shows great promise for libraries looking to enhance long-term end-user discovery and use of library collections.

CatalogWS

Netscape co-founder Marc Andreessen’s definition of “platform” neatly describes the fundamental difference between platforms and applications:

... a “platform” is a system that can be reprogrammed and therefore customized by outside developers — users — and in that way, adapted to countless needs and niches that the platform’s original developers could not have possibly contemplated, much less had time to accommodate.

In contrast, an “application” is a system that cannot be reprogrammed by outside developers. It is a closed environment that does whatever its original developers intended it to do, and nothing more. [[3](#)]

In early 2007 the NCSU Libraries began to experiment with the “catalog as platform” approach through an initiative called CatalogWS.

The original designers of CatalogWS, Tito Sierra and Emily Lynema, were members of the team that implemented the first Endeca-powered OPAC [4] in January 2006. Following the release of the new OPAC, the implementation team faced a list of post-launch feature enhancements. Two of these enhancements were RSS feeds for catalog searches, and the integration of catalog search results into Quick Search [5], our library website search tool. Rather than modify the OPAC application, the designers decided it would be better to create a separate application programming interface (API) to the search indices used by the OPAC. Once in place, the API would make it easier to build the desired feature enhancements, as well as enable more versatile catalog data use in the future. After a few weeks of development, the CatalogWS API was born.

CatalogWS currently provides two functions: the “search” service and the “availability” service. The search service returns item and facet values for a given search query, whereas the availability service returns item availability information for a given ISBN. Documentation for both services is available on the CatalogWS project homepage [6]. To date, most of our CatalogWS-powered applications have used the search service.

The scope of the API is necessarily limited because it uses search indices, rather than querying the full set of data stored in the ILS. For example, the API does not expose all the data in the MARC record for our catalog items, since not all MARC fields are indexed by our search server. The designers decided to use search indices as a data source because they provide easy access to normalized data. Using search indices also made the data modeling process much easier because the indices already captured and consolidated the most useful catalog data for end-user application development. This convenience comes at the price of limiting the type of applications that can be built with the API. Since the API is read-only, we could not build tools for modifying bibliographic data or updating the inventory status of items in the catalog. These ILS-oriented functions were beyond the scope of what the designers intended to accomplish with the API.

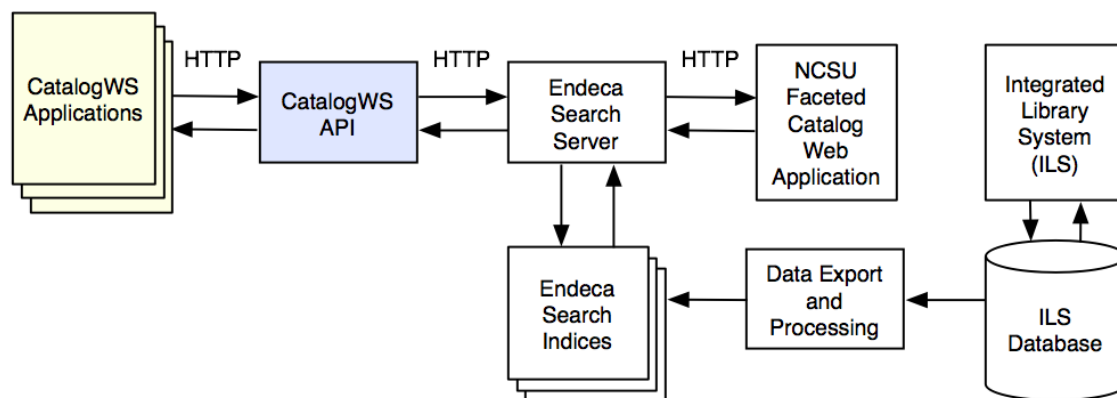


Figure 1: NCSU Libraries catalog architecture [[View full-size image](#)]

Despite the scope limitations, we have realized several benefits by using our existing Endeca-generated indices as a data source. Architecturally, search indices are designed to accommodate a high volume of low-latency requests. Building on search indices has also enabled us to include search-specific features in our applications, such as faceted data and spelling suggestions. The earlier Endeca implementation decision to include holdings information in the search indices proved to be very useful for the API. It means we are able to include inventory data in our applications, such as the name of the library that holds the item, the call number of the item, and even the current availability status of the item since the last index update.

The technical implementation of the API was modeled after the REST web APIs implemented by companies such as Yahoo! [7], Facebook [8], and Amazon.com [9]. Requests to the API are made through the HTTP GET protocol. Here is an example of a basic search request:

```
http://www.lib.ncsu.edu/catalogws/?service=search&query=usability
```

The search service supports many features one might expect from a search API, such as the ability to limit the number of results returned, and ability to change the sort order of returned results. By default, the service queries our catalog's "keyword anywhere" index, which looks for keyword matches across many fields in the catalog record. One can also search specific fields such as title, author, or ISBN/ISSN. It is even possible to supply a null keyword to browse the entire catalog. Including facet results in the response makes it possible to query specialized slices of the catalog using shared facet attributes, such as commonly shared subject headings, material formats, or author names.

CatalogWS responses are available in a variety of output formats, with a custom XML format as the default. The decision to define a custom base XML format for the search and availability services was motivated by several factors. First, the API designers believed it would be useful to include as much of the data from the search indices as possible into a single search request, rather than requiring multiple requests to the same data source. For example, the search service returns several categories of response data including search metadata (e.g. total results, spelling suggestions), bibliographic metadata (e.g. item title, author name), holdings data (e.g. library location, call number), and facet results. Although there are many existing standards such as MARC-XML, MODS, and Dublin Core that could describe some of this data in varying levels of granularity, no existing standard seemed suitable for such a heterogeneous collection of data. Second, the API was intended for use by developers at NCSU, rather than for programmatic external application consumption, so cost-benefit analysis of distilling the base API profile into one or more specific standard formats weighed in favor of a custom approach.

Below is an abbreviated XML response for the earlier example. A cached version of the full response is available [here](#).

```

<?xml version="1.0" encoding="UTF-8"?>
<searchResponse xmlns="http://www.lib.ncsu.edu/catalogws/1.0">
<requestUri>http://www2.lib.ncsu.edu:9921/catalogws/?service=search&query=usability</requestUri>
<catalogLink>http://www2.lib.ncsu.edu/catalog/?Ntt=usability&Ntk=Keyword&N=0&Nty=1</catalogLink>

<searchInfo>
<query>
<terms>usability</terms>
<key>Keyword</key>
</query>

<totalResults>477</totalResults>
<offset>0</offset>
<itemsPerPage>30</itemsPerPage>
</searchInfo>

<item id="1791483">
<catalogLink>http://catalog.lib.ncsu.edu/web2/tramp2.exe/do_ccl_search/guest?
setting_key=files&record_screen=record_brief.html&*search_button=keyword&servers=1home&index=ckey&query=1791483</
catalogLink>
<title>Cost-justifying usability : an update for an Internet age</title>

<pubDate>c2005.</pubDate>
<format>Book</format>
<isbn>0120958112</isbn>
<holdings institution="ncsu">

<library name="D.H. Hill Library">
<holdingsItem>
<callNumber>QA76.9 .U83 C67 2005</callNumber>
<location>Stacks (6th floor)</location>
<status>Available</status>

</holdingsItem>
</library>
</holdings>
</item>
<!-- repeating <item> elements-->

<facet id="format">
<title>Format</title>
<catalogLink>http://www2.lib.ncsu.edu/catalog/?Ntt=usability&Ntk=Keyword&N=0&Nty=1&Ne=200043#200043</
catalogLink>

<value>
<requestUri>http://www2.lib.ncsu.edu:9921/catalogws/?service=search&query=usability&N=206437</requestUri>
<catalogLink>http://www2.lib.ncsu.edu/catalog/?Ntt=usability&Ntk=Keyword&N=206437</catalogLink>

<title>Book</title>
<count>461</count>
</value>
<!-- repeating <value> elements -->

</facet>
<!-- repeating <facet> elements -->
</searchResponse>

```

Although the base XML format is custom, the CatalogWS search service has built-in support for returning results in several standard output formats. Search results can be returned in RSS 2.0, OpenSearch, and JSON formats. Additionally, the API supports an optional “style” parameter that enables requests to specify a path to an XSL stylesheet. When the style parameter is passed, CatalogWS provides server-side XSL transformation services for the request. The style parameter makes it possible for developers to create interactive CatalogWS-powered web applications using only XSL stylesheets.

Current Applications

Since going into production in January 2007, CatalogWS has provided the infrastructure for a variety of applications at NCSU Libraries. Some of the applications developed to date include specialized or experimental catalog interfaces, while others are innovative collection promotion display tools. Below, we provide a few examples of how we have used CatalogWS thus far.

To begin, we will discuss the MobiLIB catalog, an application designed for known-item searches in a mobile context [10]. Some example use cases for this application are looking up call numbers and checking the availability of items in the library bookstacks. Unlike our information-rich OPAC, the MobiLIB catalog is a barebones search tool that allows users to look up an item by keyword, title, author, or ISBN. Given the importance of item location and item availability implied by the mobile use context, the application emphasizes the display of inventory data, such as call numbers, and enables users to limit their search to items that are not checked out.



Figure 2: MobiLIB catalog [[View full-size image](#)]

Developing MobiLIB presented several challenges. The first was getting the application to function and display properly on a variety of mobile device platforms including handheld PCs, web-enabled cell phones, and PDAs. The smaller displays on these devices limit the amount of information that can be displayed in each view. Mobile devices also have

speed and cost-related bandwidth issues. These unique constraints forced us to rethink the catalog search experience from the ground up.

While the MobiLIB catalog is optimized for known-item searches, FacetBrowser is an experimental catalog interface that was designed to emphasize facets for use in exploratory searches [11]. FacetBrowser places facets persistently at the center of the application's interface. This design encourages users to browse the entire catalog at once, promoting use of facets as a way to explore or navigate the results set at every level. We believe this approach has the potential of exposing users to items in the catalog that they otherwise might not have discovered in a more conventional catalog search application.

FacetBrowser

Available Facets	Facet Grid		
Subject: Topic	Science fiction, English (60)	Science fiction (16)	Fantasy fiction, American (12)
Format	Fantastic fiction, American (11)	History and criticism (10)	Women (10)
Library	Women authors (8)	American fiction (8)	Short stories (8)
Subject: Region	Short stories, American (5)	Robots (5)	Authorship (4)
Subject: Era	Life on other planets (4)	Fantasy fiction, English (4)	Time travel (3)
Language	Androids (3)	Social life and customs (2)	Theory, etc (2)
Author	Horror tales, American (2)	Children's stories, American (2)	Space colonies (2)
New Titles	War stories, American (2)	Human-alien encounters (2)	Gay's writings, American (2)
Availability	Sports stories (2)	Science fiction, Australian (2)	Mathematics (1)
Library of Congress Classification	Jews (1)	Children (1)	Technology (1)

Applied Facets

Subject: Genre
Fiction

Subject: Topic
Science fiction, American

Start Over **Results: 388**

Maximum Number of Items: 10 | Sort By: Date Added | [Apply Selections To Search](#)

Search In Results

[Search In Results](#)
[Remove Search Term](#)

No Search Term Applied

Save Results

[Select All Items](#)
[Deselect All Items](#)
[Save Selected Items](#)

No Items Saved

	<input type="checkbox"/> The best of the best. Volume 2, 20 years of the best short science fiction novels
	<input type="checkbox"/> The collected stories of Robert Silverberg. Volume two, To the dark star 1962-1969. Silverberg, Robert.
	<input type="checkbox"/> The complete stories of Theodore Sturgeon. Sturgeon, Theodore.
	<input type="checkbox"/> Worlds enough & time : five tales of speculative fiction. Simmons, Dan.
	<input type="checkbox"/> Four novels of the 1960s. Dick, Philip K.

Figure 3: FacetBrowser [[View full-size image](#)]

Besides serving as an experimental catalog interface, FacetBrowser has also provided us with an opportunity to promote library collections in new ways. As part of our recently opened NCSU Libraries Learning Commons, several large screen

displays were installed in the new space to promote a variety of library services. Some of these displays are dedicated to showing “bookwalls,” which are displays of book cover images for a thematic collection of books from the catalog. FacetBrowser provides the means for library staff to quickly generate these bookwalls. The process involves applying one of several built-in output styles to an editorially selected list of items in FacetBrowser. The application automatically generates the HTML used to display the bookwalls on our large screen displays, enabling library staff with limited technical skills to promote specialized collections of books in a visually appealing way.

Start Over

7 Items Displayed | Style: Bookwall

Science Fiction Books

October 7

How to post the list:

- Open a new, empty document in your web editor. If you are using Dreamweaver, use the "Code" view.
- If your web editor has already inserted code, delete it.
- Cut-and-paste the following code into your document. This is all the code you will need.
- Save your file.
- Give the URL for your file to the E-board administrator.

Code:

```

<html>
<head>
<title>Science Fiction Books</title>
<link rel="stylesheet" type="text/css" href="http://www.lib.nyu.edu/facetbrowser/css/bookwall_static.css" />
</head>
<body>
<div id="topBanner">
<h1>Science Fiction Books</h1>
</div>
<div id="bookDisplay">



<img class="bookImage" src="http://images.amazon.com/images/P/1540159744.01._SCLIIIIIIIIII_.jpg" hspace="10" alt="Mad professor : the uncollected short stories" title="Mad professor : the uncollected

```

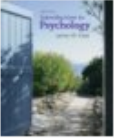
Figure 4: FacetBrowser bookwall output display [\[View full-size image\]](#)

In addition to bookwalls created by FacetBrowser, we use CatalogWS to promote library collections in other ways. The New Books Bookwall is similar to the staff created thematic bookwalls, except that the focus is on featuring books recently added to the collection [12]. This bookwall is different because the items it displays are not hand selected. Instead, a scheduled CatalogWS request retrieves the 300 newest items, with the goal of providing a peek at what's new at the library.

The New Faculty Books display is a variant on the New Books Bookwall [13]. This application focuses on featuring new books published by NCSU faculty. Although the user interface is slightly different, the underlying technical implementation is the same. The New Faculty Books display adds an additional filter constraint to items authored by NCSU faculty.

Recent Books Authored by NC State Faculty


Search:




[Introduction to psychology](#)
Kalat, James W.

[School of Design : the Kamphoefner years 1948-1973 : reflections and recollections](#)
Clark, Roger H.


[Park and recreation maintenance management](#)
Warren, Roger, 1932-




[Doing psychology experiments](#)
Martin, David W., 1943-



[The Longman writer's companion](#)
Anson, Christopher M., 1954-



[The Longman handbook for writers and readers](#)
Anson, Christopher M., 1954-



[Discourse, technology, and change](#)
Faber, Brenton D., 1968-

Figure 5: New Faculty Books list [\[View full-size image\]](#)

Closing Thoughts

Returning to Andreessen's definition of a "platform," we see that the final part of his definition is that a platform enables outside developers to adapt the system "to countless needs and niches that the platform's original developers could not have possibly contemplated, much less had time to accommodate." This observation mirrors the NCSU Libraries' experience perfectly. Although the original motivation for the API was to enable development of some OPAC feature enhancements, we have been able to use the API to create completely new applications that exist independently of the OPAC.

The platform approach has provided us with two distinct benefits. First, the API has reduced catalog application development time because it exposes data in an easy-to-use format. A second benefit is that our API has enabled versatile use of catalog data, as illustrated by the applications we have built so far.

As we reflect on our experience to date, we find that the collection of data exposed by our API could be more comprehensive. We encourage those who are developing a similar platform to cast a broad net when deciding what content to include in their API. Although our lightweight approach has worked well for us, we acknowledge the limitations of relying on search indices as a catalog data source. Those interested in learning from a more ambitious "catalog as platform" approach should look at the Talis Platform [14], which supports a wide range of interfaces and technologies.

Overall, we believe the time invested in developing our catalog platform was worth the effort and should continue to pay dividends in the near future. We never anticipated that CatalogWS would serve as a basis for such a broad range of applications. In our organization, it has fostered a culture of experimentation and enabled a larger group of staff to work with library catalog data in new and interesting ways. Although we are in the early stages of working with our platform, we recognize there is a large amount of untapped potential. We would like to see other institutions experiment with the platform approach and we welcome the opportunity to learn from other libraries about how to fully exploit the value of catalog data beyond the OPAC.

Notes

1. Marshall Breeding. Next-Generation Library Catalogs. Library Technology Reports 43(4). July/August 2007. ([COinS](#))
2. Lorcan Dempsey. The Library Catalogue in the New Discovery Environment: Some Thoughts. Ariadne 48. July 2006. <http://www.ariadne.ac.uk/issue48/dempsey/>
3. http://blog.pmarca.com/2007/06/analyzing_the_f.html
4. <http://www.lib.ncsu.edu/catalog/>
5. <http://www.lib.ncsu.edu/dli/projects/quicksearch/>
6. <http://www.lib.ncsu.edu/catalog/ws/>
7. <http://developer.yahoo.com/search/>
8. <http://wiki.developers.facebook.com/index.php/API>
9. <http://aws.amazon.com>
10. <http://www.lib.ncsu.edu/m/catalog/>
11. <http://www.lib.ncsu.edu/facethbrowser/>
12. <http://www.lib.ncsu.edu/display/bookwall/eboard.php>
13. <http://www.lib.ncsu.edu/display/facultybooks/list>
14. <http://www.talis.com/tdn/platform/>

About the Authors

[Tito Sierra](#) is Assistant Head for Digital Library Development at NCSU Libraries where he leads a small research and development team that builds new digital library applications.

[Joseph Ryan](#) is an NCSU Libraries Fellow in the Digital Library Initiatives and Administration departments.

[Markus Wust](#) is an NCSU Libraries Fellow in the Digital Library Initiatives and Special Collections departments.

Facet-Based Search and Navigation With LCSH: Problems and Opportunities

by Kelley McGrath

Introduction

Facet-based search and navigation interfaces are becoming increasingly popular on commercial websites, and several facet-based interfaces for library catalogs are now available. Many of these interfaces attempt to provide Web-style faceted interfaces to Library of Congress Subject Headings (LCSH) in order to provide options for browsing and for navigating large result sets.

LCSH was designed to deal with constraints that do not exist in the current computerized environment. In this paper, I examine the suitability of LCSH for presentation in an online facet-based interface. I look at some of the benefits and potential pitfalls of exposing current LCSH in this manner and ask what the implications of facet-based search are for LCSH.

Three types of solutions are proposed. Changes to:

- System design, which can be implemented with existing data
- LCSH practice or the rules for applying LCSH
- Structure of LCSH or data encoded in LCSH authority records

Ten problems are examined. By general type of problem, these are:

Hierarchical, broader/narrower relationships

1. [Top-level categories](#) (system design, LCSH structure, LCSH practice)
2. [Hierarchy and specificity](#) (LCSH structure) The lack of a consistent and complete hierarchy and the large number of top-level terms prevent LCSH from generating useful top-level categories for browsing without selecting a search term. The use of specific terms without explicit relationships to broader terms prevents easy collocation of classes of items (all the dog breeds or all the communicable diseases) or navigation to different levels of detail.
3. [Geographic/space facet](#) (system design, LCSH structure) Lack of hierarchical relationships for places hinders effective browsing of the geographic facet because users cannot adjust the level of focus to fit a particular inquiry, i.e., they cannot easily choose a broad geographic area for a topic on which the library has few materials or a narrow area for a topic on which there are many.

Access to cross-references

1. [Lead-in terms](#) (system design, LCSH structure) Lack of incorporation of cross-references in LCSH authority records into search results makes it more difficult for users to find the LCSH term for what they are looking for. Searches for terms in cross-references, such as WWII, do not pick up all the records with the related authorized LCSH term "World War, 1939-1945."

Forms of terms

1. [Single concepts, compound terms, factoring](#) (LCSH structure, system design) Use of terms, such as "Cookery, Japanese" or "Adult children of alcoholics, Writings of," that incorporate more than one facet or aspect of a concept reduce the power and flexibility of faceting by preventing users from limiting by the individual aspects separately.

Coextensivity

1. [Multi-work records](#) (LCSH structure, bibliographic record format) The inability to tell whether different LCSH strings on one bibliographic record represent different aspects of one topic or work, as opposed to separate topics or works leads to misleading results. If a user searches for waltzes and narrows the results by the subject heading “Violin and piano music,” he or she may get mismatched results, such as a CD that includes a minuet for violin and piano and a waltz for piano.

Missing information in facets

1. [Implied facets](#) (LCSH practice)
2. [Time facet, also covers coextensivity and machine-readable coding](#) (LCSH structure, LCSH practice)
Some facets are not included in LCSH strings because they were not necessary for identification or might lead to unhelpful splitting up of materials in a left-anchored browsing context. This is particularly true of chronological information. Users won’t know that these facets are incompletely populated and will make wrong assumptions about the comprehensiveness of some searches.

Semantic relationships between terms

1. [More specific categories of topical headings](#) (LCSH structure) More browsing options could be provided in faceted interfaces if subject headings were marked as belonging to certain groups or categories, such as types of terms (e.g., classes of persons) or terms that are inherently related to a certain discipline (e.g., legal topics).

Syntactic relationships between terms

1. [Facets as building blocks](#) (LCSH structure, system design) Implementing a larger number of explicit facets to express relationships between terms in LCSH would allow us to more flexibly manipulate and re-use information in subject strings without losing meaning.

Problems and Solutions

1. Top-Level Categories and Browsing without Searching

Problem:

One of the nice features of current faceted interfaces for library catalogs is the ability to browse without first choosing a search term. In this context, it is important to ask what makes something a good set of top-level categories for browsing. The psychologist Eleanor Rosch (1978) argues that people tend to think of things first of all as belonging to what she calls basic-level categories. That is, people are more like to initially categorize something at the level of “chair” than either “furniture” (superordinate) or “kitchen chair” (subordinate). There is probably some sort of similar “sweet spot” for initial browsing categories for library catalogs (although these might vary for public, academic, and specialized libraries) that would best agree with most users’ intuitions.

LCSH or any large system of subject headings are too narrow to be useful as top level vocabulary for browsing unless they can be mapped to higher level categories. However, as Hemmasi (1996) pointed out, there are “more than 2000 top terms related to the discipline of music terms or headings that are not attached to any broader concept,” which makes it difficult to create a concise list of terms for browsing. There would be a substantial amount of work needed to create the top-level categories and syndetic structure that would make LCSH useful as a basis for this kind of browsing.

Possible solution (system design):

The easiest alternative is to use something other than LCSH to generate top-level categories. Options include classification schemes, such as the Library of Congress Classification (LCC) or the Dewey Decimal Classification (DDC), or BISAC (Book Industry Standards and Communications subject headings, available at http://www.bisg.org/standards/bisac_subject/index.html), all of which could potentially be mapped to LCSH strings.

Possible solution (LCSH structure):

If the hierarchy of LCSH were made more comprehensive, all or most terms could be mapped to a small set of top-level categories.

Possible solution (LCSH practice):

In terms of improving top-level browsing access, there may be a place for broad-bucket headings for some types of materials, such as major literary forms. In NCSU's Endeca catalog, many searches result in a form facet like "\$v Fiction" or "\$v Poetry," which is actually misleading because these headings are unlikely to be present on all the fiction or poetry, only those (mostly newer records) that happen to have some topical heading subdivided by the form subdivision "\$v Fiction" or "\$v Poetry." Unfortunately, a naïve user could easily believe that narrowing a result set by the drama form facet would bring up all the dramatic works.

In our catalog, we have retrospectively added these sorts of broad genre-form headings for some literary forms, including fiction, poetry, and drama, where they could be easily identified and as we have had time. Fiction is the easiest to identify. Print/text fiction records (MARC leader type "a") include a fixed field called literary form in the MARC 008 that is coded "1# for fiction (<http://www.loc.gov/marc/bibliographic/ecbd008s.html#mrcb008b>) so a fiction genre heading can programmatically be added to all these records. Although there are additional codes for other literary forms, these were added to MARC21 relatively recently from UKMARC and are less commonly found in most records created in the U.S. If you would like additional advice on possible methods of identifying literary works, feel free to contact me directly.

In a faceted interface, ideally these literary forms would be searchable in combination with other characteristics, such as original language, date of composition, or nationality or other attributes of the author. Unfortunately, this data is often not found in existing bibliographic or authority records. Some information about original language and nationality and time period of the author could be extracted from classification numbers and coded language information in bibliographic records once the examples of literary works are reliably distinguished from criticism.

2. Hierarchy and Specificity versus Fragmentation

Problem:

Specificity of subject headings (i.e., a book about cats is entered under "Cats" and not "Mammals" or "Siamese cat") is potentially a great strength of LCSH, but in our current browse lists it often leads to a frustrating amount of fragmentation. In practice, the appropriate level of specificity to help a user is not fixed, but is related to the size and depth of the collection being searched, as well as the user's search terms and needs. Many subject browse lists in libraries are too specific for the collection or search and thus lead to fragmentation because the items the user wants are scattered among many subject headings, which are not easy for the user to identify and include in a single search.

In our current system, it is hard to move up and down levels of specificity because the syndetic structure is incomplete, the type of relationship is not always recorded clearly, and our systems don't support this type of navigation. For example, a user might want to search for everything about communicable diseases (broad) in Kenya (narrow) or AIDS (narrow) in Africa (broad), but for either of those searches, it's hard to see how a user could do a search in a current catalog that wouldn't be labor-intensive and involve manually looking for a lot of narrower terms and potential combinations. For example, a user would have to ascertain and manually combine Kenya with all the types of communicable diseases (e.g. sexually-transmitted diseases) and all the specific diseases (e.g., AIDS, malaria, tuberculosis), in order to do a comprehensive search.

Possible solution (LCSH structure):

If LCSH's syndetic structure were made more complete and systematic, and relevant relationships were coded in the authority records, interfaces could be designed to "explode" search terms to automatically include narrower terms or to shift to the next broader level. This is commonly done in interfaces using more rigorously designed thesauri (e.g., Medical Subject Headings (MeSH)). This relieves the burden on the searcher who wants to do a comprehensive search to find relevant things put under very specific headings without undermining the ability of another searcher to do a very precise search.

Faceted interfaces, in combination with better syndetic structure may particularly help users easily move up and down the chain of specificity to find what they need by allowing them to manipulate facets independently. This would make it easy for users to adjust the specificity of different aspects or facets of their search (e.g., a broad topic in a narrow geographic area) in a way that isn't possible in a linear pre-coordinated list.

The Flamenco Search Interface Project (UC Berkeley School of Information, n.d.) demonstrates some of the possibilities of a hierarchical, faceted interface. This works better when the top level categories and hierarchies are likely to be clear and predictable for users, but since users can use search terms to focus on specific topics, they can effectively start anywhere in the hierarchy.

3. Hierarchical Access to the Geographic/Space Facet

Problem:

It is difficult to search standard LCSH comprehensively for geographic areas because in many cases the form of the name of the area is given inconsistently. LCSH geographic terms are given in unabbreviated indirect order only when they are qualifying a base heading (e.g., "Mentally ill \$z Illinois \$z Chicago"). When a geographic term is used as the base heading, it is often qualified by the larger place (state or country) using the AACR2 abbreviation if there is one (e.g., "Chicago (Ill.) \$x Description and travel"). It is also difficult to search for a geographic area and all of the geographic areas within it (e.g., works that cover the U.S. as a whole or any part of the U.S.) because the broader term is not given explicitly in the heading (i.e., subjects for a book about birds in Wisconsin would not include the word United States even though Wisconsin is in the U.S. and the authority record does not include a broader term that could be leveraged to determine that Wisconsin is in the U.S.).

Possible solution (systems design, LCSH structure):

The approach that FAST takes provides hierarchical access to the geographic facet and enables the user to easily broaden or narrow the scope of a search. The FAST geographic facet is consistently given in unabbreviated, indirect order (e.g., Illinois \$z Chicago). Mapping the direct terms to the indirect terms and faceting all equivalent geographic terms in the same form increases precision and recall. It helps get all the mentally and terminally ill out of searches for "Illinois OR Ill" when looking for the state. FAST has also mapped the top-level terms in their indirect geographic headings to the MARC geographic area codes (<http://www.loc.gov/marc/geoareas>). These can be very useful for easily generating short searches that comprehensively search a whole continent or region. In the example of AIDS in Africa, the use of "f" in the geographic area code, combined with a few large regions defined as codes beginning with "f" (e.g., fb for Africa, Sub-Saharan, which is used directly in FAST rather than being entered as "Africa—Africa, Sub-Saharan; the latter, although redundant, might be better for faceting) would give comprehensive results that would be much more time consuming to duplicate by searching standard LCSH subject headings directly and looking for all the African countries individually. The indirect forms of geographic headings (field 781) and the mapping to hierarchical geographic area codes (field 043) that FAST does are largely encoded in current geographic name (151) authority records and could be leveraged by current systems.

4. Lead-In Terms and Incorporating Cross-References

Problem:

Most current catalog interfaces do not effectively integrate cross-references for authorized subject terms into their keyword search interfaces. This makes it harder for users to find the LCSH terms that correspond to what they are looking for.

Possible solution (system design):

This is something that could be remedied in existing facet-based interfaces by incorporating cross-references from subject authority records in keyword searches and by displaying matching cross-references in the list of relevant subject facets in some way that is clear to users. If a user searches a term that is in an LCSH cross-reference, such as “Non-violence,” it should show up in the list of relevant topical subject facets using some convention that makes in clear what’s going on. The Internet Movie Database (<http://www.imdb.com/>) displays authorized terms resulting from a keyword search as hyperlinks, with succeeding non-hyperlinked lines giving cross-references in italics, such as “aka ‘Larry Fishburne’” or “birth name ‘Clinton Eastwood Jr.’” Perhaps the cross-reference term searched by the user could be listed in the subject facet in parentheses following the authorized term. The list of subject facets could then display something like “Nonviolence (Non-violence)” to communicate the connection between the term the user searched and the authorized subject term.

For example, if a user searches for “non-violence” in NCSU’s Endeca catalog, there are 62 hits, 13 of which have the subject nonviolence as one word. Non-violence is a cross-reference on the authority record for nonviolence. If the search targeted subject cross-references as well as terms actually appearing in bibliographic records, the user’s search results would incorporate everything with the keyword “non-violence” plus everything with the subject “nonviolence,” thus expanding the result list to include the additional 136 records that have the subject nonviolence, but didn’t show up in the original search.

Without searching subject cross-references	With searching subject cross-references
62 records	198 records
Subject: Topic	

- Nonviolence (13)
- Gandhi, (10)
- Politics and government (8)
- History (7)
- Passive resistance (7)

Subject: Topic

- Nonviolence [Non-violence] (149)
- Gandhi, (??)
- Politics and government (??)
- History (??)
- Passive resistance (??)

Incorporating cross-references into searches in a facet-based environment might also be effective in helping with the problem mentioned on the Next Generation Catalogs for Libraries (NGC4Lib) email list in July 2007 about World War II in France and cross-references. If a user searches for “France World War II,” the system could look for each individual word as well as groups of words in both the authorized headings and cross-references in authority records and find the best matches (weighting things that occur in a phrase or occur together so “world war II” could point to “World War, 1939-1945#” and “France” could pick up subdivisions with variations on “France” and “French” (e.g., “\$x Aerial operations, French”). Cross-references for subject subdivisions generally do not work well in existing interfaces.

Problem:

Sometimes a broad topic (e.g., boxing in general) that a user may be interested in is entered under different subjects that emphasize particular aspects of that topic (e.g., “Boxers (Sports)” for a biography or “Boxing” for an instructional

video). This also happens when the same meaning is represented by different terms (e.g., “Church history \$y *Middle Ages*, 600-1500# and “Architecture, *Medieval*) and there are some other variant combinations that are very similar in meaning, such as “Older people \$x Psychology” and “Aging \$x Psychological aspects,” that might be missed by users interested in both.

Possible solution (LCSH structure):

Connections between the values in various facets (e.g., the action and the class of persons) related to a given topic or other types of cross-references could be made. This would require that these relationships be identified and recorded in subject authority records in some way and would improve recall for general searches on many topics.

Problem:

Some topical subdivisions vary in form, but not in essential meaning, depending on the type of base heading that they follow. For example, “\$x Economic conditions” is used after places (651), but “\$x Economic aspects” is used after topical base headings (650). In a faceted interface, these will appear to users as two different limiters.

Possible solution (system design or LCSH structure):

These pairs of subject subdivisions could be identified and merged for search and display purposes in a faceted interface.

5. Form of Terms in LCSH: Single Concepts, Compound Terms, and Factoring

Problem:

Vickery (1970) says that subclasses should be “derived from a parent class by application of a single characteristic from an array, and should as far as possible be mutually exclusive” (p. 39). In faceted thesauri (as opposed to notation-based classifications), these demands for step-by-step and clearly defined division are often in conflict with the desire to have phrases that make sense to users (e.g., “African American women poets” versus “African Americans” + “Women” + “Poets”). Svenonius (2003) notes that the issue of compound terms and single concepts in thesauri is often not simple and straightforward. Single concept does not necessarily equal single word. She asks if “information retrieval” is usefully considered as one concept or two. She also provides examples of typical situations in which it is recommended that compound terms, generally a modifier and noun, be retained.

The compound or complex nature of many individual topical terms in LCSH, such as “Cookery, Indic” and “Absurd (Philosophy) in literature” causes difficulties for the development of clean, consistent faceted interfaces because the component parts cannot be manipulated individually. The *Subject Cataloging Manual* (SCM H 180, point 15) lists a number of forms of headings, ranging from simple nouns (“Children”) to more problematic forms such as nouns with adjectives (“Gifted children”) or with prepositions (“Teachers of gifted children”).

Possible solution (LCSH structure):

Consistent use of broader terms, ideally with characteristics of division marked, could help compensate for the lack of faceting of compound terms. However, LCSH does not make machine-readable broader-narrower terms under all circumstances. For example, The *Subject Cataloging Manual* (SCM H 370) instructs catalogers to not create broader term references for “inverted headings qualified by names of languages, nationalities, ethnic groups, or terms that designate time periods, when the only appropriate BTs are the identical heading without the qualifier.” That means that “Art, Chinese” does not get a broader term of “Art.” Even where there is a broader term given in the authority record, there may be more than one characteristic of division used under that broader term and there is no machine-understandable way to differentiate between these. “Art, Chinese” is for all practical purposes a narrower term of art by nationality; “Art, Buddhist” is a narrower term by religious tradition; and “Art, Elizabethan” by a time period and/or style. There can also be a hierarchy of divisions of a certain type, e.g., “Sculpture” to “Metal sculpture” to “Bronze sculpture.” In a true faceted system, under art there would be a comprehensive list of types by nationality, by geographic region, by style, by time period, and so on.

Problem:

Compound headings, such as “African American women tennis players,” are often problematic when found on individual biographies. According to the Subject Cataloging Manual, such headings are supposed to be accompanied by a more general term. The instructions say to “assign headings that specify the sex or ethnic group of the class of persons, such as Women architects, African American lawyers, etc., if that is a significant aspect of the work. When assigning this heading, assign it *in addition to the unqualified heading for the class of persons*” ([SCM](#) H 1330). These instructions are often overlooked, leaving some biographies segregated out from the mass of American tennis players (at least in a browse list where “African American women tennis players” is not easily connected to “Tennis players \$z United States”), and certainly from the broader yet term “Athletes \$z United States.”

Possible solution (LCSH structure):

Factoring out the parts of a heading to their most specific parts (e.g., African Americans + women + tennis players), which could have broader and narrower terms for each individual bit, would be more flexible and would allow for any combination of terms desired.

When Yee ([2001](#)) compared genre terms from LCSH and MIGFG (the Moving Image Genre-form Guide, <http://www.loc.gov/r/mopic/migintro.html>), which is a faceted list, she pointed out that a disadvantage of single concept facets in a controlled vocabulary is that they can result in artificial terms (e.g. “Gangster–Feature” in MIGFG versus LCSH’s “Gangster films”) that don’t resemble users’ vocabulary. However, it is possible to map combinations of individual concepts to phrases that are more likely to be known to users. Alternatively, if the multi-concept headings are kept for display purposes, they could have explicitly coded broader terms, as well as explicitly coded characteristics of division (this would enable a user to get all the available types of art by nationality in a separate list, for example). In the long run, it would probably be better to create headings that are single-concept as much as possible and map them to relevant phrases when needed. This could be done by explicitly entering the phrases as cross-references for specific combinations of single-concept headings or, in many cases, through rules for order and singular/plural forms when combining certain types of terms, such as for classes of persons. There seem to be regular patterns to the phrase (or phrases) that are used for classes of persons (e.g., “African American women tennis players” not “Women African American tennis players” or “Tennis player African American women”) so rules could be developed for mapping combinations of single terms to sensible phrases if the types of single-concept terms were marked (e.g., classes of persons, ethnic groups, professions). Phrases that are not amenable to a rule-based approach could be established explicitly.

Possible solution (system design):

Tunkelang ([2006](#)) suggests an alternative approach for helping match users’ vocabulary to the system’s vocabulary. His Dynamic Category Sets approach is intended to effectively display combinations of facets that meet a user’s request. For example, a user query for “audio technology” could be mapped to both “media=audio, subject=technology” and “subject=audio technology, subject=history” (p. 2). This approach could also support stemming and cross-reference or synonym searching.

6. Coextensivity, Particularly with Multi-Work Records**Problem:**

Anderson and Hofmann ([2006](#)) make the point that, although LCSH is supposed to provide subject analysis that is coextensive with the topic of the work, this is undermined by the use of more than one heading to approximate the coextensive subject. There are many situations where it is not possible to construct a single subject heading string that brings together all the aspects of a single topic for a single work. One of the examples given in H 180 of the *Subject Cataloging Manual* is the use of the combination of “Ruminants \$x Metabolism” and “Lipids \$x Metabolism” for the title *Lipid Metabolism in Ruminant Animals*.

The difficulty in determining whether multiple subject headings represent aspects of a single topic or two different topics is exacerbated when there are multiple works on one bibliographic record. For example, a DVD containing two episodes

of the television program *The Long Search* might come up in a user's search for Buddhism in England in a faceted interface, even though it contains one episode about Buddhism in India and one about the Catholic Church in England. There is no way to tell in current records whether two subject headings represent different aspects of the same topic or two different topics covered in the same item.

The problem is particularly acute for musical sound recordings, which typically feature multiple pieces with no machine-interpretable relationship between the various types of metadata about a given piece (title, composer, performer(s), form or genre, instrumentation, date of composition, etc.). It is not uncommon for subject access for a given musical piece to be split between two subject headings in such a way that a computer cannot be taught to correlate instrumentation with genre, although users want to know this. It is hard to even make a good guess, since a given heading might be used in more than one combination. The following list of subject headings from one bibliographic record could theoretically include both a minuet for violin and piano and a minuet for solo piano.

- Waltzes.
- Violin and piano music.
- Minuets.
- Piano music.

Possible solutions (MARC practice and structure, LCSH structure):

Some sort of linking of the subject headings and/or relevant coded fields could help compensate for this problem. Effective ways of expressing hierarchy and relationships within individual bibliographic records are needed. McBride (2000) discusses the problem of relating the different aspects (or facets) of individual pieces on bibliographic records for musical sound recordings. He examines the potential of combining coded information in 04x fields with subject headings and other information using linking subfields (\$8) to create the appropriate connections. He discusses the development and potential use of coded information in fields such as 045 (date of composition), 047 (form of composition), and 048 (instrumentation) for faceted access to music. He points out that despite potential improvements to these fields, they provide fundamentally inaccurate results if some way cannot be found of tying together the related fields. It is unclear how the underlying problem in existing data can be remedied without review by human beings with the expertise to encode the connections between the various bits of information. I think of this as the Humpty Dumpty problem—all the pieces might be there, but it's impossible to get them back together again. Some sort of hierarchical XML record seems likely to outperform a typical MARC record in ease of use, both for input and manipulation of hierarchy and relationships within records. Although the MARC linking subfield (\$8) could be made to work for most of these connections, it fails in those cases where information about different works appears in a single field, such as 511 (performers) or 505 (contents note). Currently, neither interfaces to easily input these links nor user interfaces to get them back out are commonly available.

7. Implied Facets

Problem:

Facets are sometimes not explicitly included in LCSH strings when they are thought to be redundant for human viewers or unnecessary in a traditional left-anchored subject browse list. An example of an implied facet is the omission of geographic subdivision by "\$z Germany" under "National Socialism." According to the authority record (sh 85090131), "National socialism" should not be followed by Germany. "National socialism" = Nazism in general and Nazism in Germany as a whole, while "National socialism \$z Germany" is only supposed to be used when it's further subdivided by a more specific place e.g., "National socialism \$z Germany \$z Berlin"). When users search for Nazism in a faceted interface and then click the LCSH-based geographic facet "Germany" to limit to Nazism in Germany, they will not actually get all the works on Nazism in Germany, even though it may appear to the user that they should.

This conflict between the need for concise human-scannable strings versus the explicit notation required by computers is described by Slavic and Cordeiro (2004) in their article on machine-based manipulation of faceted classification numbers.

For an effective facet-based interface, explicit strings, even if longer or redundant for human viewers, lead to more effective, comprehensive results.

Possible solution (LCSH practice):

There is no systematic way to search for areas where there might be implied facets in LCSH, but as these are discovered, the instructions can be changed to make implicit facets explicit and old records could be updated.

8. Time Facet: Often Omitted, Not Coextensive with the Work, and Not Coded for Machine-Based Parsing

Problems:

1. Chronological information is often omitted from LCSH subject strings. Because of this, the chronological designations that users see as options in faceted interfaces only represent a fraction of relevant materials. Chronological subdivision is not allowed where it has not been explicitly established except for the ability to mark time periods at the century level in most circumstances. Chronological subdivisions are omitted where they are implied by the main topic. For example, catalogers are instructed to use “Underground Railroad \$x History” rather than “Underground Railroad \$x History \$y 19th century” ([SCM H 620](#)). [SCM H 1592](#) points out that many events are not known by a specific proper name and therefore are not established independently under a specific heading that includes a time period. Instead, these events are put under general headings by category and place, e.g. “Dust storms \$z Illinois \$x History,” which do not support searching by time period.
2. Chronological information is often entered as words, which are not related to specific dates or date ranges that could be used for searching. [SCM H 620](#) lists various ways of expressing chronological periods in LCSH, including periods named in words (“Iron age,” “Eighteenth century”), implied periods (“Post-communism”), headings with broad adjectival qualifiers (“Literature, Ancient”), events with dates (“Pan Am Flight 103 Bombing Incident, 1988#), and chronological subdivisions to be used after topical or geographical headings, with or without dates (“\$y 500-1400,” “\$y Jurassic”). The first four types are coded in MARC 650 \$a and are not easy to systematically distinguish from topical headings.
3. Chronological information is only entered in pre-selected ranges and is not coextensive with the coverage of the work. A book on an event in 1868 might be marked with nothing more specific than “\$y 19th century.”
4. Chronological information is not marked in a way that exact dates or date ranges can be reliably identified or manipulated by a computer.

Possible solutions (LCSH practice, structure):

RSWK, the German subject headings system ([Frommeyer, 2004](#)), encodes the exact time period covered by the document in the chronological facet, as does FAST in most cases ([Dean, 2003](#)). FAST uses the MARC 648 field to record this data and catalogers using regular LCSH could also start using this field. Exact date ranges allow for more precise chronological searching and can potentially be mapped to standard time periods. Chronological information in subject headings can be given both as date ranges and in words, some of which (e.g., Middle Ages) cannot be mapped to exact dates or for which the meaning may vary based on geographic area.

Frommeyer ([2004](#)) makes recommendations for improving the way chronological information is recorded, both for explicit date ranges and for named (and sometimes vague) periods, such as the Middle Ages. She recommends storing exact time spans in bibliographic subject headings as numeric time spans that can be used with greater/lesser than and equal operators. She acknowledges that this would be difficult to do retrospectively with LCSH as the time span in the existing subject heading is often not coextensive with the period covered by the work, but points out that sometimes this information can be found in titles or other parts of the bibliographic record. She also recommends that time spans be included in authority records for named periods. These could then be inherited by the associated bibliographic records. Finally, she would like to see the creation of a “chronology authority file” that would support more sophisticated browsing and provide insight into various temporal relationships (for example, by providing views limited to certain geographic or

subject areas). She envisions a display modeled on the sort of timeline interface commonly seen in digital encyclopedias. Petras et al. (2005) describe a project to map named time periods or events to their associated locations and dates or date ranges in the way that gazetteers map place names to latitude and longitude coordinates. Their test case is LCSH, although they limited themselves to the chronological information easily identified by machine (i.e., \$y subdivisions and not explicit or implicit chronological information in other fields, such as \$a of topical subject headings or \$d of personal name headings). They created an interface for searching and browsing by location, time, time period type (e.g., civil revolutions; these categories were manually entered) and to search for specific time periods and events. Faceted interfaces using LCSH would benefit from the inclusion of more chronological information in subject headings. Current selective use of chronological subdivisions is likely to mislead users into thinking they are getting more comprehensive search results than they actually are. With the appropriate interface, a switch to recording exact dates of chronological coverage would enable more precise, flexible, and comprehensive searching. Many works have specific date ranges or historical periods specified in their titles or in tables of contents or publisher's blurbs that could easily be added at the time of cataloging. Some works cover a vague contemporary period, which may be harder to decide how to encode. In addition to the limits on when time-related information may be included, chronological data is not added to LCSH in forms in which it is useful for machine manipulation.

Marking of beginning and ending dates would enable more manipulation of time-related data. Mapping of named time periods and events to date ranges would also improve access to chronological information.

9. More Specific Categories of Topical Headings

Problem:

More browsing options could be provided in faceted interfaces if subject headings were marked as belonging to certain groups or categories. These categories could support more precise browsing in some situations. For example, if applicable headings were marked as classes of persons, it would be possible to allow users to browse a list of biographies arranged by class of person covered (e.g., artists, presidents, rock musicians). Another example of this type of categorization is the way that the U.S. Board on Geographic Names (<http://geonames.usgs.gov>) assigns specific geographic names to various feature categories, such as lakes, summits (mountains), and populated places. Headings could also be usefully grouped by broad subject area (e.g., topics that are inherently legal or economic in nature). Svenonius (2000) points out that "a major reason for classifying terms in a subject language is that the resulting categories can be used to formulate the syntax rules of the language" (p. 19). In terms of LCSH, this would permit automated validation of free-floating subdivisions that are only supposed to be used with certain types of main headings.

Possible solutions (LCSH structure):

There are a number of possible ways to attempt to identify headings in certain categories or belonging to certain topic areas. Several are described below, including patterns in the use of free-floating and pattern-based subdivisions in a large collection of bibliographic records, occurrence of certain words in headings, broader and narrower term relationships in authority records, and correspondence between subject headings and classification numbers in authority and bibliographic records.

Use of Free-Floating and Pattern-Based Subdivisions in Bibliographic Records

The *Subject Cataloging Manual* lists a number of "free-floating" topical and form/genre subdivisions. Free-floating means that these subdivisions are permitted to be used following base headings without the combination being explicitly established and given its own authority record. In addition to free-floating subdivisions, there are also so-called pattern headings, where one heading of a given category is designated as the one under which relevant subdivisions will be explicitly established. For example, subdivisions applicable to any religion (except Christianity) are explicitly established under "Buddhism." A list of these categories can be found in the *Subject Cataloging Manual*.

The categories in the *Subject Cataloging Manual* are intended to define groups of topical headings in order to specify which topical and form subdivisions may be freely combined with them. For example, classes of persons, such as "Librarians," can be followed by the subdivision "\$v Biography" or diseases, such as "Cancer," can be followed by the subdivision "\$x Patients." With the addition of the free-floating subdivision "\$x Patients," the status of the string as a

whole changes to class of persons and can only be further subdivided by subdivisions permitted under classes of persons. So “Cancer” is a disease and cannot be followed by “\$v Diaries,” but “Cancer \$x Patients” is a class of persons to which “\$v Diaries” could be appended. It is also the case that sometimes specific types of patients, such as “Diabetics,” are editorially established, and a cross reference is made from the form using the free-floating subdivision (“Diabetes \$x Patients”).

The category or categories of headings after which a free-floating heading can be used are encoded in the 073 field of subject subdivision (180 and 185) authority records. \$a contains the section of the *Subject Cataloging Manual* that discusses that type of heading, such as “H 1100,” and \$z contains “lcsb.” If more than one section applies, each section is given in a separate \$a. The instructions for use are also given as notes in 680 \$i in the form “Use as a topical subdivision under classes of persons.”

Knowing what individual subject headings fall into these categories could be useful for arranging and displaying these headings for user consumption. Not all subject headings fall into one of these categories, but once the headings that belong to currently defined categories have been identified, it might be possible to examine the remaining headings to see if they can be usefully grouped in some way. Unfortunately, there is nothing in individual topical subject heading authority records that says what category a given heading belongs to. A human being is expected to infer what subject headings belong to what categories on an individual basis. However, it is possible to identify the category into which many base subject headings fall by looking at patterns of associated topical and form subdivisions in bibliographic records.

The best free-floating subdivisions for this purpose are those that can be used with only one category of base headings. Some free-floating subdivisions are useable under all topical headings and thus have no discriminatory power for this purpose. Some free-floating subdivisions are difficult to use for this type of analysis because they are used under different categories of headings with different meanings. Free-floating subdivisions that are used under different categories of base headings with similar meanings cannot be used to unambiguously determine the category of the preceding base heading. However, it is possible that an analysis of the pattern of all the free-floating headings that are found to follow a given topical base heading could be used to extrapolate the category of the base heading. Any analysis of this sort would not reach a long tail of headings that have not been used in combination with a subdivision, but many common headings could be automatically determined and this might be enough to do something useful.

As a test of this theory, I searched for records in our catalog with three reasonably common subdivisions that occur only after classes of persons or ethnic groups (which are essentially a subset of classes of persons). These are “\$x Attitudes,” “\$v Diaries,” and “\$x Ethnic relations.” I pulled all the records containing one of these subdivisions in a 650 topical subject field from our catalog, de-duplicated the headings, and removed a few obvious errors. This left 816 unique headings. The results reveal some potential problems with this approach. There were a number of errors in tagging (e.g. “\$z Attitudes” as a geographical subdivision) and subdivisions used after base headings that don’t belong to an authorized category. A large number of misused subdivisions were based on a misinterpretation of the specific meaning of “\$x Attitudes” in LCSH.

Including the ones that are possible, but questionable, for inclusion as classes of person as errors, there is around a 7% error rate. If a larger set of data was used and a threshold for number of occurrences of a heading with an appropriate subdivision were set, the rate of accuracy might be improved somewhat.

Looking for Specific Words in Subject Headings

Multi-word subject headings that include certain words (“women” or “artists”) that are not also used as adjectives and don’t include any prepositions (“Adult education of women” or “Artists and museums”), can often be assumed to fall into a certain category (classes of persons). So all headings that include the word “women,” but don’t include a preposition could provisionally be assigned to the category “classes of persons”

To test this approach, I extracted all the 650 \$a subfields (base topical subject headings) that included the word “women” from our catalog. After removing duplicates and a few typographical errors, 795 unique headings remained. My assumption was that most of the headings that included prepositions would not be classes of persons and most of the headings that did not would represent classes of persons. I provisionally marked all the headings that included the following prepositions: “against,” “and,” “for,” “in,” “of,” “on,” “to,” and “with” as not classes of persons. This totaled

250 headings. I then scanned the headings for erroneously assigned headings. For headings without prepositions, there were only three that appeared not to be classes of persons. Two began with the phrase “Women-owned” and the other was “National Women Veterans Recognition Week.” Eliminating headings containing “owned” and “week,” as well as “day,” “month,” and “year,” from the list of probable classes of persons would resolve these problems. There were more problems with headings that included prepositions turning out to be classes of persons. There were seven “and” headings (“Women track and field athletes”), four “of” headings (“Women heads of state”), and three “with” headings (“Women with mental disabilities”).

However, the most problematic (and common) preposition was “in.” In particular, headings that include the phrase “women in X” can be hard to judge. There are a number of headings of the form “X in literature” or “X in motion pictures” that are intended to represent the portrayal of X in those forms and are not classes of persons. There are also some that might be construed as classes of persons, such as “Women in the advertising industry,” which, despite its broader term of “Advertising,” seems like it could refer to a class of persons. The note in the authority record (sh 85147608) states that “here are entered works discussing all aspects of women’s involvement in advertising. Works discussing the portrayal of women in advertising are entered under \$a Women in advertising.” On the other hand, perhaps it is intended to refer to the activity of a class of persons. However, its use in the subject string “Women in the advertising industry \$z United States \$v Biography” in a Library of Congress-generated bibliographic record in our catalog suggests that it might be intended to be considered as a class of persons. It’s not clear to me what “Women in Buddhism” might mean (as distinct from “Buddhist women” or “Women \$x Religious aspects \$x Buddhism”). Ignoring the “in” headings, which probably need further clarification, this method miscategorized 14 out of 250 headings with prepositions (6%) and 3 out of 545 headings without prepositions (.5%).

This approach works less well with terms that are also used as adjectives in LCSH. It can’t distinguish between “African American poets” (a class of persons) and “African American art” (not a class of persons). However, a more limited version of this approach may still work. In many cases, if the category of the final term in the heading is known (e.g., “poets” or “art”) and there are no prepositions in the heading, the heading can be assumed to be in the same category as the final term.

Broader and Narrower Term Relationships

Broader and narrower relationships could theoretically be used to assign narrower terms to the same category as the corresponding broader term. However, for historical reasons, there are many types of broader-narrower relationships in LCSH and these are not all of the species-genus type. As Svenonius (2000) points out, the Library of Congress took the “quick-fix” approach and used an automated process to convert all of its historically inconsistent *see* and *see also* references to broader and narrower terms in one fell swoop, with only “a few ‘Band-Aid’ reparations ... to fix some of the more egregious structural deficiencies” (p. 22). Although broader and narrower terms may not be useful for automatic identification of narrower terms that belong to the same category as broader terms, it might be possible to use the existing broader and narrower relationships to confirm categorizations made by the above methods. It might also be a useful exercise to manually examine uncategorized narrower terms of broader terms identified by the above methods in order to identify areas where the syndetic structure could be made more consistent. If a manually-examined narrower term does belong to the same category as its broader term, it could be assigned to that category. If the narrower term does not belong to the same category, its place in the syndetic structure could be reevaluated.

Classification Numbers in Authority or Bibliographic Records and Broad Subject Areas

Another way to potentially divide up topical facets is by broad subject area, such as topics that are inherently legal or economic in nature. This might allow mapping of a specific subject to a broader area or areas that would be more useful entry vocabulary for browsing. This is somewhat complicated by the fact that subject headings are generally held to represent topics while classification numbers represent disciplines. However, many topics can be primarily associated with a specific broad subject area, even if they might often be discussed from the perspective of other disciplines.

It might be possible to identify relevant subject areas for many headings by mapping them to commonly associated classification numbers. Some subject heading and name authority records contain LC classification numbers or ranges in the 053 field (possibly qualified by perspective in \$c) so these could be used if present. Some examples include.

Chemistry: QD Lungs \$x Cancer: RC280.L8 Dandelions: QK495.C74 \$c Botany Computers: QA75.5 \$b QA76.95 \$c Mathematics TK7885 \$b TK7895 \$c Electrical engineering Death: BD443.8 \$b BD445 \$c

Philosophy GR455 \$c Folklore GT3150 \$b GT3390.5 \$c Manners and customs HQ1073 \$b HQ1073.5 \$c General QH671 \$c Cytology QP87 \$c Physiology RA1063 \$b RA1063.5 \$c Medical jurisprudence
Clearly “Lungs \$x Cancer” has a closer correspondence to medicine than “Death” does to any given discipline.

Another possible approach is that taken by the Library of Congress’ Classification Web product (<http://classificationweb.net/>). ClassWeb maps classification numbers in bibliographic records to the first subject heading used in a record when it is a 650 (topical) or 651 (geographic) heading. It does not work for mapping names and titles to classification numbers. Traditionally, this is the subject heading that is supposed to most closely correlate with the subject of the work and the classification number. At least in cases where a clear predominant mapping or cluster of mappings occurs, the related broad subject area or areas could be determined.

Using WordNet or Another Existing Set of Semantic Relationships

Another possible approach is that taken by Yee et al. (2003), who semi-automatically assigned words in free text to metadata categories using their higher-level category labels in WordNet (<http://wordnet.princeton.edu/>).

Use of these Categories to Improve Consistency and Authority Control

Most of the LCSH heading strings that are not established editorially are constructed by attaching geographic or topical or form subdivisions to a base subject as described above. Currently OCLC’s control headings function (http://www.oclc.org/support/documentation/connexion/browser/authorities/apply_ac_bib_records/default.htm) does a fairly good job of knowing where a geographic subdivision should go in a given string because whether or not each element of a string can be geographically subdivided is coded in its authority record. The control headings function can also inform the cataloger that a heading cannot take a geographic subdivision at all.

As described above, free-floating and pattern subdivisions are often permitted only under certain types of base headings (e.g., classes of persons, religions). If the appropriate categories were coded in authority records and combined with a set of rules, a computer would be able to validate any possible combination of elements in an LCSH string. Although this approach might be more complicated up front, it seems likely to be much more useful and less of a monumental undertaking than trying to explicitly establish individual combinations, as the Library of Congress is currently attempting. It would be impractical to explicitly establish all the possible combinations and even if the practice were limited to commonly-occurring headings, it seems like a large-scale investment in something that may not be helpful to us if we want to move to a more faceted, and perhaps post-coordinate, system. Machine-based rules might also be more accurate—our local system is set up to require authority records for individual combinations, but these authority records are sometimes created based on their use in our database without being sufficiently vetted. This leads to the establishment of incorrectly-constructed headings or heading strings that should be mutually exclusive (e.g., at one time we had local authority records for both “Youth \$x Alcohol use \$z United States” and “Youth \$z United States \$x Alcohol use”)

10. Facets as Building Blocks for a More Hospitable, Machine-Controllable Subject Vocabulary and Better Display of Relationships Between Terms (Syntactic Relationships)

Problem:

An insufficient number of facets to express relationships between terms in LCSH hinders our ability to manipulate and re-use information in subject strings. As Vickery (1970) points out, assigning terms to specific facets provides meaningful relationships between terms. For machine-manipulation of faceted classification numbers, Slavic and Cordeiro (2004) emphasize the “need to fully declare and encode each compositional element of a synthesized notation” (p. 5). Unfortunately, the syntax of LCSH covers only a limited number of types of relationships and these are often unclear. These are generally the same as the facets used in FAST, i.e., the topical main and subdivision terms (150 and 180), the genre-form main and subdivision terms (155, only beginning to be established, and 185), geographic terms (151), chronological subdivisions (\$y, which is only established in conjunction with another term; stand-alone chronological terms, e.g., “Sixteenth century” or “Middle Ages,” are established as topical headings in 150 in LCSH), as well as personal (100), corporate (110), conference/meeting (111), and title (130) headings borrowed from the name authority file and established under the aegis of AACR2 rather than LCSH rules.

In many cases the relationship between the different subfields of subject heading strings is unclear. An example of this is the use of geographic subdivision after the heading “Prisoners of war.” The relevant authority record (sh 85106971) explicitly instructs catalogers that “when subdivided by place, the name of the place may designate either the current location of prisoners of war, or the place of origin. For prisoners of war of a particular nationality held in another country, two headings are assigned: \$a 1. Prisoners of war–[country of nationality]. 2. Prisoners of war–[place where held].” Therefore, “Prisoners of war \$z China” can mean either Chinese prisoners of war or prisoners of war of any nationality being held in China.

This exemplifies two problems that lead to ambiguity in the meaning of geographic subdivisions in LCSH. One is that sometimes geographic subdivision is used to represent nationality instead of an actual place, and this isn’t always obvious to users. In many cases, nationality and place are so often the same that they are conflated, which makes for sticky situations when they are different. Normally, LCSH deems it redundant to bring out both place and nationality unless they are different. For “classes of persons” headings on biographies, usually either a geographic subdivision or a nationality qualifier is used, but not both. “Authors, French” seems to refer to nationality (and generally seems to be doubled when someone has relocated; biographies of Nabokov are mostly under both “Authors, American” and “Authors, Russian”), but based on the fact that Picasso’s biographies are put under “Artists \$z France,” but not “Artists \$z Spain,” France seems to mean place of activity rather than nationality.

The second example (Prisoners of war–[place where held]) could possibly confuse a country as a place with a country as a governmental body or an entity capable of acting as an agent. Although country names can be treated as corporate bodies in the name authority file where they are tagged 110 when they are the issuing official documents of a signatory of a treaty or otherwise responsible for certain types of reports or other works, there is usually no way to differentiate between a country as a place and the government of that country in a subject subdivision. This lack of clear and consistent expression of relationships among parts of subject heading strings hinders our ability to manipulate these strings and express these relationships in different ways.

Some relationships occur within an individual subject term (i.e., within a single subfield). Although the relationships (e.g., “War and literature,” “War in literature,” and the topical subdivision under individual wars “Literature and the war”) are grammatically marked for human consumption, they cannot be explicitly marked for machine manipulation. This is part of the problem of single concepts and compound terms described above.

Possible solution (LCSH structure, system design):

More explicit relationships and information about categories of headings would help retain the benefits of pre-coordinated subject headings (precision, context for browsing) while overcoming some of the limitations of our current implementation (cryptic syntax, inflexible citation order). More precise coding of syntactic relationships would increase our ability to make the meaning of subject strings clearer and the display of LCSH less cryptic.

For example, “\$x History” as a subdivision generally means “history of” something. So what currently shows as “United States–History” could be displayed to users as “History of the United States” or “United States, History of” depending on context or need. Significant parts of the meaning of an LCSH string depend on citation order, but if these meanings were displayed more explicitly rather than just using “dash dash,” they would be clearer to users. For example, “United States \$x Geography” could be displayed as “Geography of the United States” and “Geography \$z United States,” could be displayed as “Geography (discipline) in the United States.” This distinction would be totally lost in a naïve conversion to a post-coordinate system using current LCSH, but could be preserved in a faceted system. Many existing headings could be converted to clearer display forms using algorithms. The cases where this cannot be done in a rule-based fashion generally point to some inadequacy in our current syntax or encoding. For example, the fact that we don’t know whether “Detectives \$z Egypt” means “Detectives from Egypt/Egyptian detectives” or “Detectives in Egypt” is a weakness of our current system.

The inflexibility of citation order in subject strings is only partially overcome by so-called rotating browse lists that provide entry points starting with each element of a heading string. Rotating browse list can also have significant drawbacks and lead to confusion or loss of precision. In our OPAC, the way the headings are rotated means that there is no way for a user to distinguish between or do an effective search for “History \$x Philosophy” (philosophy of history) and “Philosophy \$x History” (history of philosophy) despite the fact that these are two very different things. If more explicit

information about the relationships between the elements of subject heading string could be displayed, more citation orders would become possible without conflating things together incorrectly.

Conclusion

A number of changes to the structure of LCSH could create a vocabulary that better supports browsing and navigation in faceted interfaces. If we think of elements of LCSH as building blocks related to each other in a rule-based manner rather than hard-coded strings, if we created more consistent and complete relationships between terms and parts of terms, and if we had a method for marking in LCSH vocabulary the single concepts and the characteristics of division by which they have been analyzed, we might be able to build a more useful and flexible system of subject access. LCSH would also benefit from the creation of a more rigorous and consistent syndetic structure, more granular coding of exact chronological ranges, and mapping headings to topical categories (legal, economic) and functional groups (classes of persons, religions).

Some changes in practice could also be beneficial. These include explicit coding of all facets of subject headings, including those currently considered to be implicit and much chronological information, as well as the use of selected broad-bucket headings for browsing access.

Faceted interfaces to existing LCSH could be improved by incorporating cross-references from subject authority records and consolidating certain synonymous terms for display, such as many geographical headings ("Arkansas" and "Ark.") and certain subject subdivisions ("\$x Social conditions" and "\$x Social aspects"). Some of the desirable features of FAST, such as hierarchical geographic information, can be mapped from existing LCSH headings. Some of the topical and functional categorization described above could probably be done without modifying LCSH and be synchronized with existing LCSH records.

Current Work on Shifting LCSH to a Faceted Syntax

The best-known example of an attempt to examine the potential of LCSH in a facet-based context is OCLC's FAST project (www.oclc.org/research/projects/fast). FAST is an attempt to use the rich vocabulary of LCSH in a context where it can be more easily assigned by personnel with less training. However, it does do some things that improve subject access. FAST includes eight facets:

- Topical
- Geographic
- Form (genre)
- Chronological
- Personal names
- Corporate names
- Conferences/Meetings
- Uniform titles

These are based on the types of terms that are tagged or subfielded separately in MARC records and therefore are easily extracted from current LCSH authority records. The forms of information in some facets, primarily the geographic and chronological, have been changed from those found in standard pre-coordinated LCSH. I have discussed some aspects of FAST that I think have implications for improving access in the relevant sections above.

FAST headings for a given record can probably be largely automatically machine-generated from existing LCSH headings assigned to that record. If this can be done, then it may be possible for system designers and implementers to bring many of the facet-based improvements FAST brings-discussed in this paper-immediately to systems independently of changes to LCSH design or practice. However, some data, such as the exact chronological coverage that FAST provides, are not in existing LCSH headings. Some data, such as the dates and geographic areas in the headings for some time periods that

FAST has mapped from LCSH 150 (topical) to FAST 111 (meetings/ events) authority records, should additionally be extracted and mapped to separate facets, but it is unclear to me whether there is an existing algorithm for doing so. More information from the OCLC FAST researchers on what they have intended and discovered in this area would be welcome.

Anderson and Hofmann (2006) take a rather different approach. Like the FAST project they base their proposal on the existing LCSH vocabulary in order to take advantage of its comprehensiveness and also to provide some level of consistency and backwards-compatibility with current practice. However, they are more interested in improving the usefulness of the headings than simplifying application. They state that faceted syntax “encourages the use of post-coordinate single-concept terms, but it provides syntactic rules for combining them in subject strings that in many cases can be coextensive with the content of the work indexed” (Anderson and Hofmann, 2006, p. 10). This enables the use of a simpler, post-coordinate syntax, without losing LCSH’s support for browsing and context.

They propose to rework existing LCSH headings into a system that includes the facets of the current Bliss Classification (a few of which they have slightly renamed). These facets are:

Topical facets:

- Thing/entity (objects, persons, institutions, texts)
- Kind (kinds of things/entities)
- Part (parts or components of things/entities)
- Property/Attribute
- Material (from which things/entities are made)
- Process (usually a naturally occurring process, like aging, maturation; usually experienced by the entity; usually does not have an object)
- Operation (usually performed by an agent, and often with an entity as the object of the action; also used for events)
- Client (for whom some operation is performed)
- Product
- By-product (unintended and often negative results of operations)
- Agent/Means
- Space (geographic place)
- Time

Non-topical facets (features);

- Approach (methodological approach, point of view, etc.)
- Format
- Medium
- Audience

By assigning LCSH terms to these more specific facets, they improve the scheme’s ability to convey syntactic relationships. Although better precision through the expression of syntactic relationships is a traditional strength of conventional pre-coordinated subject systems over post-coordinate ones (Svenonius, 1992), the number and type of relationships that are explicitly and consistently accounted for in LCSH is much less than in the Bliss-based list above. Anderson and Hofmann generally use the terms in LCSH “as is,” but they do recommend decomposing what they call precombined headings, such as “Homosexuality and education” and “Homosexuality in literature” in order to create single-concept terms.

They claim that making these changes would create a more consistent syntax and that the one long, but flexible, heading string that they advocate would be coextensive with the work being cataloged in a way that current headings are not (where the combination of more than one heading string is often held to be coextensive with the subject of the work as a whole). Their proposal could not be easily applied retrospectively, even with manual review of bibliographic records,

because it adds information that currently isn't in the headings and can't always be accurately inferred from existing bibliographic records

References

- Anderson, J. D., & Hofmann, M. A. (2006). A Fully Faceted Syntax for Library of Congress Subject Headings. *Cataloging & Classification Quarterly*, 43(1), 7-38. ([COinS](#))
- Dean, R. J. (2003). FAST: Development of Simplified Headings for Metadata. Paper presented at Authority Control: Definition and International Experiences Conference, Florence, Italy. Retrieved October 14, 2007, from http://www.sba.unifi.it/ac/relazioni/dean_eng.pdf
- Frommeyer, J. (2004). Chronological Terms and Period Subdivisions in LCSH, RAMEAU, and RSWK: Development of an Integrative Model for Time Retrieval across Various Online Catalogs. *Library Resources & Technical Services*, 48(3), 199-212. ([COinS](#))
- Hemmasi, H. (1996). The Music Thesaurus: A Faceted Approach to LCSH. Paper presented at Authority Control in the 21st Century: An Invitational Conference, Dublin, OH. Retrieved from <http://digitalarchive.oclc.org/da/ViewObjectMain.jsp?fileid=0000003520:000000091791&reqid=354&frame=false>
- Library of Congress Cataloging Policy and Support Office (CPSO). (2007). *Subject Cataloging Manual: Subject Headings*. Retrieved October 13, 2007, from Cataloger's Desktop. ([COinS](#))
- McBride, J. L. (2000). Faceted Subject Access for Music Through USMARC: A Case for Linked Fields. *Cataloging & Classification Quarterly*, 31(1), 15-30. ([COinS](#))
- Petrás, V., Meiske, M., Larson, R., Zerneck, J., Carl, K., & Buckland, M. (2005). Leveraging Library of Congress Subject Headings to Improve Search for Events - A Time Period Directory. Retrieved from <http://metadata.sims.berkeley.edu/tpd/TPD-report.pdf>
- Rosch, E. (1978). Principles of Categorization. In E. Rosch & B. B. Lloyd (Eds.), *Cognition and Categorization*. Hillsdale, NJ: Lawrence Erlbaum Associates (pp. 27-48). ([COinS](#))
- Slavic, A., & Cordeiro, M. I. (2004). Core Requirements for Automation of Analytico-Synthetic Classifications. *Proceedings International Society for Knowledge Organization Conference*. Retrieved from <http://dlist.sir.arizona.edu/651>
- Svenonius, E. (1992). Proposal #2: The Expanded Use of Free-Floating Subdivisions in the Library of Congress Subject Headings System: Arguments in Favor In M. O. Conway (Ed.), *The Future of Subdivisions in the Library of Congress Subject Headings System: Report from the Subject Subdivision Conference Sponsored by the Library of Congress, May 9-12, 1991* (pp. 36-38). Washington, DC: Library of Congress Cataloging Distribution Service.
- Svenonius, E. (2000). LCSH: Semantics, Syntax, and Specificity. *Cataloging & Classification Quarterly*, 29(1/2), 17-30. ([COinS](#))
- Svenonius, E. (2003). Design of Controlled Vocabularies. In M. Drake (Ed.), *Encyclopedia of Library and Information Science*. New York: Marcel Dekker (pp. 822-838). ([COinS](#))
- Tunkelang, Daniel. (2006). Dynamic Category Sets: An Approach for Faceted Search. Paper presented at SIGIR '06 Workshop on Faceted Search Conference, Seattle, WA. Retrieved October 14, 2007, from <http://www.cs.cmu.edu/~quixote/DynamicCategorySets.pdf>
- UC Berkeley School of Information. (n.d.). The Flamenco Search Interface Project. Retrieved October 13, 2007, from <http://flamenco.berkeley.edu>

Vickery, B. C. (1970). *Faceted Classification: A Guide to the Construction and Use of Special Schemes* (Reprinted with additional material). London, England: Aslib. ([COinS](#))

Yee, K., Swearingen, K., Li, K., & Hearst, M. (2003). Faceted Metadata for Image Search and Browsing. *Proceedings of the ACM Conference on Computer-Human Interaction*, 401-408. ([COinS](#))

Yee, M. M. (2001). Two Genre and Form Lists for Moving Image and Broadcast Materials: A Comparison. *Cataloging & Classification Quarterly*, 31(3/4), 237-295. ([COinS](#))

About the Author

Kelley McGrath is Cataloging & Metadata Services Librarian (Audiovisual) at Ball State University and also serves as the chair of Online Audiovisual Catalogers (OLAC) Cataloging Policy Committee. She can be reached at kmcgrath@bsu.edu.

The Rutgers Workflow Management System: Migrating a Digital Object Management Utility to Open Source

By Grace Agnew & Yang Yu

Introduction

Ingest of digital objects is a core service of a repository architecture. Most repository services, from preservation and storage to discovery and retrieval, are dependent on the information collected about the digital object at ingest. The scalability of the repository, particularly to support the simultaneous ingest of many digital collections, is also dependent on this critical service. The Workflow Management System (WMS) was originally developed to meet RUL's need for a flexible, extensible web-based service to support repository development. The WMS includes a sophisticated metadata architecture that was designed to support any digital collection, from any contributor, whether a faculty member depositing the research products of a large scientific experiment or a small historical society participating in a collaborative cultural heritage portal. This article describes the development of an object ingest and metadata creation application that began as a front-end service for RUL's Fedora repository. As we presented our WMS to our peers at conferences [1], we were approached about sharing the application. Before sharing WMS with libraries, archives, and other organizations, we needed to significantly retool the software to remove Rutgers-specific dependencies and to support customization for each institution's unique circumstances for information management and delivery. This paper describes the background and rationale for developing the WMS; its design and functionality, particularly to provide a sophisticated event-based data model and metadata architecture within a METS (Metadata Encoding & Transmission Standard) framework; and the re-engineering decisions that were required to create a robust open source application. The article closes with the policy and procedural issues that must be addressed before the WMS is released in the open source community, as well as the next steps in WMS development.

Background

In 2002, the Rutgers University Libraries began exploring open source repository platforms to serve as the basis for a comprehensive cyberinfrastructure that would manage the preservation, access and use of the intellectual property of a large research university. The Fedora repository architecture [2] was selected for the sophistication of its service-oriented design and the simplicity of its approach to resource management. Within Fedora, anything can be an information object. No assumptions are made about the nature of the information to be managed or its intended use. Core services for preservation and management are provided for digital objects, but most services beyond basic preservation and access must be locally developed and layered upon the core architecture.

Our survey of Rutgers research, based on an analysis of grant-funded projects and other research products hosted on Rutgers websites, revealed an enormous breadth of digital content and a wide range of approaches for managing that content, from large scientific databases maintained on multiple Excel® spreadsheets to a complete portal with metadata, digital objects, and a suite of user services. We realized that we would need a flexible, extensible metadata architecture to encompass the wealth of information available on campus and to support the metadata decisions that many faculty had already made to support discovery and access by colleagues in their disciplines.

In addition to developing an institutional repository to support faculty research and publications, Rutgers had received a grant from the Institute of Museum and Library Services to develop a statewide cultural heritage repository, the *New Jersey Digital Highway* [3]. Discussions with archivists, museum curators and librarians around the state identified a need to provide not just discovery and access but management of the state's cultural heritage resources. As one of the original 13 U.S. colonies, New Jersey has a rich historic heritage, with many artifacts housed in small historical societies and museums around the state. The cultural heritage community was very interested in an information architecture that would support ongoing management of the analog source materials—photographs, papers, artifacts, etc., as well as the digital surrogates deposited in the New Jersey Digital Highway repository.

The Rutgers University Libraries needed an information architecture that would support several critical needs:

1. to enable the libraries to integrate and support the heterogeneous research products and publications of Rutgers University faculty;
2. to support management, preservation and access to historic and cultural source materials, particularly the New Jersey Digital Highway collection and RUL's special collections and
3. to integrate with the Fedora core architecture which supports both Dublin Core and the Fedora native XML schema, FOXML.

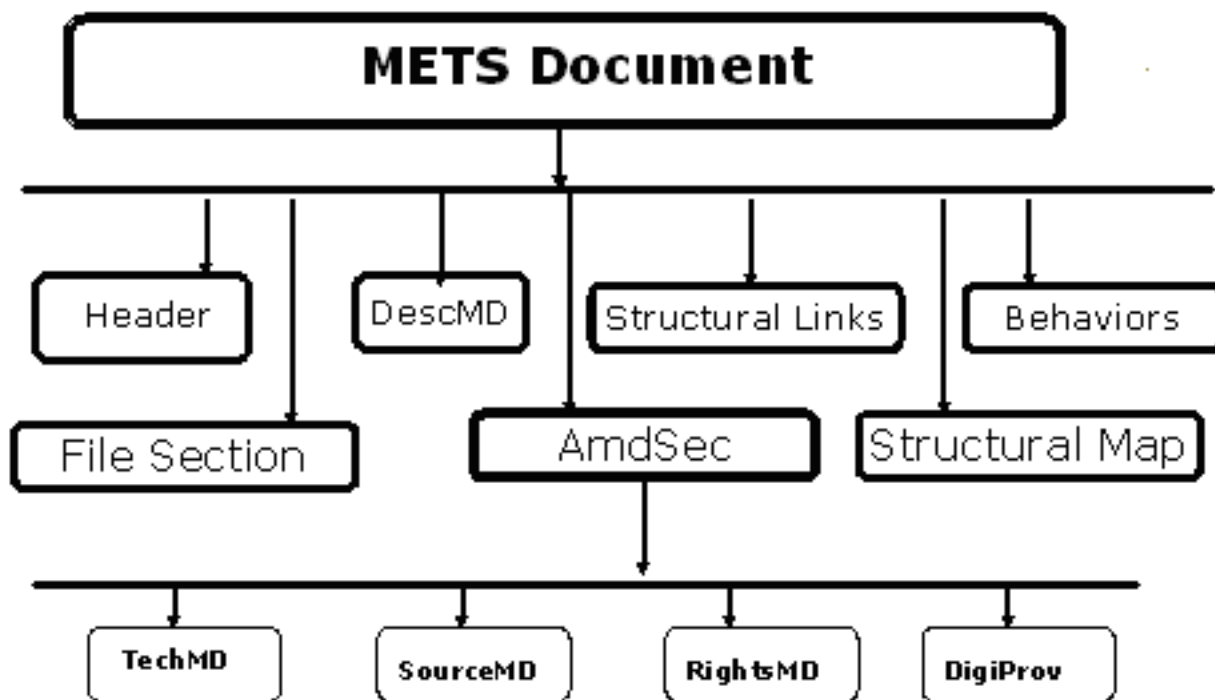
The Rutgers Information Architecture

Rutgers decided to use the Metadata Encoding & Transmission Standard, a metadata architecture supported as an international standard by the Library of Congress [4]. Fedora was initially designed with a METS data architecture. Fedora migrated its data architecture to FOXML, which maps to METS and has been described as "METS Lite," so the choice to use METS made sense from an architectural standpoint. [In addition, METS provides all the categories of information needed to manage and provide access to a resource.](#) METS concatenates different types of metadata with one or more versions of the object, as well as structural information and behaviors for relating METS components, navigating complex objects, such as the pages of a book and for displaying and using the digital object. The METS envelope provides a standardized XML wrapper for organizing, storing, managing and transporting all the METS components as a single object. The METS document is a standardized transmission package that can be shared across METS-compliant repositories.

There are seven component metadata documents within a METS document:

- Header, which provides basic information about the creation of the METS object
- Descriptive metadata, supporting discovery and access to resources
- Administrative metadata, providing metadata about the creation, provenance and use of resources. Administrative metadata includes four subtypes: technical, source, rights and digital provenance
- File section, which groups together related files, such as the different digital manifestations of a resource-the TIFF digital master file, the JPEG access copy, etc.
- Structural map, which provides the hierarchical structure of a complex resource, and links the elements of the structure to relevant content files and metadata
- Structural Links, which enable hyperlinks between nodes in the structural map
- Behaviors are procedures or applications that can be executed upon content contained within the METS package. [4]

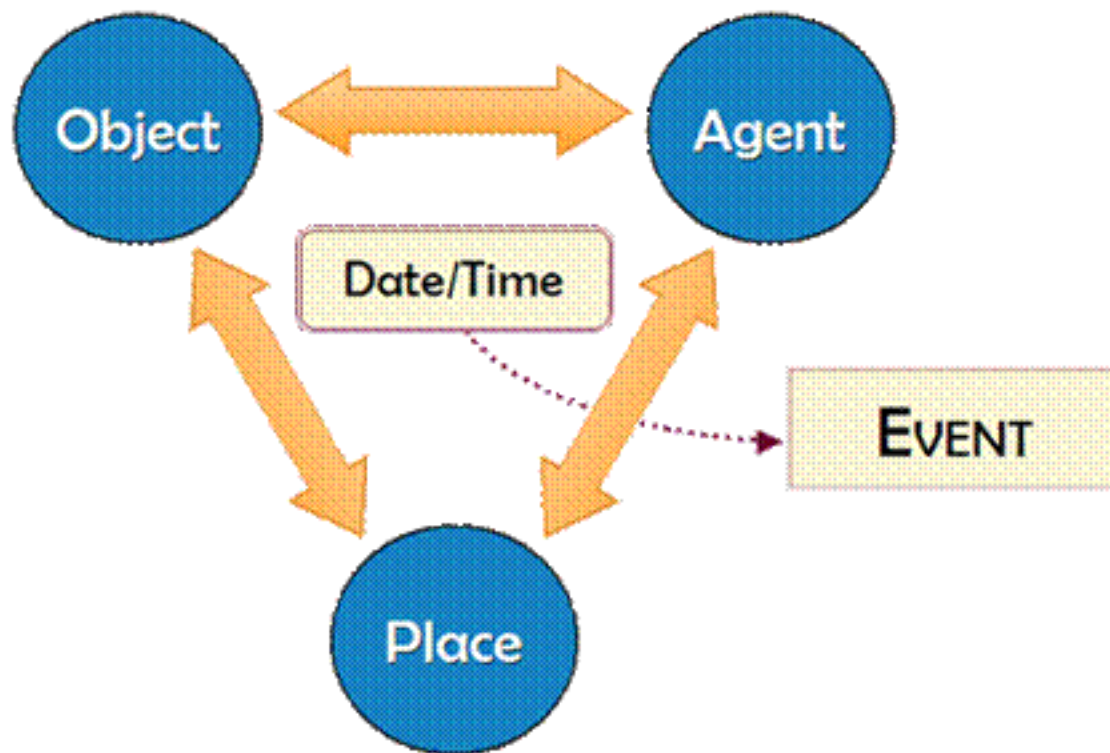
Figure 1. METS Information Package



METS accommodates all the physical manifestations of an information entity. In Rutgers' implementation of METS, the source object is the first generation of information under the control of the organization. For example, RUL may own a photograph of the Venus de Milo. The famous statue itself is not owned or managed by the library. Instead the source object, which is the first generation of information under the control of the organization, is the photograph of the Venus de Milo. Source information objects are generally information that can pass the "hurricane test," in other words it is the information you will save if your archive is in the path of a hurricane and you must evacuate the premises. RUL documents information about the provenance and condition of analog source materials in source metadata. RUL documents the characteristics of born digital source objects and digital master files in technical metadata. RUL places an equal focus on access to resources and long-term availability. This requires capturing information at ingest sufficient to manage objects and to safeguard and document intellectual property rights for rights holders. The Rutgers METS implementation includes a complete rights metadata implementation, with the ability to link rights documentation (deeds of gift, permission request letters, privacy releases, etc.) to the rights events that secure for RUL the ability to make copyright-protected resources available for Web dissemination.

One important way to ensure the long-term usefulness of information is to provide durable context for resources, so that a user in the future knows what they are accessing, how it is created, and what rights they have to its use. Since copyright currently extends 70 years beyond the death of the creator, durable provenance is important both to ensure authenticity of information and to ensure its legal availability over time. We address provenance by supporting the ability to add lifecycle and use information via event metadata continuously throughout the life of a digital resource. The RUL data model is primarily an event-based data model, intended to document the lifecycle of each digital resource incrementally, over time.

Figure 2: Rutgers University Libraries Data Model



RUL adds preservation and condition events, provenance events, rights events, and descriptive events, which document the cultural, pedagogical and research usefulness and impact of resources over time to the core metadata we create at ingest within the different METS documents. We chose the “event”, what happens to a resource at a specific time and place, within the context of METS metadata categories (descriptive, source rights, and digital provenance), as a standard conceptual data model that will work with all information domains. An event can have associated entities, such as a granting agency, an awards agency, a preservation service provider, a rights holder, or an exhibit curator. An event can have associated objects, such as a deed of gift or website. A descriptive event will also allow us to use social networking technologies, to capture critiques, recommendations, and use patterns by colleagues in a standardized way to provide nuanced access to resources. Such tracking is particularly critical for information products, such as experiment documentation and multimedia files, that may not have peer review status.

The event data model addressed our need to create rich “living” data objects that add the context necessary for provenance, discovery and use, and the METS document provides a framework for the contextual events we wanted to capture. Examples of provenance events include acquisition, donation, etc. Examples of preservation events include repair, reformatting, etc. Examples of rights events include license or permission, rights transfer, etc. Events provide meaningful context and document the entire lifecycle of an object. Figure 3 provides an example of a descriptive event—the exhibition to which an object belongs. Events enable us to document associated entities and objects, such as the curator of an exhibit and the exhibit catalog.

Figure 3: Example of a Descriptive Event in the WMS Exhibition to which the resource belongs

Event entries for **Event 1** [Existing event(s): 1]

Type: **Exhibition**

Label: **Remembering Newark's Greeks: An American Odyssey**

Place: **Newark Public Library**

Date & Time: **2002-10-21**
(YYYY OR YYYY-MM-DD OR YYYY-MM-DD hh:mm:ss)

Detail: **"Remember Newark's Greeks: An American Odyssey. A look at 100 years of the Greek Community in Newark, Photographs, Documents and Memorabilia, October 21, 2002 -December 31, 2002." Curated by**

Associated Entity

Role: **Curator of an exhibition**

Name: **Angelique Lampros**

Affiliation: **Newark Public Library**

Reference:

Detail:

AssociatedEvent Entry List

- [Curator of an exhibition] Angelique Lampros Newark
- [Curator of an exhibition] Peter Markos Newark Public
- [Curator of an exhibition] Charles F. Cummings Specie

Change Remove Add More

Associated Object

Type:

Name:

Reference:

Detail:

AssociatedObject Entry List

Another important event that can be captured in the WMS is the rights event. Whenever possible, we document the deed of gift or permission received from the rights holder that enables the repository to make a copyright-protected resource available for users.

Figure 4: RUL rights event -deed of gift

Firefox prevented this site from opening a popup window.

http://jms3.libraries.rutgers.edu/workflow/WMS_Main.php

Options

Rights Event

Event entries for: [Existing event(s): 1]

Type:

Label:

Place:

Date & Time:
(YYYY OR YYYY-MM-DD OR YYYY-MM-DD hh:mm:ss)

Detail:

Associated Entity

Role:

Name:

Affiliation:

Reference:

Detail:

Associated Entity List

Add More

Associated Object

Type:

Name:

Reference:

Detail:

Associated Object List

The Workflow Management System

Once we had a conceptual data model, a metadata architecture and a data element registry, we needed to incorporate the complex metadata architecture into a web-based object ingest and metadata capture tool for the RUL Fedora repository architecture. This tool was the Workflow Management System, which at that point consisted of a skeletal “placeholder” metadata implementation and a pipeline application for ingesting digital resources into the repository, creating access copies in multiple formats, and creating OCR files for textual materials or transcripts accompanying media files, which could be searched via full-text. The Workflow Management System needed to support a range of large and small libraries, museums and archives that would add resources and create metadata independently of the Rutgers University Libraries, as well as Rutgers faculty, who would deposit publications and research products in the RUcore, the Rutgers Community Repository [5]. For most *New Jersey Digital Highway* and RUcore participants, the Workflow Management System would be their only exposure to the inner workings of the repository. The WMS needed to be robust and intuitive while supporting a very sophisticated data architecture that used concepts and terminology that were unfamiliar even to experienced catalogers, who have worked primarily with MARC, Dublin Core and MODS (Metadata Object Description Schema) metadata standards.

The Workflow Management system that we designed over several years to support our sophisticated data architecture is now a core enabling technology for the RUL cyberinfrastructure. For two years, the WMS was extensively tested through its use for NJDH by participating museums, libraries and historical societies around the state of New Jersey. Three important issues emerged that had a great impact on the continuing development of the WMS. To begin with, most participants had already created some level of metadata for their collections, even if just a simple spreadsheet or word

processing file, and they were understandably reluctant to re-create this information, even through cutting and pasting. They were willing to iteratively add to their metadata once it was ingested, particularly if they felt the incremental event metadata added value, but they were unwilling to recreate existing information from scratch. A flexible mapping utility that could accept and map data elements from any metadata schema, from the complex to the rudimentary, became a critical component for enabling museums and libraries to participate in the New Jersey Digital Highway. The mapping utility received a further test recently when RUL assisted the Virginia Tech Library in developing a commemorative repository for the April 16 shooting tragedy. RUL was able to successfully map the spreadsheets that inventoried the thousands of banners, cards, and other tributes that the university received in the aftermath of the tragedy into useful metadata to enable Virginia Tech to quickly create a permanent digital archive.

A second issue was the need to ingest large amounts of source objects in bulk rather than uploading each digital object one at a time. The WMS initially required that each object be individually loaded. This proved to be very time-consuming, particularly for large digital files. Requiring that users individually load each digital object is very inefficient and definitely not scalable. It became important to develop a mass-ingest capability that supported unattended bulk loading of objects to address this issue.

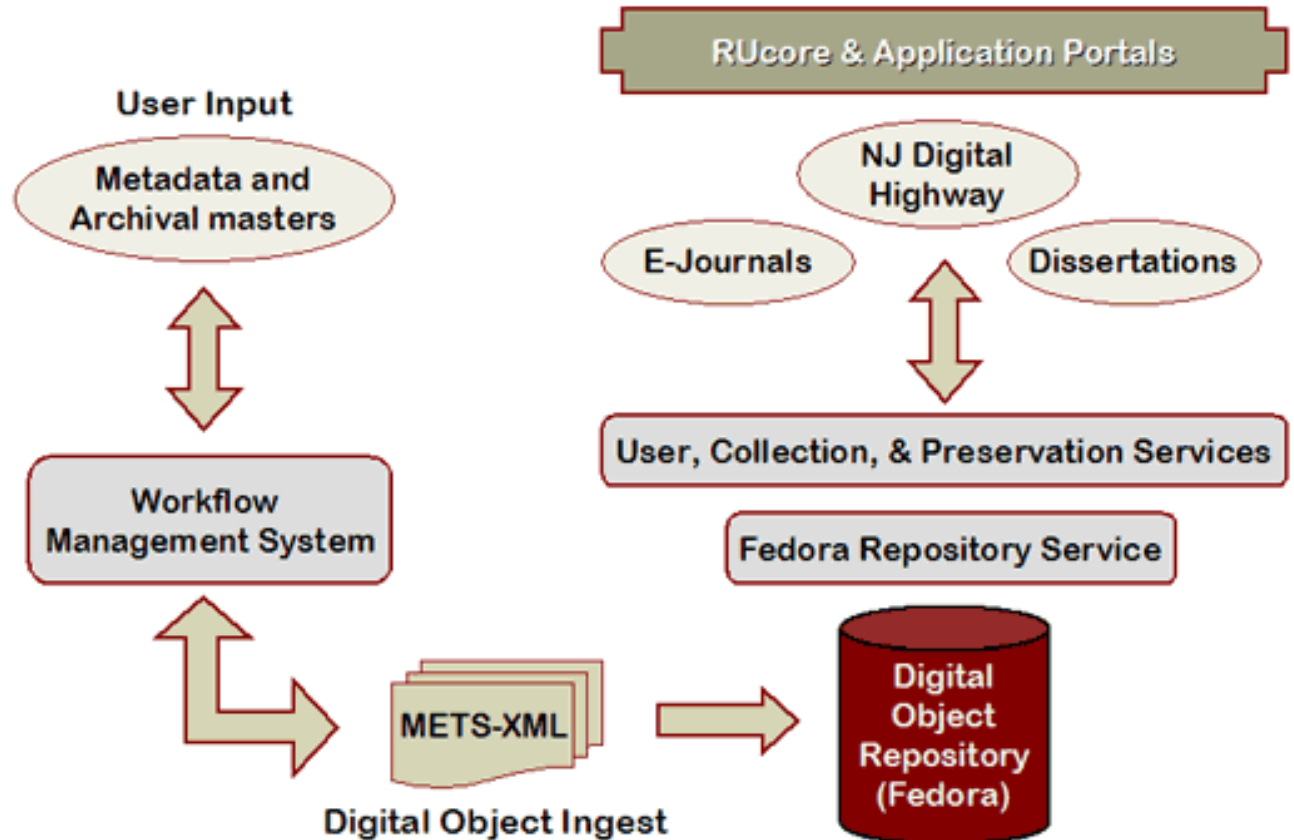
A final requirement was to support some level of customization for participants. We were not able to anticipate the vocabulary needs of all participants for populating metadata elements. We also needed to support local decision-making about data elements to display to their catalogers. We added a template capability to enable participants to select mandatory and recommended data elements for display and to add default values for data elements. Default values for technical metadata are particularly useful, since resources are either digitized to standard technical specifications or created as born digital objects in a standard digital format. The template capability allows users to customize the look and feel of the metadata input to suit their needs. Given the complexity of the RU metadata implementation, this was a critical feature. The template enables organizations to use the complex data architecture iteratively-adding from the large array of available data elements over time, as their expertise grows or their information management needs change.

The Workflow Management System was initially developed beginning in 2003 to provide a robust and intuitive user interface for the Fedora repository system. Its initial design reflected its dual purpose for supporting the Fedora architecture and the needs of a diverse group of libraries, museums and archives with cultural heritage resources. As the WMS developers began to speak about the WMS in a wider environment, interest among other organizations in using the WMS intensified. In 2006, the Library of Congress Motion Picture, Broadcasting and Recorded Sound Division contracted with RUL to develop a bibliographic utility to support the moving image archives community, as part of its MIC (Moving Image Collections) project. [6] In the course of WMS development, RUL began thinking about adding modularity and additional customization features to the WMS so that it could be used as a stand-alone application or integrated with other repository architectures by a wide range of organizations.

The features and functionalities associated with WMS have been constantly growing, and WMS is moving towards becoming a generic integrated digital object workflow management system to be released to public as open source package in early 2008. The obligations imposed by open source required that the design of WMS must be flexible to meet different needs of diverse users; the functional components must be highly modular for software developers to easily add or remove; and the software must be readily maintainable. At Rutgers, the WMS currently serves as a front end for several user applications, the New Jersey Digital Highway, the Rutgers Community Repository (RUCore), the faculty deposits module, the open journal platform and the electronic theses and dissertations (ETD) module. The WMS is a cornerstone application for the RUCore cyberinfrastructure.

Figure 5: Role of the WMS in the RUCore cyberinfrastructure

RUcore - How it Works



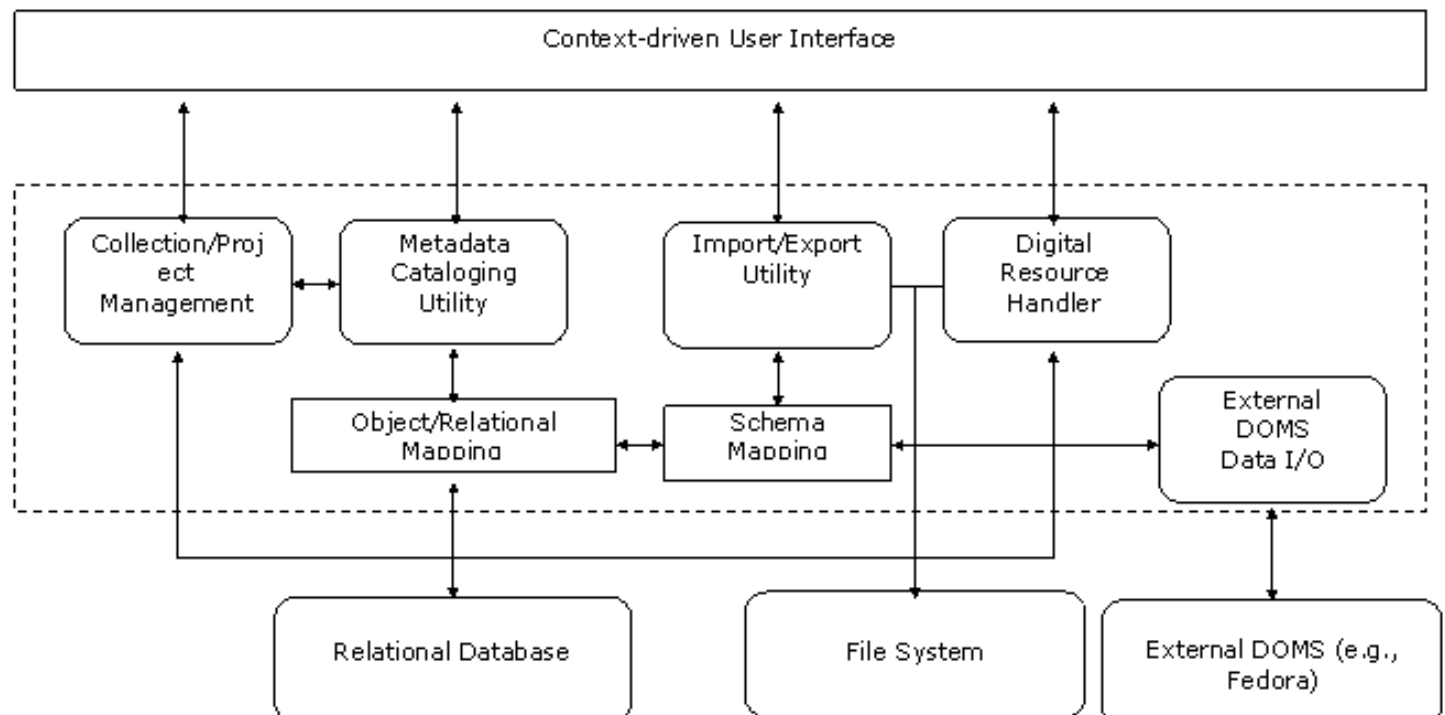
Capabilities of the Workflow Management System

The Workflow Management System provides a complete object ingest and metadata creation system, with services to ingest objects and metadata and to export these objects and metadata, individually and in bulk. The WMS consists of object handling, which includes object ingest, object structuring for any format (image, text, multimedia), and object reformatting. The pipeline application that creates multiple formats from the ingested digital master file is very RUL specific, since we currently use a PDF server application as the “middleware” for reformatting multiple access formats for text and images. This capability has been modularized and abstracted from the WMS so that organizations can integrate their own application, which may be JPEG2000-based, for example. Object handling includes the ability to ingest transcripts and to provide OCR for text and transcript files. The WMS provides a full METS metadata architecture that supports access, discovery and management of an information resource in all its manifestations, from analog or born digital source objects to digital technical masters and access copies. Local customization includes authentication and authorization for managers and metadata creators; the ability to customize and add vocabularies to data elements; and the capability to customize the look and feel of the metadata input and to add default values to data elements. Objects are organized within collections and sub-collections. Hierarchical organization provides intrinsic relationships among objects (such as articles within the issue of a journal) and enables objects to “inherit” collection level information via templates, such as a rights event for the collection level deed of gift. This ensures that any collection level context is always readily available at the individual object level.

The Architecture of the Workflow Management System

Architecturally, the WMS is a container that includes many functional modules. All WMS modules follow the three-tiered design pattern that separates the user interface and persistent data storage from the business logic.

Figure 6: Architecture of the WMS

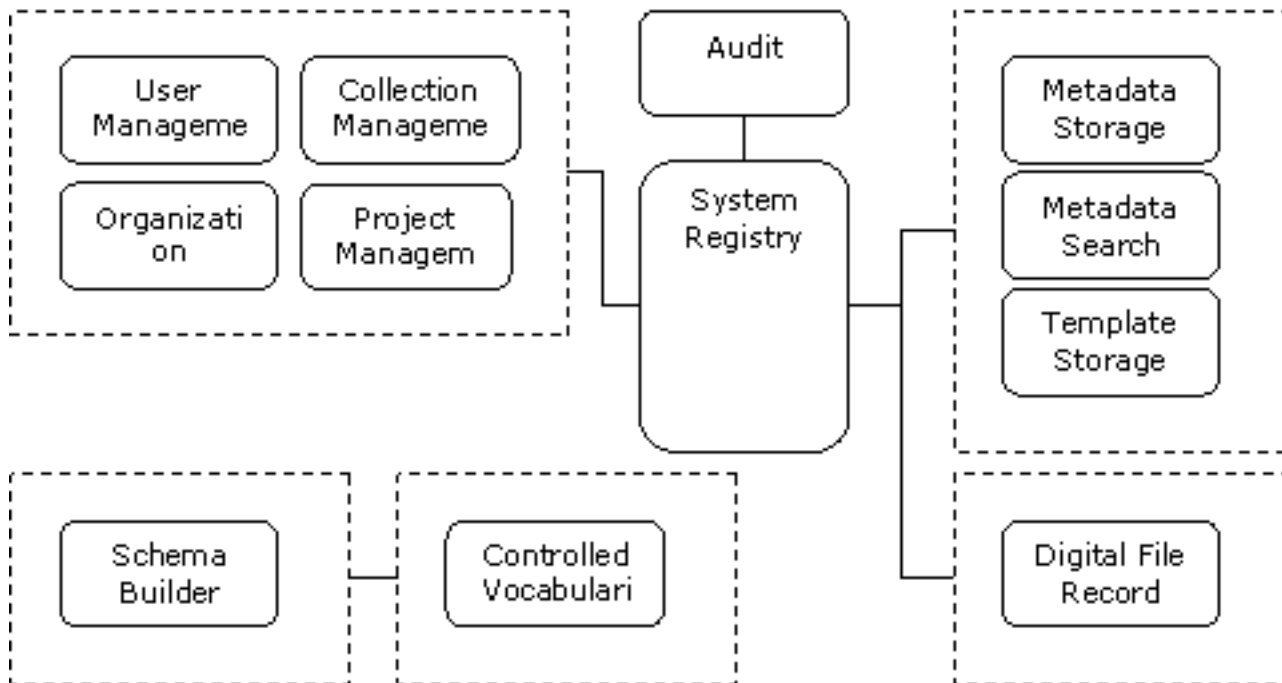


As figure 6 demonstrates, the WMS can be conceptually divided into three layers - the context-driven user interface layer, workflow policy/business logic handling layer, and persistent data storage layer.

WMS Database Design

The internal persistent data storage includes a relational database and file system. This layer does not directly deal with workflow logic, but its design, especially the schema design for the relational database, affects the data integrity, the system performance, and the overall ability for the system to function.

Figure 7 WMS Database Design



The WMS relational database schema is built upon the functionality class of the data. As Figure 7 demonstrates, there are 5 major functional blocks in the schema: WMS collection management, digital file handling, metadata and template storage, metadata schema builder, as well as controlled vocabulary management.

In addition to the standard relational database design considerations, 2 important decisions were made for the WMS database structure design. One is that digital object metadata is treated as an XML document as a whole instead of as a hierarchical data set, with a separate searching table for selected indexed terms. This approach ensures that the WMS database structure does not depend on any specific metadata schema; increases the overall system performance; and makes it much easier to port existing data to XML-enabled database systems to utilize their XML searching capabilities. The second decision concerned the schema builder. The schema builder block stores metadata input form layout information as well as the associated XML structure for each metadata element. This method significantly reduces the amount of work needed for web form maintenance, and provides the project manager with some capability to build and edit the metadata schema from a web input form.

Workflow Modules

The workflow logic layer is where all the workflow policy and business logic handling take place. This layer can contain as many functional plug-in modules as needed. Currently, the following functional modules have been developed:

- Authentication/authorization
- Collection/project/user management
- Digital resource handling
- Metadata cataloging utility
- Metadata schema mapping utility
- Digital object batch import/export utility
- External digital object access module

Each module operates independently from one another, but they share the following common design patterns:

- Object-oriented software design. Though the WMS is written in PHP, the design of the software uses an object-oriented approach. Object/relational mapping is used as the solution for data access.

- Database driven application. A relational database structure forms the foundation for all WMS modules. Many of the modules, especially the metadata cataloging utility, heavily depend on information retrieved from the database for data processing and form rendering.
- XML as both data model and data carrier. The WMS uses XML to model internal data as well as exchange data with external resources.
- PHP session handling. The session matrix is used for all modules for keeping track of state information.

The metadata cataloging utility is one of the core modules and is the most heavily used module in the WMS. Currently the metadata entry forms use a METS compliant element structure. The descriptive metadata is built upon MODS and the technical metadata utilizes data elements from diverse schemas such as PREMIS [7], MIX [8], and the forthcoming AES-X098 metadata standard for audio resources. [9] The WMS can be easily changed to fit other metadata schemas through the use of the schema builder configuration utility. The WMS was intentionally designed to be schema independent and thus to support all metadata schemas. This is important because metadata is still an emerging area and schemas to support different communities and types of resources continue to proliferate. Metadata entered are stored using the internal XML data model, and conversions between different metadata schemas are done through a custom-built XSLT script. This design pattern makes the cataloging utility flexible and efficient for both utility users and software developers. The mapping architecture for both data import and data export is key to supporting interoperability with other communities, through data sharing using the OAI-PMH (Open Archives Initiative-Protocol for Metadata Harvesting) [10] or federated searching.

Digital resource handling deals mainly with digital file formats and conversion between them. WMS was initially tied to the RUL digital file handling policies, which require TIFF as the archival master for images, and DJVu, PDF, and jpeg for presentation formats, for example. DJVu, which is a very functional but proprietary and not widely used digital file format, served as the intermediary format for the conversion of digital master files to multiple presentation formats, particularly JPEG and PDF. RUL is in transition for image presentation formats as we integrate Adobe PDF server into file handling, particularly to support faculty deposits and ETDs, and as we evaluate migrating from TIFF to JPEG2000 as the RUL master format for digital images. The goal now is to abstract this file transcoding capability and enable the individual institution to implement file handling capabilities specific to its needs.

The metadata schema mapping/import utility allows batch import of metadata and digital resources into the WMS system from an existing external database or files. Conversion between schemas is achieved with XSLT scripts custom developed for each specific schema. The import utility currently supports mapping and conversion from plain text bibliographic records to MODS, Dublin Core, and the RUL metadata schema used in WMS. PBcore and MPEG-7 are currently in development, particularly to support the MIC bibliographic utility. There are no limits to the number of mappings that can be supported, other than the limits on the cataloger time and effort needed to develop the mappings. A metadata mapping facility developed for the MIC project enables any participant in the MIC union catalog to input the data elements from the institution's unique schema, map data elements to MIC data elements, test the mapping through a sample record load and then batch load their metadata. The mapping is stored for the organization's future use. A MARC mapping exists as a standard map. This mapping functionality was critical for MIC because many moving image archives utilize their own custom metadata schema. This mapping utility will migrate to a future version of WMS to enable the WMS owner to map metadata that doesn't conform to a schema recognized by the WMS. This is particularly critical for supporting university faculty, who often create their own metadata implementations to provide access to their research. The export module converts the metadata and digital resources in the WMS into a user specified format, then stores as files or exports to an external digital object management system, such as Fedora. Currently the WMS exports in METS and Fedora Object XML (FOXML) format. MARC is under development.

For the software developer, the key to maintaining the flexibility of the WMS is to follow the WMS modular design pattern. Using the common module libraries provided with the WMS core release, additional modules could be easily plugged into the system, for example, the Fedora object editing module developed for the Rutgers University RUcore repository, which provides for metadata or object editing after the object and metadata have been ingested into Fedora.

User Interface

The aim for the WMS user interface design is to provide a simple and logical workflow pattern that a cataloger or someone uploading digital objects can easily follow. At the same time, the interface code needs to be easily maintained and modified, particularly since requests for changes in the WMS have mostly involved the user interface. Several measures have been taken to fulfill this goal:

- WMS interface is dynamic and context driven. A user can configure the application for what he/she needs. Depending on the user's choice during the working process, the WMS only displays sufficient information for him/her to accomplish the task.
- Template for metadata cataloging. The template can be easily created, edited, enabled, or disabled, at the collection, project, or personal level.
- Context driven help is provided for metadata cataloging.
- WMS forms are designed to be clean and simple. Distracting elements, such as over-decorating, audio, flash video, etc., are avoided.
- WMS user interface html code is dynamically generated using a handful of code templates, based on the business logic provided to the utility.

WMS System Requirements

The WMS is written in PHP, and it depends on a relational DBMS to function. The WMS runs on PHP 4 and above, and theoretically any RDBMS, commercial or open source, can be used with WMS. However, considering the needs of open source software users, we focused on testing the open source RDBMS only. Currently it has been tested to run without problems on the 2 most popular open source RDBMS, MySQL (4.0 or above) and PostgreSQL (6.0 or above). WMS can be run on UNIX, Linux, or Microsoft Windows system with any web server, as long as it is configured to support PHP.

Re-engineering the WMS to Support Open Source

Open source or free software? The two terms are sometimes used interchangeably, but the difference between the two concepts is significant: for both open source and free software, you can download, install it on your system, and run it without a fee. However, for open source software, you can also read the source code and modify it to suit your needs. You can also build upon the code to create new applications, as long as you abide by any licensing or copyright restrictions.

From the software development point of view, providing open source software to the public implies fundamental changes in design decisions compared to creating proprietary institutional utilities, simply because the user base and their needs become much larger and unpredictable. We can no longer expect detailed requirement specifications and quick, convenient interactions with the potential users. Things that seemed natural to one institution or organization, such as institutional policies that usually are embedded in the proprietary software, become problems to others. The commercial software that would have naturally been used in a proprietary utility for one institution may not be available to others. The operating systems and supporting software for the utility that worked fine with one institution could be unavailable or out-of-date for others. Open source means developers all over the world can modify or contribute to the code. This will exponentially increase the code maintenance and documentation issues and tasks. Therefore, moving from a proprietary institutional utility to open source software doesn't mean simply making the utility available for anyone to download. It is a whole new concept and usually means the software needs total redesign and development.

WMS was a typical proprietary institutional utility when the project was started. In order to convert the utility to open source software, WMS has undergone a total redesign since the beginning of 2006. The design philosophy is that the WMS needs to be flexible enough so that it:

- does not depend on any specific external digital object management system (Fedora, DSpace, etc.)
- does not depend on the policies of any specific institution
- does not depend on any commercial software product

The design philosophy must also ensure that it is easy for individual institution to add functional modules and to customize for their needs, while being easy to maintain and support.

The redesigned WMS is no longer a utility tightly coupled with Rutgers University Libraries, it is now an open source digital object workflow management framework. Under this framework, the WMS development team provides core modules for WMS to perform basic functionalities and code libraries for partners to use for extending the capabilities of the WMS. The core modules conform closely to this design philosophy. They are highly modularized and can be reassembled as needed. The tasks for which each institution or organization will most likely have its own policies, or want to have its own implementations, are moved from hard-coded software implementation to software-supported configuration decisions. The handling of digital resource files gives a good example of this scenario. Some institutions want TIFF to be the archival master file format for images while others might prefer JPEG2000. The WMS no longer forces people to use TIFF but instead hands over the decision to each individual institution, allowing each institution to decide what format to support and what software to use to create each format. With respect to metadata, the metadata schema builder module is an attempt to decouple the utility from any specific metadata standard. Creating a one-for-all metadata converter is very difficult, and while we are working toward this capability, we haven't yet reached it. However, creating a utility that enables users to significantly modify the existing schema and associated HTML form input fields has proven to be totally feasible.

Documentation is a big issue. There are 2 kinds of documentation related to a piece of software: documentation for the user of the software (e.g., a user manual), and documentation for the developer or potential developer/supporter of the software. For open source software, both kinds of documentation are very important. RUL's current user manual includes a full data dictionary for every data element in the METS metadata architecture and a tour of the WMS, demonstrating both navigation and the purposes and use of each METS metadata document. The user manual reflects a much earlier version of the metadata, which predates the rights metadata schema and the use of descriptive events in the descriptive metadata. The user manual will need to be completely rewritten before the WMS software is released. The RUL metadata architecture is based both on METS and an event structure in each METS document to capture the lifecycle of the resource in many dimensions. These are somewhat novel concepts for catalogers, who have mostly focused on the descriptive information captured in MARC or MODS. Our experience has been that the training and support for initial metadata users within the *New Jersey Digital Highway* has been significant, involving hands-on training and considerable remote support. Training tools that encompass the user manual and beyond need to be produced before the WMS is offered as open source. This is not an easy task and is the subject of much discussion at present. The documentation for the software architectural and coding details needs equal attention because of the need for developers to understand the code to add modules or customize options for each organization. This task inevitably falls exclusively on the software architect and developers.

The WMS Open Source Process

The Rutgers University Libraries are moving cautiously into the open source environment with the Workflow Management System, primarily due to the sophistication and complexity of the metadata architecture and the modular, customizable design. Changes to the WMS are incorporated in RUcore versions, which are specified and implemented at least twice and sometimes three times per year. While versions are well documented for programmers and stringently tested, user documentation lags significantly and is generally several versions out-of-date. This has not been a real problem to date because the Rutgers staff using the software generally specified, approved and tested the modifications to the WMS. In the future, when the number and composition of users is unknown, user documentation will be very important. RUL is discussing how to handle documentation requirements when no position with a specific assignment to provide documentation currently exists. RUL also currently develops version specifications based on the needs of Rutgers users, as articulated by library users, or dictated by grant requirements, and as authorized by the RUL cyberinfrastructure steering committee. RUL utilizes a modification request process that identifies bugs; either during testing or during routine use, and incorporates fixes for any bugs that don't actively interfere with standard use into future WMS versions. RUL is discussing how bugs identified by open source users, as well as recommendations for enhancements (or enhancements created by open source users), can be effectively incorporated into the WMS development and testing workflow. Initially, RUL is exploring the development of a small user community of institutions with similar needs, particularly institutions currently using the Fedora repository architecture, who can provide testing and collaborative

development for the WMS. Currently, RUL is discussing an open source development collaboration for the WMS with Northwestern and Penn State, all of whom are currently testing both the installation and the functionalities of the [open source bibliographic utility based on the WMS](#) that will be provided to the Library of Congress in 2008.

Future Developments for the WMS

One major development for FY08 is the integration of Encoded Archival Description (EAD) finding aids into the WMS. This integration will support both ingest and export of metadata as an EAD finding aid with associated digital objects. EAD support is critical because the special collections and archives of the Rutgers University Libraries base their description and access practices exclusively on finding aids. In addition, a document management capability will be added to enable users to automatically store and associate relevant PDF documents with events, such as a deed of gift with a rights event or a bill of sale with an accession event, within an administrative documents section of the repository.

Rutgers University Libraries are also recipients, with William Paterson University and NJEdge, the statewide Internet2 utility, of an Institute of Museum and Library Services grant to build a statewide digital video network, NJVid. RUL will be extending the WMS METS structure map to enable faculty and K12 educators to easily segment and annotate videos via a simple web form. RUL will also use XACML to place constraints on object use, again based on a simple web form completed by faculty, to enable faculty to reserve video course lectures to users within a course. NJVid will also implement a statewide Shibboleth facility that provides centralized Shibboleth services to all participating organizations, regardless of technical readiness for Shibboleth support. Digital video functionalities will appear over the course of the grant, from FY08-FY10.

Conclusion

There are a number of open source bibliographic utilities that support the creation and organization of digital information for libraries and archives. We feel that the WMS makes several unique contributions, including the event-based data model; a complete METS implementation with fully functional METS documents for description, source, technical and rights metadata; and XML-based customization capabilities that enable users to tailor metadata to their needs while still supporting metadata standards and interoperability. RUL has also extended and enhanced technical and source metadata for digital multimedia, to support the needs of the moving image archives community. We feel the WMS will be a strong option particularly for organizations that want to document both analog source objects as well as digital surrogates and that want to document and manage resources with copyright and other digital rights issues. We also feel it is a strong utility for organizations wanting to ensure interoperability with other initiatives, even as they customize metadata to meet local needs. We are addressing the remaining issues, that are primarily organizational and policy driven, to enable release of the WMS in open source in early 2008.

Notes

[1] "Presentations Given about RUcore" *Development of RUcore*. Accessed 1 November 2007. <http://rucore.libraries.rutgers.edu/collab/index.php>

[2] Fedora Commons. Accessed 14 October 2007. <http://www.fedora-commons.org/>

[3] *New Jersey Digital Highway*. Last updated 24 September 2007. Accessed 13 October 2007. <http://www.njdigitalhighway.org/>

[4] Library of Congress. *METS-An Overview and Tutorial*. 13 September 2006. Accessed 1 November 2007. <http://www.loc.gov/standards/mets/METSOverview.v2.html>

[5] RUcore-Rutgers Community Repository. Accessed 1 November 2007. <http://rucore.libraries.rutgers.edu/>

- [6] Library of Congress. *MIC-Moving Image Collections*. Last updated 25 September 2007. Accessed 13 October 2007. <http://mic.loc.gov/>
- [7] Library of Congress. *PREMIS-Preservation Metadata Maintenance Activity: Official Web Site*. Last updated 31 July 2007. Accessed 13 October 2007. <http://www.loc.gov/standards/premis/>
- [8] Library of Congress. *MIX-NISO Metadata for Images in XML Schema: Official Web Site*. Last updated 17 August 2007. Accessed 13 October 2007. <http://www.loc.gov/standards/mix/>
- [9] Audio Engineering Society Standards Committee. October 2003 Meeting of SC-03-06. c2007. Accessed 13 October 2007. http://www.aes.org/standards/b_reports/b_meeting-reports/aes115-sc-03-06-report.cfm
- [10] Open Archives Initiative. Open Archives Initiative Protocol for Metadata Harvesting: Protocol Version 2.0 of 2002-06-14. Accessed 13 October 2007. <http://www.openarchives.org/OAI/openarchivesprotocol.html>

About the Authors

Grace Agnew, Associate University Librarian for Digital Library Systems and WMS metadata designer, Rutgers University Libraries ([gagnew at rci dot rutgers dot edu](mailto:gagnew@rci.rutgers.edu))

Yang Yu, Database Architect and WMS Architect, Rutgers University Libraries ([yangyu at rci dot rutgers dot edu](mailto:yangyu@rci.rutgers.edu))

Communicat: The Next Generation Catalog That Almost Was...

by [Ross Singer](#)

As the pursuit of the Next Generation Catalog (NGC) gains momentum, librarians and libraries are frequently looking outside of their traditional integrated library systems to achieve the kinds of functionality needed. Whether they be alternate search and discovery interfaces, such as Endeca, the University of Virginia's BlackLight or Ex Libris' Primo; social software applications such as PennTags or SOPAC; or combinations of the two such as Scriblio and VuFind, it becomes apparent that most libraries have a different definition of "next generation" and that few believe that "next generation catalog" has to be a component of the historical backbone of the modern library: the OPAC.

Georgia Tech's library chose to take on many Web 2.0 concepts in its vision of the NGC: tagging, annotations, recommendations, faceted search, etc. The main focus of the project was to allow users and groups to collect their own personal libraries and use that community-based collection as the source of the central catalog: the Communicat. As users aggregate, annotate and share the resources they find and use, this in turn would expand and inform the larger Georgia Tech collection. The library and librarians would continue to add value, enhancing the metadata and content around the resources gathered that fit the scope of the institute's mission.

Envisioning a Communicat

Roughly two years ago, the library formed a committee to work on this project. It was an informal group, with membership from Systems, Reference, Cataloging and Digital Initiatives, and had no clear objective or mandate. The main goal was to brainstorm ideas of what an ideal "catalog" would look like. A recurring theme was the rigidity of the incumbent catalog (Ex Libris' Voyager) and the absence of any semblance of community control or input to what the library defined as its "collection". The integrated library system had no capacity to evolve as new sources of information appeared about existing items or to build connections between items in the collection or with objects on the web. While the group did not actually produce anything substantive, it did manage to document many concepts and desires collectively via whiteboards and helped set the priority for how to proceed with such a project. Through the brainstorming, the essential services and their project names were conceived.

The major points uncovered were:

- The "catalog" does not begin to address the free web in any serious capacity.
- The de-emphasis of traditional cataloging (and subsequent de-emphasis of the cataloging department) requires a decentralized approach to metadata aggregation, composition, and maintenance.
- Relationships between items can be inferred by how they are used. If two resources are used for a particular project, there is a much stronger chance they have something in common. Explicit relationships between items should be set when appropriate: a movie version of this book; the soundtrack to this movie; the following proceedings appear in this conference. There was no intention to *merge* these records, but merely to make the connections between them clear.
- The scope of the "collection" is strongly dependent on the context in which the user is searching. If the user is a physics graduate student, weighting the resources that are defined as physics or sciences over, say, sociology would be desirable. There are times that searches would start "context-neutral", but the ability to focus on particular subject areas, especially as the collection grows, could offset large result sets.

As these ideas began to take shape, some requirements for what such a system would need to be able to do became clearer. Versioning was essential; if authorship of metadata were to be distributed, a mechanism to roll back to a previous, stable snapshot would be necessary. Access control would also be important. For the integrity of consistency and discoverability in the "main" collection, some elements (such as the original MARC, Dublin Core or ONIX record metadata) would need to be controlled, but other aspects of the record would still be editable.

There also needed to be a separation between the machine-readable and highly structured elements and the public display interface. While the MARC records, et. al., would help define the display record, the public display would not merely be a literal representation of MARC (or ONIX or any other standard). The metadata records could then be preserved, without danger of being adulterated; yet the discoverability and readability of the resources could be altered as data and needs surrounding an asset changed.

Such a project also required a sense of users, groups and permissions. Not only would groups (or people) likely prefer that non-members not be able to manipulate the groups' items and content, that sort of control would probably be necessary within the groups themselves as well, especially larger groups. This necessitated the concept of role on top of user and group. The downside of such granularity is that it generally makes interfaces to maintain resources more complicated, but the alternative, in the committee's opinion, was too limiting.

The brainstorming work also provided a handful of desired services that could be broken out into individual projects that, together, would make up the Communicat. First, there needed to be a means to aggregate references to resources to add to the collection. Social bookmarking services, such as del.icio.us and especially Connotea and Cite-U-Like, were looked at as models to draw from. Given the need for more robust metadata, however, it was decided that utilizing the OpenURL link resolver to gather data from appropriate sources would be easiest way to meet this requirement.

Obviously, a system to store, index and manage these resources was also needed. It had to be able to deal with a variety of metadata schemes and flexible enough to accommodate data that it might not understand or be able to use. This data store also needed to be accessed through several different interfaces and act upon or display its contents accordingly. This repository was the largest and most important decision, since it influenced much of the direction of the project.

The other major service the group envisioned was the means to be able to organize one's resources into various lists or categories. This would open the door for faculty to create their own reserves lists; librarians to create subject and course guides; and users to be able to make web-accessible bibliographies and reference lists. A possible outcome of this service was a concept the committee called *Research Trails* in which, as a user compiled and annotated resources for a given assignment or project or article, he or she created a bucket of information of *all the resources* discovered while researching their topic. While only a subset of these items would appear in the bibliography, the resources that led to the discovery of the actual materials cited would still be associated with the final product. The researcher in essence would, through the course of their work, make associations between our resources for us.

The goal was for any service to be independent of any other. The library could enable the personal library space without the community collection piece or the link resolver or any combination thereof. If the repository became too demanding to implement, it could be replaced with a simpler incarnation that met the revised needs of the users.

Putting the Vision into Place

Prioritizing the pieces of a project like this was non-trivial. There was only one developer, me, that could work on this project. It was difficult to commit significant resources towards it given its highly experimental nature, size, complexity and, frankly, its potential for failure. To devote time and people to a project that ultimately might not work or be largely ignored by our users runs counter to the traditional risk-averse culture of libraries. Also, like so many other libraries, Georgia Tech has a backlog of needs that also must be addressed, further straining the commitment that the committee could make on the Communicat.

With these conditions in mind, it was decided that it would make the most sense to begin with a piece of the project that solved other library needs outside of the requirements of the Communicat. The link resolver enhancements, known internally as the übeResolver, were targeted as a good starting point, since they also addressed problems we had resolving print materials and, possibly even more importantly, solved a major issue with conference proceedings. If this service was successful and obvious benefits could be seen in its development, then its success could be used to garner the political capital needed to prioritize development of the rest of the project.

Putting the Ü in OpenURL

The link resolver seemed the logical place for users to enter their data into the Communicat. The preferred variety of resources submitted to the system were scholarly (at least for the sake of the larger goals of the project), and theoretically, the OpenURL link resolver (at Georgia Tech, Ex Libris' SFX) *should* be a nexus for citations as the researcher searches in the library's licensed databases. Quite a bit of information can be gathered from the resolving process: where the user came from, what the user wants (eliminating the need to enter detailed metadata by hand or via screen scraping) and where they went. In practice, however, OpenURLs' metadata can be rather spotty, so one of the goals of this service was to improve that as well as enable discovery of open access materials in repositories on the web.

What Tech developed was the Ümlaut (the name übeResolver was deemed too self-aggrandizing): an OpenURL middleware layer written in Ruby on Rails. The goal of the Ümlaut was not to replace SFX, since it handled the task of linking to full text resources quite well, but to enhance it, accessing SFX via its XML interface.

The Ümlaut queries Crossref, Pubmed, Georgia Tech's catalog, the state union catalog, Amazon, Yahoo, Google and a host of social bookmarking sites for each incoming OpenURL to gather the most complete metadata snapshot of a given citation. Through Google's and Yahoo's web search APIs, it also analyzes the links to determine if any point to free manifestations in open access archives and present its findings along with the vended full text links.

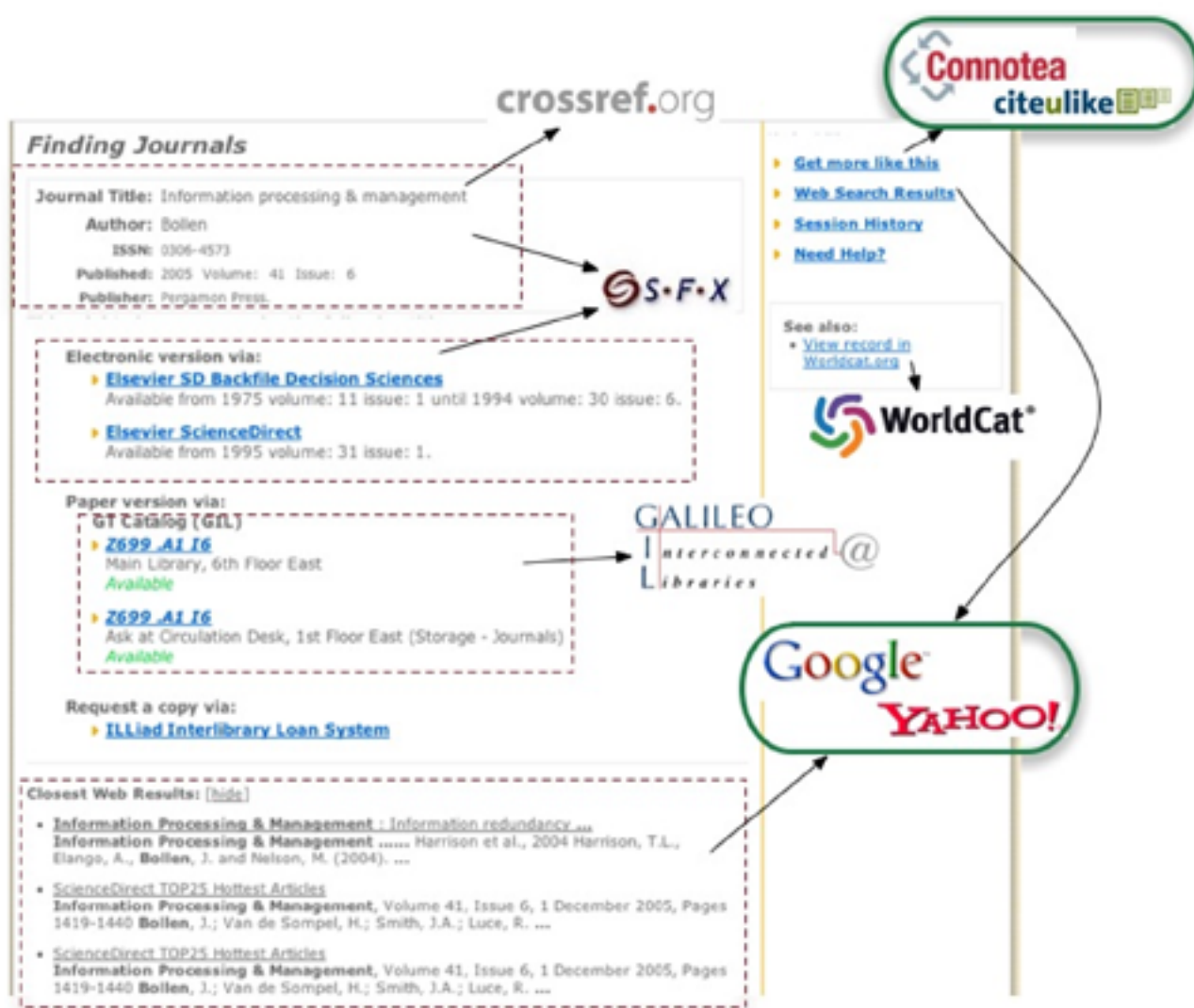


Figure 1. Diagram of Ümlaut Workflow

From the catalogs, it pulls MODS records (if the item was found), which offer much richer metadata than what is generally available in the OpenURL context object. By leveraging OCLC's xISBN service, other manifestations of a particular work could also be found rather than relying on the exact ISBN sent in the OpenURL request. More

sophisticated searches can be performed for different objects, further increasing the success rate for items such as conference proceedings.

Since the Ümlaut was to have fit into the broader context of the Communicat, machine-readable XML and JSON interfaces were created and exposed via unAPI. This allowed the Ümlaut's services to be consumed by the OPAC or other applications. When the time came to develop the bookmarking service, resources could be enriched in the background without user intervention.

Development of the application proceeded quickly, going from proof-of-concept to production service in fewer than five months. Outside of the expected fits and starts of launching a new, heavily used resource, the Ümlaut was generally considered a success and the value it added over SFX offset its somewhat sluggish performance. The few months after its August 2006 launch were largely consumed with enhancements and bug fixes, and by January of the next year, planning had already begun on the next version, dubbed Ü2. The successful implementation and subsequent national attention of the Ümlaut (it was the winning entry in the 2006 OCLC Software Contest) gave us the credibility with the rest of the library needed to proceed with the next phase of the project.

In Search of a Repository

Now that we had a means to collect metadata, it seemed appropriate to find a place to store it. The challenge was to locate an existing system that was both extensible, versatile and, ultimately, replaceable if the needs or direction of the Communicat were to shift dramatically. As a last resort, the metadata store could be built from scratch, but we felt this was less desirable given the time it would take to design and develop and the fact that it would simply be easier to modify an established project and possibly alter our expectations than make something new.

Early in the process, before the more inspired goals from the whiteboard were drafted, the library considered using Unalog, a Python-based social bookmarking engine based on the Quixote web framework. While Unalog's current implementation was rather modest in scope, Dan Chudnov, the project's creator, had a desire to redesign its data model to utilize more detailed metadata, such as MODS. While appealing, the scalability of such an endeavor was unclear and, ultimately, the necessary refactoring never was made.

The nascent design ideas through this point had an assumption that the data would be stored as MODS, since it was more detailed than Dublin Core, but considered easier to work with than MARC. The Evergreen ILS was also briefly considered for the data store; Tech was interested in it as an ILS and it used MODS internally as its data model. The shortcomings of Evergreen (lack of acquisitions module or serials support) were a non-factor, since it was only going to be used as a bibliographic database. It, too, was eventually considered inappropriate for the expanding requirements of the project and was abandoned.

Through continued discussion, the expectation of using MODS was dropped and focus turned to more flexible repository frameworks. Georgia Tech's Digital Initiatives department had considerable experience with MIT's DSpace, but its architecture was deemed too monolithic to apply to all of our use cases. Fedora looked more attractive, as did the JSR-170 based repositories, such as Apache's Jackrabbit. Either of these solutions would have required a considerable amount of development time to implement.

In the end, we finally settled on Outerthought's Daisy Content Management System. While a CMS may seem an odd choice at first for such an application, Daisy has a unique design that suits itself well for alternative uses outside of traditional web page publishing. It actually is two separate Cocoon-based services: a backend XML document repository using MySQL and an optional front-end wiki interface. The document repository can be used entirely on its own and has Java, Javascript and HTTP APIs to access and manage its collection. The documents themselves are defined by schemas which are made up of various internally defined *types*: *parts*, where content is defined and stored, and *fields*, which are basically triples defined as strings, booleans, dates, long integers, etc. These are combined with *document metadata*, such as title and identifier to define the schema. Schemas can be changed after data has been entered using them, but fields cannot be deleted if any document is using them (although they can be marked *deprecated*).

Daisy maintains every version of a document, making it possible to roll back to a previous iteration. It also has the concept of branching a document, which is intended for having multiple editable versions of the same document if, say, the content of a document needs to be different based on the context in which it is being viewed. Coupled with yet another axis that Daisy documents can branch on, language, this can make for quite a complicated model. For our purposes, however, simple versioning would be sufficient.

Pushing Daisy

Now that we had identified our repository backend, it was time to build a framework to interact with it and model our data to make it work effectively within said framework. Since we wanted a layer of abstraction between the Communicat and its underlying data, we created another Ruby on Rails application named Cortex to interact with Daisy's ReSTful HTTP API and manage the documents for indexing and display. Cortex was also tasked with manipulating MARC data to enter into Daisy, harvesting and ingesting Dublin Core records from OAI-PMH servers as well as normalizing and storing data from other sources (i.e. ONIX from the publishers). This made for a rather complicated metadata model.

The Daisy document schemas were divided into two groups: *component* documents, which stored the original metadata records; and *page* documents, containing the publicly viewed data as well as the association of the viewable record to its components. Page documents also defined the relationships between themselves and other page documents. Component documents held the original metadata, whether they were MARC21, Dublin Core, ONIX, OpenURL or whatever was available. For data for which no Component schema had been created, there was a *GenericComponent* so the resource could be preserved until it could be identified.

To aid with indexing and retrieval, important metadata - such as author, subject, ISSN/ISBN etc. - was stored in fields in the Component document. The page document held more high-level data about the resource, such as what its item type might be (serial, book, etc.) or whether or not it was a conference or a government publication. All pages stored the title, this being a requirement of Daisy. The relationship to the component documents was kept in a repeatable *link* field. In many ways, the page documents were very similar in philosophy to RDF, making connections between documents through URIs.

This architecture meant that for each item in the catalog, there had to be at least two Daisy documents. Phase II would require at least one more document per record as authors were to be split from the item records as *Identity* records. Communicat user records would also be Identity documents, so users could be associated with any works they may have created.

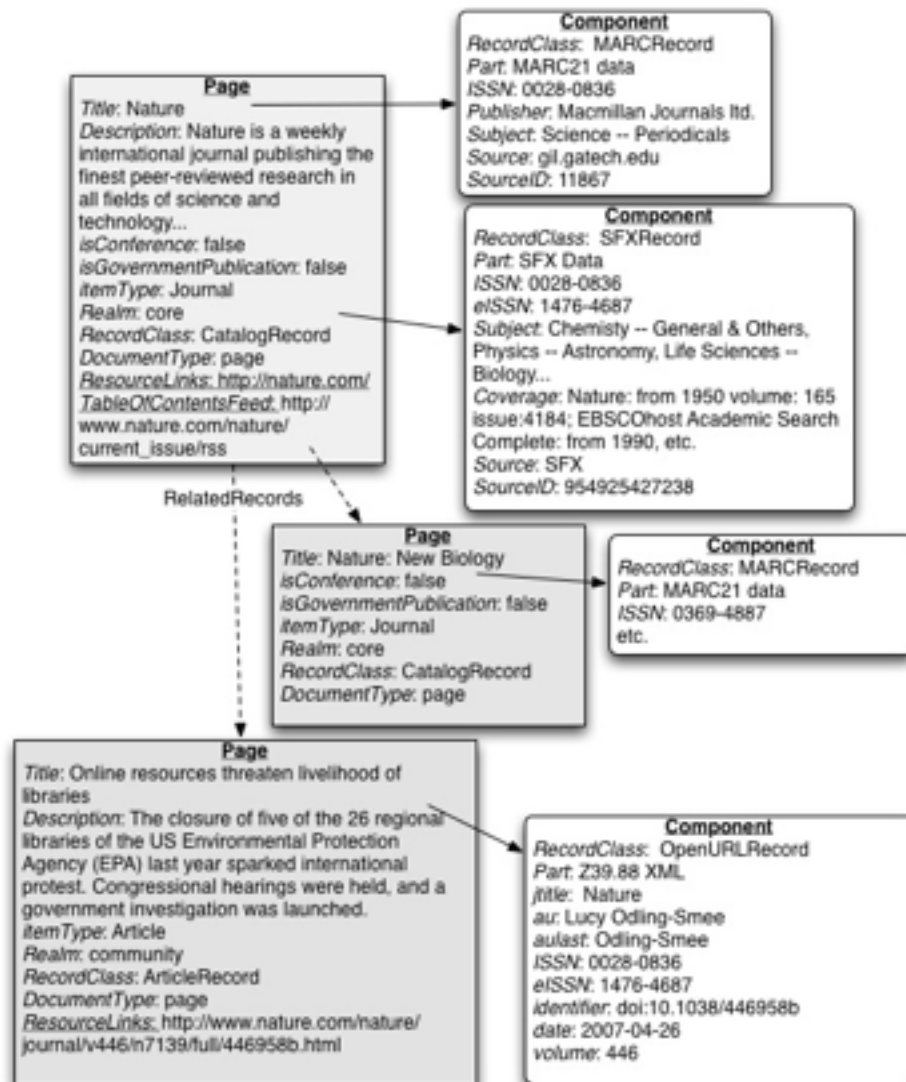


Figure 2: Schematic of Daisy table design.

The relevant documents surrounding a particular resource (a public Page document, its requisite component documents and identity records) were merged and indexed in Solr. The individual components were also indexed in Solr, so besides powering the main catalog search, Solr could also be used for administrative and maintenance tasks. Managing the relationship between Solr documents and Daisy documents was another responsibility of Cortex.

A fourth planned document type was the *BookmarkPage*, essentially a page to a page document. These documents would be associated with users (Identities) and contain annotations, tags and comments. Through BookmarkPages, users would be able to organize and share the item level page documents. Public bookmark annotations would be searchable alongside the standard bibliographic data, allowing dialog about resources to aid in their discoverability.

In an effort to allow users to control the scope of their searches, Communicat documents are assigned one of three *Realms*: *core*, *community* or *world*. The realms break down rather simplistically. Core resources are purchased or created by Georgia Tech. These would be items in the catalog; in DSpace, the institutional repository; databases; journals; or other sources across campus that serve the university's mission. Community resources are those added by users actively enrolled or employed by Georgia Tech. This way a user's search could include materials that their peers are vetting and commenting upon. Lastly, documents in the world realm are anything created by users *outside* the Georgia Institute of Technology. Searching these resources is optional, since they may or may not have any relevance to Tech users.

It was determined early that for maximum effectiveness, *GaTher* (as the social bookmarking/citation management service was called internally) would need to be open to people outside of Georgia Tech. Researchers frequently work with

colleagues from other institutions, making this a necessity. This philosophy of openness also prevailed in the institution's Sakai implementation, with each site receiving 50 guest logins for membership outside campus.

After a bit of development, some limitations in Daisy became apparent. Daisy had never been intended to scale to the numbers of documents we were proposing (which would be in the hundreds of thousands or much more, depending on popularity).

The first flaw we came across concerned the retrieval of large numbers of documents. Daisy has a very SQL-like query language that allows for retrieval of documents by structured queries. However, because of the access control lists, which must be determined post-query, Daisy returns all matching documents before sorting them by relevance and applying any query limits. For queries that could produce an extremely large number of documents (such as “select the ID for every CatalogRecord page document”, which would be required to delete them), the load on Jetty, MySQL and Cocoon overwhelms Daisy and the only way to keep all system resources from being consumed is to kill the Daisy repository server. An alternative to this method would be necessary since large maintenance tasks would be inevitable, especially during the development process.

It also was discovered that there was no capability to create or modify document schemas via the HTTP API. For these sorts of administrative tasks, the Daisy Wiki would need to be run in order to access the administrator interface. Fortunately, this would only be necessary in very specific cases.

Due to a shift of institutional priorities at Georgia Tech (largely due to my position being assigned to support the campus Sakai initiative and a massive reorganization within the library), development had to be stopped shortly after the capability to parse and insert records into Daisy and Solr was written. It is unclear whether or not this project will be completed.

GaTher: Bringing it all Together

The last component of the Communicat project, GaTher, was to be the public interface. This was the means that people would add items to their personal or group libraries and it was the way they would search within them. The concept is modeled strongly after social bookmarking sites such as del.icio.us or Connotea but with the capability of much richer metadata. It would also serve as a very crude citation manager, but merely for storage and organization of references. For formatted bibliographies, GaTher would export to applications more suited for that purpose, like EndNote, Zotero or BibTex.

The preferred path to add citations would be through the Ümlaut, which would have a link to the user's preferred bookmarking service. If it is a service other than GaTher, the user will have the choice of adding it to both places, including a syndication feed in GaTher to harvest their public links or bypass GaTher altogether. We felt it important that the functionality did not interfere with the users' normal workflows, so efforts would be made to allow them to retain the status quo.

Since it is likely that users will desire to add resources that would not fall in the typical OpenURL resolving chain, bookmarklets or browser extensions would need to be generated to jumpstart the process. It would also be desirable for GaTher to work with the LibX and Zotero extensions. As there was no desire to maintain a suite of screenscrapers to pull metadata from the HTML pages, piggybacking on the screenscrapers used by Cite-U-Like, Connotea, or Zotero would lessen the burden, as would COinS (Context Objects in Spans) and UnAPI support.

Users would be able to organize their bookmarks any way they wish. They may annotate them, share them and open them for comments. Since users are members of groups, they also can comment on other's entries. The bookmarks are associated with groups, it becomes possible to weight the Solr queries: resources saved by a group, or similar groups, would be ranked more highly than other materials. As such, the groups become a focused portal into the larger community collection.

The Future of Communicat

Sadly, with my departure from Georgia Tech, it is unlikely that the library will complete this project. There are not the resources to enable such a project. With the proposed launch of VuFind in the Spring 2008 Term, the library will be addressing its next generation catalog needs.

Now that I am currently employed by Talis, it is possible that the Communicat could become a Talis Platform application. Since the Platform is an RDF triple store, the data model would only need minor tweaking to apply. It was this sort of scenario, the Daisy component being replaced, that inspired development of Cortex, so architecturally, the design would not need to change. It would most likely have to become much simpler to be marketable from a vendor standpoint, however.

Observations

While several next generation catalog projects attempt to address some or part of the Communicat's goals, none seemed to be quite as ambitious as Georgia Tech's vision; perhaps pragmatically so. This was an extremely large project with each individual piece itself a noteworthy endeavor. Institutional support was imperative, but the goals of the initiative were difficult to comprehend. This, in turn, made it difficult to prioritize.

Daisy, despite its limitations, was a strong choice. It would have provided a flexible backend for a host of future services. Since it was primarily being used as a data store, however, possible alternatives could be a native RDF database or a document-centric database, such as CouchDB. Daisy has a very strong user community, though, and its natural capability to strongly define documents and create links between them is functionality that would need to be duplicated.

A project such as this is difficult to manage within a committee. When the concept is abstract and the end-result could change dramatically from week to week, an agile process is necessary. Many library cultures have a difficult time subscribing to that philosophy. It might be better for a core team of developers and implementers to design a proof-of-concept before a larger committee is formed.

The ideals of the Communicat, an organic organization of resources, collected by the library's users and expanded and enhanced by librarians from every department, and not just cataloging are quite revolutionary and require considerable commitment from an organization. The benefits, however, are enormous: a catalog that reflects the actual research performed by those in the constituent community with current and topical added value layered on top by professionals within the library. Instead of shoehorning these services into our existing data silos or export the data from our silos verbatim into a third party search and discovery interface, wholesale revision of how data acts and interacts is necessary. Georgia Tech was unable to see this vision through completely, but perhaps a few of its goals can see their way into other library's projects.

About the Author

Ross Singer is Interoperability and Open Standards Champion at Talis. He may be reached by email at rossfsinger@gmail.com.

Connecting the Real to the Representational: Historical Demographic Data in the Town of Pullman, 1880-1940

by Andrew H. Bullen

The Pullman House History Project

The Pullman House History Project, a part of the Pullman State Historic Site's virtual museum and web site (<http://www.pullman-museum.org/>) and henceforth referred to as the PHHP, links together census, city directory, and telephone directory information to describe the people who lived in the town of Pullman, Illinois between 1881 and 1940. This demographic data is linked through a database/XML record system to online maps and Perl programs that allow the data to be represented in various useful combinations.

Demographic data collection is particularly important for research into the Pullman factory and community because both were a major destination for immigrants throughout the 110 years of the company's existence, allowing scholars to study 19th and 20th century migration and ethnographic trends. It has also revealed hidden facets of the Pullman story, allowing us to discover, for instance, the existence of a local militia extant between 1882 and 1890.



A Pullman Family, mid 1890s

The History of the Town of Pullman

In the mid-1870s, George Mortimer Pullman decided to expand his very successful rail passenger car service, as he had outgrown his Detroit shops and needed to dramatically expand his production facilities if he wanted to capitalize on the explosive demand for railroad car sales and service. He chose Chicago as a location for his new factory complex. Pullman had a fondness for Chicago as he had made an earlier fortune there in 1858. He also respected the wide open, entrepreneurial, winner-take-all aspect of the city. Pullman began acquiring land on the far south side of Chicago in 1880, setting up in secret the Pullman Land Association which was based in Massachusetts to avoid land speculation. Pullman

purchased 3,600 acres of prime industrial property, located between the Rock Island railroad and Illinois Central railroad lines and near the natural harbor of Lake Calumet. He commissioned a young architect named Solon Beman, 26, to design his factory complex and surrounding town.

The Pullman area was much more than a rail car manufacturing facility. George Pullman wanted to create what he envisioned as a workers' paradise, charging Beman to design and build what was eventually to become 32 blocks of row houses laid out in neat neighborhoods directly north and south of the factory complex. Almost unique in their day, each row house was equipped with running water, an indoor toilet, a spacious back yard, and lots of natural light and ventilation. The Pullman Company provided power, fixed the houses as needed, picked up garbage, and maintained the carefully designed landscaping. Pullman also provided shop spaces in a Market Hall and an indoor shopping Arcade; private entrepreneurs could rent from the Company and set up grocery stores, department stores, etc. in these spaces. George Pullman accepted numerous accolades for his "most perfect town" and made it a Chicago show piece during the 1892-1893 Columbian Exposition.



Houses on Cottage Grove under construction, 1881

As a response to cancelled orders and business downturns during the 1894 economic panic, Pullman laid off many workers and slashed the wages of his remaining employees. He did not, however, similarly slash rents, waive fees, or in any way attempt to mitigate the dire economic circumstances faced by his employees. The effect of this disastrous decision was the greatest of the 19th century labor struggles, the Pullman strike of 1894. The strike, spread nationwide by Eugene Debs and his attempt to organize all railroad workers into one all-encompassing union, lasted throughout the summer. It was eventually broken with help from federal troops dispatched by President Grover Cleveland which were sent in to ensure that the mail trains would be kept running. Pullman viewed the strike as a personal affront and never forgave the striking workers. He died shortly thereafter in 1897.

As a result of a 1907 U.S. Supreme Court decision, the company was forced to sell the town to private owners, and the Great Experiment ended. The city of Chicago annexed the town of Pullman, renumbering the properties and renaming the streets. Today, Pullman still has 98 percent of its original housing stock. It is a thriving Chicago neighborhood, home to an economically, socially, and racially diverse mix of people who deeply love the community. It has been my honor and privilege to attempt to represent the rich history of this fascinating community.

Challenges in Representing Pullman History

As of this writing, we have identified three main sources of primary historical material that describe the town of Pullman: contemporary newspaper articles, images, and census and other demographic data. Melding together these disparate data sources into a useful research tool has proven to be challenging. My first step was to cast about for a common thread that could tie these sources together and discovered that they could all be categorized in one or more of three different ways: as people, places, or events. This core assumption provided the main design philosophy for the whole project, and guides the path of future developments.

I prefer to program in Perl (for no other reason than that is what I am most familiar with) and I am very comfortable working with databases. My approach to developing an online Pullman history research tool was, therefore, to use a MySQL database to hold the data, using the paradigm of “people/places/events” as the core design guideline, and Perl to access the data tables and dynamically build web pages from the results set. Information that is unique to a particular data set is housed in external (to the database) XML files, referred to by MySQL records and displayed dynamically using Perl programs.

Three classes or groups of tables for the Pullman database have emerged: images, demographic data, and newspaper clippings. This article will describe in detail the three demographic data tables, which together form the core of the Pullman House History Project, and describe how they relate to the image and (coming in the future) newspaper clippings tables.

Part I: Representing the People of the Town

In order to best represent Pullman demographic data in the *PHHP*, I created a database-based/XML record system containing the data from the 1883 and 1889 city directories, a 1916 telephone directory, and the small portion of the 1900 U.S. census that we have entered. Eventually, the 1900 census will be completed, as well as adding in data from the 1910, 1920, 1930 and 1940 censuses. The 1890 census was, of course, lost in a fire in New York City; the data in the 1883 and 1889 city directories has proven to be an adequate substitute, allowing us to at least see who lived in which house and what they did for a living.

A database/XML record system is an ideal way to maintain disparate and unique data sources and yet still be able to manipulate them in a useful manner. For instance, demographic data in the *PHHP* database consists of records from censuses, city directories and telephone directories, each of which has unique data representation requirements. In addition, each decennial census asked questions unique to that census. Gathering together such a seemingly unruly and disparate herd of data at first glance seemed to me to be an impossible chore. Upon reflection, I realized that all of these data sources share a few common elements such as names and addresses. A simple MySQL table, shown below, became the structure that ties together the common elements. XML files, pointed to from within the individual records, can be used to contain the unique data elements of the different data sources.

The main table itself is laid out in the following manner:

```
mysql> describe census;
```

Field	Type	Null	Key	Default	Extra
address	varchar(100)	YES		NULL	
aptNumber	varchar(50)	YES		NULL	
street	varchar(50)	YES		NULL	
lastName	varchar(100)	YES		NULL	
firstName	varchar(50)	YES		NULL	
occupation	varchar(255)	YES		NULL	
ownOrBoard	char(1)	YES		NULL	
source	varchar(50)	YES		NULL	
uid	int(10) unsigned		PRI	NULL	auto_increment

This MySQL database also consists of several image metadata management tables. Like the *census* table shown above they contain brief records that point to more extensive and descriptive corresponding XML files describing scanned

images of Pullman. Our scanned images and their metadata may be viewed at <http://www.pullman-museum.org/cgi-bin/pvm/objectsViewRecords.pl?letter=A>. Description of these tables is, alas, a subject for another article.

Most of the fields are self-explanatory. The **ownOrBoard** field lets me distinguish between renters and sub-letters. The **source** field describes the source of the data. **uid**, present as a unique key in all tables, is the **unique identifier** for the record, which may also be used to point to any corresponding XML record(s). When a record is retrieved, its **uid** is then used to return the corresponding XML record from a file held on the server as 1900census.**uid**.xml.

As I mentioned, the data set from the 1900 census (and subsequent censuses) is much more complex, and must also be represented by a corresponding XML record. The DTD for the census information is:

```

<!ELEMENT houseDocumentation (houseDescription, physicalDescription)>
<!ELEMENT census1900 (houseInformation,
personalInformation,
ancestry,
immigrationStatus,
educationAndOccupation,
housingStatus,
censusInformation)>

<!ELEMENT houseInformation EMPTY>
<!ATTLIST houseInformation
recordID      CDATA      #REQUIRED
address       CDATA      #REQUIRED
aptNumber     CDATA      #IMPLIED
street        CDATA      #REQUIRED
dwellingNumber CDATA      #REQUIRED>

<!ELEMENT personalInformation EMPTY>
<!ATTLIST personalInformation
familiesNumber CDATA      #IMPLIED
lastName       CDATA      #REQUIRED
firstName      CDATA      #REQUIRED
relation       CDATA      #IMPLIED
race           CDATA      #IMPLIED
sex            CDATA      #IMPLIED
dobMonth       CDATA      #IMPLIED
dobYear        CDATA      #IMPLIED
age            CDATA      #IMPLIED
maritalStatus  CDATA      #IMPLIED
yearsMarried   CDATA      #IMPLIED
motherOf       CDATA      #IMPLIED
childrenLiving CDATA      #IMPLIED>

<!ELEMENT ancestry EMPTY>
<!ATTLIST ancestry
placeOfBirth   CDATA      #IMPLIED
placeOfBirthFather CDATA      #IMPLIED
placeOfBirthMother CDATA      #IMPLIED>

<!ELEMENT immigrationStatus EMPTY>
<!ATTLIST immigrationStatus
immigrated      CDATA      #IMPLIED
yearsInUS       CDATA      #IMPLIED
naturalized     CDATA      #IMPLIED>

<!ELEMENT educationAndOccupation EMPTY>
<!ATTLIST educationAndOccupation
occupation      CDATA      #REQUIRED
monthsUnemployed CDATA      #IMPLIED
monthsAtSchool  CDATA      #IMPLIED
read            CDATA      #IMPLIED
write           CDATA      #IMPLIED
english         CDATA      #IMPLIED>

<!ELEMENT housingStatus EMPTY>
<!ATTLIST housingStatus
ownOrRent       CDATA      #IMPLIED
ownOrMortgage   CDATA      #IMPLIED>

<!ELEMENT censusInformation EMPTY>
<!ATTLIST censusInformation
pageNumber      CDATA      #REQUIRED
lineNumber     CDATA      #REQUIRED
reelNumber      CDATA      #IMPLIED
notes          CDATA      #IMPLIED>

```

The XML record itself:

```

<?xml version="1.0" ?>
<Census1900>
<houseInformation
recordID="1405"
address="100"
aptNumber = ""
street="Stephenson"
dwellingNumber="69" />
<personalInformation
familiesNumber = "124"
lastName="Jeffery"
firstName="William"
relation="Head"
race="White"
sex="Male"
dobMonth="Jan"
dobYear="1841"
age="59"
maritalStatus="Married"
yearsMarried="36"
motherOf=""
childrenLiving="" />
<ancestry
placeOfBirth="England"
placeOfBirthFather="England"
placeOfBirthMother="England" />
<immigrationStatus
immigrated="1871"
yearsInUS="29"
naturalized="Naturalized" />
<educationAndOccupation
occupation="Police officer"
monthsUnemployed="0"
monthsAtSchool=""
read="Yes"
write="Yes"
english="Yes" />
<housingStatus
ownOrRent="R"
ownOrMortgage="Unknown" />
<censusInformation
pageNumber="111B7"
lineNumber="72"
reelNumber="290"
notes="" />
</Census1900>

```

This approach—using tables to hold brief records of disparate data sources that share a common thread and “offloading” the unique part—has the added advantage of being able to collect any data from any source, as long as it can be tied back to a Pullman address and combine it with other disparate but related data. As long as a record contains the bare minimum of a numerical street number, street name, and data source type, the basic data can reside in the census table. The **uid** field can then point to an external XML record containing the additional information. For instance, we will eventually need to index newspaper articles that describe events that occurred in specific houses. By creating a *census* table record that contains the street address in question and by adding the value of **article** to the **source** field, I can successfully index the article (at least as it relates to one or more specific addresses) and can retrieve a more detailed external XML file when I retrieve any record which matches the specified address.

The data collected also varies among the different censuses and can also be dealt with in this manner. Each Federal census is slightly different; while all of them record respondents’ names, addresses, etc., each decennial census asks questions unique to it. Representing this data in a database can be a challenge. One could, for instance, create a separate table for each census year (“1900#,” “1910#,” etc.) and record the data from each decennial census into the appropriate table. Searching across the various tables would involve a fairly complex SQL *join* operation, resulting in a lot of unnecessary

wear and tear on the SQL server. I have chosen, instead, to place the common information that all the censuses share in one table (creatively called **census**), offloading the unique information in the individual censuses into external XML files.

As I have described, this approach can accommodate a wide variety of disparate data. In addition, it allows the collection to be exposed to OAI harvesters. While I have not yet implemented the software necessary to expose the image and demographic XML records, it is being developed. Such a step will allow other institutions to make use of our data and—frankly—help us in securing grants, since we can quickly point to this facility as an indication of commitment to resource sharing.

Because the town was annexed and renumbered in 1907, I have had to create an “authority” table that translates between the old and new addresses, which leads us to the next demographic data table. The table is laid out as:

```
mysql> describe addressMaster;
```

Field	Type	Null	Key	Default	Extra
oldStreetNumber	varchar(50)	YES		NULL	
newStreetNumber	varchar(50)	YES		NULL	
oldAddress	varchar(50)	YES		NULL	
newAddress	varchar(50)	YES		NULL	
uid	int(10) unsigned		PRI	NULL	auto_increment

Here is an example record, which indicates that the modern address of 100 Stephenson is 11114 S. Champlain Ave.:

```
mysql> select * from addressMaster where oldStreetNumber="100" and oldAddress="Stephenson";
```

oldStreetNumber	newStreetNumber	oldAddress	newAddress	uid
100	11114	Stephenson	Champlain	3046

This particular record tells me that the modern address of 11114 S. Champlain Ave. was known before 1907 as 100 Stephenson Ave.

The Perl programs that make up the *PHHP* use this table to perform, essentially, an authority check on post-1907 addresses. I have decided to declare the pre-1907 address the “official” one simply because I had to pick one of the two to ensure data precision in my `SELECT` statements. When a Perl program needs to find information in the **census** table, it first sends a `SELECT` command to the **addressMaster** table requesting the old street number and street name of the address to ensure that all of the records pre- and post- 1907 are retrieved.

The last demographic information table in the database contains a housing survey, created in 2000 by Art Institute of Chicago students working in the Historic Preservation department. The students, as part of a massive year long project, collected data about each private structure in the neighborhood. We were then presented a copy of the resulting Excel spreadsheet data and accompanying photographs, which I promptly massaged into a table form. Like the census information, most of the data is resident in external XML files:

```

<?xml version="1.0" ?>
<!DOCTYPE houseDocumentation SYSTEM "/XML/DTD/houseDocumentation.dtd">

<houseDocumentation>
<houseInformation
recordID="94"
imageFileName="11114champlain.jpg"
address="11114"
street="Champlain"
ownerName="Balderama / Stigler"
nonPullmanAddress="Both owner- occupied as of 02/04" />
<houseDescription
houseType="Workers Cottage"
houseTypeOther=""
buildingType="Rowhouse"
buildingTypeOther=""
dateOfConstruction="1880"
dateOfSurvey="2002-04-28" />
<physicalDescription
roofType="Hipped"
roofTypeOther=""
roofMaterial="Asphalt"
roofMaterialOther=""
porchType="Wooden"
porchTypeOther=""
railings="Wooden"
railingsOther=""
mortar="Brown"
mortarOther=""
trimMaterial="Wood"
trimMaterialOther=""
chimney="Corbelled"
chimneyOther=""
contDetails="Dormers"
contDetailsOther=""
windows="Wood"
windowsOther=""
windowLights=""
storms="Aluminum"
stormsOther=""
windowSills="Limestone"
gutters="Other"
guttersOther=""
mainDoor="Modern Wood"
mainDoorOther=""
screenDoor="Metal"
screenDoorOther=""
otherSalientFeatures="Replacement windows and doors, replacement porch, painted brick Lg. center dormer flanked
    by two smaller front gable dormers w/ decorative rafter tails Brick corbeling below cove and between 1st.
    & 2nd. stories."
brick="Original"
brickOther=""
basementWindows=""
basementWindowsOther="" />
</houseDocumentation>

```

For various internal rights and permissions issues, I chose to place the data in its own separate table. The table is fairly simple, like the **census** table:


```
mysql> describe houseDocumentation;
```




Field	Type	Null	Key	Default	Extra
fileName	varchar(100)	YES		NULL	
address	int(10) unsigned	YES		NULL	
street	varchar(50)	YES		NULL	
mapReference	varchar(100)	YES		NULL	
vistaRecordID	int(10) unsigned	YES		NULL	
uid	int(10) unsigned		PRI	NULL	auto_increment

The **fileName** field points to an external JPEG house photograph file. **address** and **street**, of course, refer to the (post-1907) address of the property, which, as described above, is pointed to by the *addressMaster* table. **uid**, like its brethren in the *census* table, is used to point to an external XML file as *houseDocumentation.uid.xml*

The **mapReference** field is not currently used, but could be developed as a link to the online maps (see below). **vistaRecordID** is also not being used, but is meant to point to a historical streetscape photograph if one exists. Should I develop this feature, I would create another table whose records pointed to image files of streetscapes and landscape vistas. I could then view a streetscape image and click on a link in that image to view in more detail the individual houses that make up that streetscape. I have not developed this feature at all except for putting in place the linkages that would make it possible; such a feature would have to be a future development.

Part II: Joining the Real and the Representational

How do all of these tables and XML records work together? My first course of action was to create Perl programs that would connect to the database through the excellent DBI::DBD library. I created a Perl script that presented the names, addresses, and professions of the people represented in the main table in alphabetical name order.

		
The Virtual Museum	PULLMAN HOUSE HISTORY PROJECT	VIEW RECORDS BY NAME - A

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ALL

Names

A. C. Caldwell: Drugstore	Market Hall	Phone Number: 18	(1916 Phone Book)
Aadam, Clark		Union Foundry: Machinist	(1883 City Directory)
Aborg, A.P.	411 Watt	Carpenter	(1883 City Directory)
Abrisc, John J.	307 113th	Phone Number: 3906	(1916 Phone Book)
Ackels, B. S.	113 Watt	Machine Hand	(1883 City Directory)
Ackerman	1 112th	Foreman	(1883 City Directory)
Acton, Edward	645 Stephenson	Laborer	(1889 City Directory)
Acton, John	Brickyard Cottages	Brickyards: Brickmaker	(1883 City Directory)
Acton, William A.	645 Stephenson	Machinist	(1889 City Directory)
Adams, Charles E.	8 112th	School	(1900 Census) More...
Adams, Elijah	8 112th	Painter	(1900 Census) More...
Adams, Elijah C.	381 Morse	Painter	(1889 City Directory)
Adams, Elizabeth	8 112th		(1900 Census) More...

Clicking on the **More...** link above for any of the 1900 census entries actually brings up that specific census record. The Perl code that actually does the work behind the scenes is in the [census.pl file](#) (comments designated by octothorpes [#]). A snippet is shown here:

```
# Pulls in the $str variable to fill in the limiting letter. This SELECT statement calls the records themselves
from the census table.

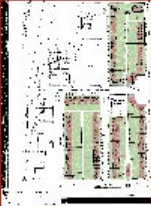


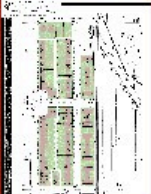
$query = "SELECT * FROM census $str ORDER BY lastName, firstName";
my $sth = $dbh->prepare($query);
$sth->execute();
while (@ADDRESS = $sth->fetchrow()) {
    $address = $ADDRESS[0]; $aptNumber = $ADDRESS[1];
    $dispstreet = $street = $ADDRESS[2]; $lastName = $ADDRESS[3];
    $firstName = $ADDRESS[4]; $occupation = $ADDRESS[5];
    $ownOrBoard = $ADDRESS[6]; $recordID = $ADDRESS[7];
    $source = $ADDRESS[8]; $uid = $ADDRESS[8];
}

if ($firstName ne "") { $nameOfPerson = $lastName . ", " . $firstName } else { $nameOfPerson = $lastName }
$count++;

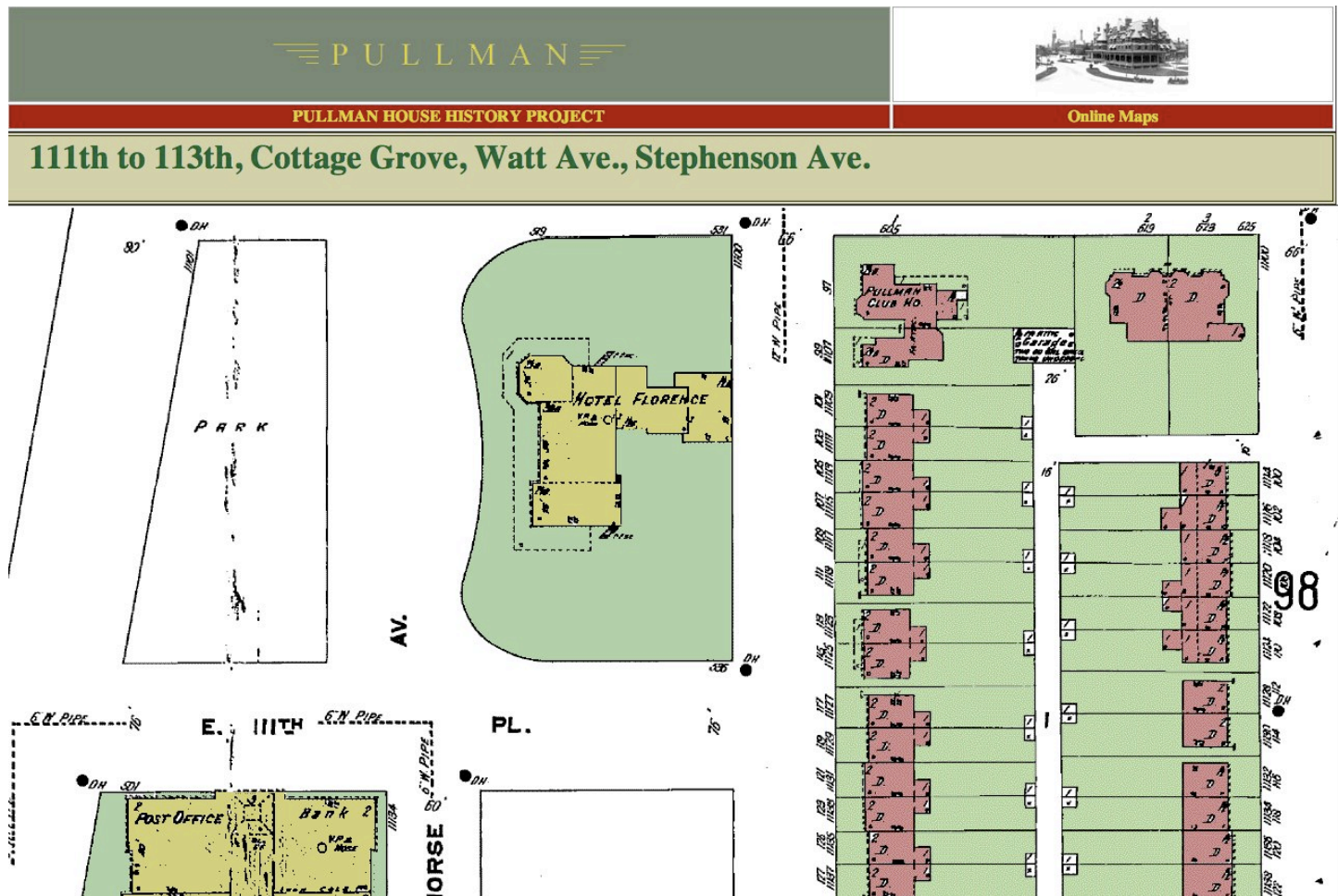
$dispstreet =~ s/th//g;

if ($ownOrBoard eq "R") { $ownOrBoard = "Primary Renter." }
elsif ($ownOrBoard eq "B") { $ownOrBoard = "Boarder." }
elsif ($ownOrBoard eq "U") { $ownOrBoard = "" }
if ($source eq "1900 Census") { $recordID = "<a href=\"\/cgi-bin\/pvm\/houseHistoryDisplay1900Census.pl?
fileName=1900Census.\" . $recordID . \".xml\">More...<\/a>" }
else { $recordID = "" }
```

1910 Sanborn maps, useful in allowing a more visual, map-based approach to discovering the demographic data in the *PHHP*, connect to the demographic tables as well. These maps were created as part of this project. They have been hand colored using appropriate Pullman colors, and, by using HTML image mapping (ISMAP), mapped to individual address records to create clickable online maps. There is also a fairly extensive set of Pullman maps online at <http://www.pullman-museum.org/phhp/maps/>. To me, maps help users relate demographic data to the actual site being described, so I have scanned in and enhanced a number of maps.

The Town of Pullman	
111th to 113th, Cottage Grove, Watt Ave., Stephenson Ave.	
111th to 113th, Cottage Grove to Watt Ave. -- 1886 Rascher Map	
111th to 113th, Stephenson Ave. to Fulton Ave. --- 1886 Rascher Map	
111th to 113th, Stephenson Ave., Fulton Ave.	

Clicking on one of the icons takes one to the actual linked map, such as the example below:



Notice that this 1910 map conveniently has both addresses on it for each property, reflecting the 1907 address changeover. Let us examine one property in detail, 100 Stephenson/1114 Champlain Avenue. Here is an 1882 photograph of this block of Stephenson, found as one of many images at the Pullman State Historic Site's site:



100 Stephenson is tinted blue and has a blue dot above it.

[I have, of course, modified a copy of the image especially for this article, which leads me to an intriguing side discussion. A possible future development for the PHHP would be to make images of the town interactive, connecting them to the demographic tables in the much the same way as the Sanborn maps are connected. As I say, this remains a future development.]

The house history record for this property is retrieved using [the Perl script houseHistoryGetAddress.pl](#).

		 <p>2000 Survey</p>
The Virtual Museum	PULLMAN HOUSE HISTORY PROJECT	100 Stephenson

Before 1907, this property was at 100 Stephenson. It is now 11114 Champlain.

1883 City Directory

Hellake, Joseph V.	<i>PPCC: Messenger</i>	
------------------------------------	------------------------	--

1889 City Directory

Asquith, George	<i>Painter</i>	
---------------------------------	----------------	--

1900 Census

Feshling ?, Christian	<i>Carpenter car</i>	More...
Jeffery, Emily		More...
Jeffery, William	<i>Police officer</i>	More...

1916 Phone Book

Seen here is the snippet of code that actually retrieves the records (comments again designated by octothorpes):

```

if ($range eq "old") {
$addressField = "oldAddressSort"; $streetField = "oldAddress"; }
else { $addressField = "newAddressSort"; $streetField = "newAddress"; }

# Allows the user to switch back and forth between old address (pre-1907) and new address (post-address)

$queryaddress = "SELECT * FROM `addressMaster` WHERE $addressField = '$address' AND $streetField = '$street'";
my $sth = $dbh->prepare($queryaddress);
$sth->execute();
($oldStreetNumber,$newStreetNumber,$oldAddress,$newAddress,$oldAddressSort,$newAddressSort,$uida) = $sth->fetchrow();

# The above code simply finds the old and new street numbers and names for the indicated property

if ($oldStreetNumber eq "") {
$addressText = "No records found for this property.";
$oldAddress = $street;
$oldStreetNumber = $address;
}
else { $addressText = "Before 1907, this property was at $oldStreetNumber $oldAddress. It is now
$newStreetNumber $newAddress." }

$imageFileName = $newAddressSort . $newAddress;
$imageFileName =~ tr/ //d;
$imageFileName = $imageFileName . ".jpg";
$imageFileName =~ tr/A-Z/a-z/;

# If we have a photo of the property from the 2000 AIC project, this will determine its name for subsequent
display

$query = "SELECT recordID FROM houseDocumentation WHERE address='$newStreetNumber' AND street='$newAddress'";
my $sth = $dbh->prepare($query);
$sth->execute();
$xmlFile = $sth->fetchrow();
$sth->finish();

$housingSurveyStr = "";
if ($xmlFile) { $housingSurveyStr = "<a class=\"centeredObject\" href=\""/cgi-bin/pvm/housingSurveyMenuViews2.pl?
address=$newStreetNumber&street=$newAddress\">2000 Survey</a> " }

# If we have a 2000 AIC survey for the property, then build a link to a CGI program that will display it

# I am skipping some code here, which simply builds up the headers to create a dynamic web page

# Now iterate through the 4 data sources for demographic data (1883 and 1889 city directories, 1900 census, and
1916 phone book




<h3 class="displayBanner">
1883 City Directory
</h3>
<p />
<table border="1">
EOF

# I will repeat this code as many times as necessary to retrieve all the names of the people who lived or worked
at this address.

$querya = "SELECT * FROM `census` WHERE street = '$oldAddress' AND address = '$oldStreetNumber' AND source =
'1883 City Directory' ORDER BY lastName, firstName";
my $sth = $dbh->prepare($querya);
$sth->execute();
while (@ADDRESS = $sth->fetchrow()) { &getValues }
print <<EOF;
</table>
<p />
<h3 class="displayBanner">
1889 City Directory
</h3>
<p />
<table border="1">
EOF
$querya = "SELECT * FROM `census` WHERE street = '$oldAddress' AND address = '$oldStreetNumber' AND source =
'1889 City Directory' ORDER BY lastName, firstName";
my $sth = $dbh->prepare($querya);
$sth->execute();

```

Officer William Jeffrey's 1900 census record from the example above is retrieved using the Perl script [houseHistoryDisplay1900Census.pl](#) (snippet displayed further down):

		
Pullman Virtual Museum	PULLMAN HOUSE HISTORY PROJECT	1900 Census Information for Jeffery, William
Name:	Jeffery, William	
Address:	100 Stephenson	
Relation to Head of Family:	Head	
Sex:	Male	
Race:	White	
Date of Birth:	Jan 1841	
Age:	59	
Marital Status:	Married	
No. of Years Married:	36	
Mother of How Many Children:		
No. of These Still Living:		
Place of Birth:	England	
Father's Place of Birth:	England	


```
# I am using the Perl module XML::Simple to reformat the XML file that contains the 1900 census data. The
# programs knows which file to retrieve because it was passed from the More... link above, and was determined by
# the getDefaults subroutine above.

use XML::Simple;

my $xmlref;
my $xmlref = XMLin("/home3/pullman/sites/pullman-museum.org/html/XML/$xmlFile");

# Is the actual file name and path

##### houseInformation #####
$aptNumber = $xmlref->{houseInformation}->{aptNumber};
$address = $xmlref->{houseInformation}->{address};
$street = $xmlref->{houseInformation}->{street};
$dwellingNumber = $xmlref->{houseInformation}->{dwellingNumber};

##### personalInformation #####
$familiesNumber = $xmlref->{personalInformation}->{familiesNumber};
$lastName = $xmlref->{personalInformation}->{lastName};
$firstName = $xmlref->{personalInformation}->{firstName};
$relation = $xmlref->{personalInformation}->{relation};
$race = $xmlref->{personalInformation}->{race};
$sex = $xmlref->{personalInformation}->{sex};
$dobMonth = $xmlref->{personalInformation}->{dobMonth};
$dobYear = $xmlref->{personalInformation}->{dobYear};
$dob = $dobMonth . " " . $dobYear;
$age = $xmlref->{personalInformation}->{age};
$maritalStatus = $xmlref->{personalInformation}->{maritalStatus};
$yearsMarried = $xmlref->{personalInformation}->{yearsMarried};
$motherOf = $xmlref->{personalInformation}->{motherOf};
$childrenLiving = $xmlref->{personalInformation}->{childrenLiving};
```

Challenges Ahead

In the future, we hope to make better use of sources that give the census entries life and breadth. The records representing people need to be expanded to have any meaning and context. Consider the case of Malcolm McQueen and his widow:

McNamara, John	Bldg. C Fulton	Sanitary Inspector	(1883 City Directory)
McNamara, John	140 Fulton	Justice of the Peace	(1889 City Directory)
McNary, John	200 Fulton	Dredgeman	(1883 City Directory)
McNorden, John	434 Watt	Striper	(1889 City Directory)
McNorden, Ranson	434 Watt	Laborer	(1889 City Directory)
McNurney, Miss Maggie	Morse	Pastry Cook: Hotel Florence:	(1883 City Directory)
McNutt, Samuel	132 Fulton	Upholster	(1883 City Directory)
McOuley, John	7 112th	Painter	(1900 Census) More...
McPeck, James F.	221 Watt	Inspector	(1889 City Directory)
McQueen, Malcolm	106 Stephenson		(1883 City Directory)
McQueen, Mrs. Sarah	224 Watt	Unknown	(1889 City Directory)
McRoan, Owen	569 Stephenson	Steam Forge Worker	(1889 City Directory)
McWilliams, Wesley	520 Fulton	Cabinetmaker	(1889 City Directory)
Meacham, A. D.	4 111th	PPCC: Clerk	(1883 City Directory)
Meack ?, ?	810 Ericson		(1900 Census) More...
Meack ?, Charles	810 Ericson		(1900 Census) More...
Meack ?, Sohisa ?	810 Ericson		(1900 Census) More...
Meack ?, William	810 Ericson		(1900 Census) More...

I have found two references to McQueen in the Chicago Tribune.

Malcolm McQueen, foreman of the repair shops at Pullman, went into the cellar of his residence at No. 124 Watt Avenue about 4 o'clock yesterday morning and hanged himself. In his pockets were found a note saying that his head pained him and a message to his wife bidding her goodbye. *Chicago Tribune*, Sept. 23, 1888

and

Robert McQueen is a guest at the home of his sister in law, Mrs. Malcolm McQueen on Watt avenue... The funeral services of the late Malcolm McQueen were held at the Presbyterian Church Monday afternoon under the charge of the Masons. Before 2:30, the hour set for the funeral, the church was filled to its utmost capacity and it was with difficulty that the ushers kept the crowd from filling the aisles. *Chicago Tribune*, Sept. 30, 1888

Mr. McQueen's story is poignant and deserves to be highlighted. These accounts should be connected to McQueen's demographic information, tying his name to the text of the article. The primary source for newspaper articles like these is the Pullman Company Scrapbooks. The Scrapbooks, held at the Newberry Library in Chicago, were kept by the Pullman Company from 1870 to 1925. They represent a priceless treasure of clippings concerning the Pullman town and company. There are over 100,000 individual clippings in the Scrapbooks, of which a few example pages can be seen at <http://www.pullman-museum.org/scans/>. We are currently working on a grant that will allow us to digitize these articles as OCR'd PDF files which can then be keyword indexed. We use SWISH-E as our search engine and harvester, and we have been very happy with it.

I can easily connect the names contained in the census table and these newspaper clippings by writing a brute force program that searches for each name as selected in a SWISH-E-based clippings index. This is, however, a crude and imprecise solution. A better solution would be to employ a natural language processing program that will create a table more precisely linking the names in the census table and the records in the keyword index. The ideal solution, of course, would be to create item level metadata records for each article, a very expensive and time-consuming proposition. How we are going to properly handle the clippings records is a matter undergoing discussion.

Another major hurdle is simply completing the census data inputting. I have, to this point, relied on volunteer efforts to completely input the 1883 and 1889 city directories and the 1916 phone directory. Volunteer efforts, however, have only been able to enter 5% of the 1900 census. Clearly, volunteers will not be able to get the rest of the census data entered, so we are currently working on yet another grant that will allow us to engage a commercial firm to enter the rest of the 31,750 census entries.

In Conclusion

The Pullman State Historic Site's virtual museum records an average of 212,854 hits a month. The use of the Pullman Virtual Museum increases on a monthly basis, and we look forward to expanding it to ever greater functionality by adding in the newspaper clippings and finally digitizing the remainder of the census records. It has been, so far, a rare opportunity to work on such an enjoyable and unique project. Please come visit us at <http://www.pullman-museum.org/> and explore the world of Pullman.

Author Information:

Andrew Bullen is the Information Technology Coordinator for the Illinois State Library, and a proud resident of the Pullman community. He can be reached at abullen@ameritech.net.

BOOK REVIEW: The Success of Open Source by Steven Weber

Weber, S. (2004). *The Success of Open Source*. Harvard University Press. ISBN: 0674012925 ([COinS](#))

By Eric Lease Morgan

Introduction

The Success of Open Source by Steven Weber details the history, process, motivations, and possible long-term effects of open source software (OSS). This scholarly yet easy-to-read, well-written, and provocative book is worth the time of anyone who wants to understand how open source software is affecting information technology. Using Linux as its primary example, the book describes how the process of open source software may affect business & economics, methods of governance, and concepts of intellectual property. It is also a great read for those of us librarians who desire to play a role in the building of “next generation” library catalogs and other library-related information systems. My acquisition of the book was rather embarrassing, and at the same time typical. As the leader of an open source software project called MyLibrary, I asked some of my fellow hackers and open source software aficionados for advice on how to promote MyLibrary and build a larger community around it. One of the suggestions was to read Weber’s book. Like most people, I searched Google for the title, and Google returned it as the first hit. I was able to read the entire text online, if I desired. I didn’t. The librarian in me then went to WorldCat where I learned the book was located down the street in another academic library. I could borrow the book for free. All I had to do was visit the library and check it out. I didn’t. Instead, I looked the book up on Amazon.com and found a “new” copy from an Amazon.com Associate. Twelve dollars and four days later my book arrived. Easy. Convenient. Cheap. Food for thought. The book can be divided into four overarching topics:

1. the history of open source,
2. the process of open source software development
3. business models and open source software’s relationship to the idea of a “commons”, and finally,
4. a summary as well as a look to the future describing how the process of open source software might affect other human endeavors.

History of OSS

Weber traces the history of open source software from its roots in AT&T UNIX and the Berkeley Software Distribution (BSD) to the present day Linux operating systems. Weaved throughout is the development of networking technologies, specifically the Internet. The history brings to light two very influential computing philosophies: the “Unix Way” and software as something to be bought and sold. The first computing philosophy is the “Unix Way”, an outline of three engineering principles for making good software:

1. write programs that do one thing and do it well,
2. write programs that work well together, and
3. write programs that handle text streams because that is a universal interface.

Software that adheres to these principles is typically considered more useful than software trying to be all things to all people. Such software is modular, portable, easy to create and maintain, and can be applied in any number of settings. The second philosophy revolves around ideas of intellectual property. Suppose I spend time and creative energy writing a piece of software. Through this expenditure I have the right to sell the software to other people in order to gain compensation for my efforts. Software, like other goods, can be exchanged for other things of value, namely money. Moreover, if people copy my software and give it to other others, then such a process is just like stealing from me since I am not being compensated for my efforts. This was the attitude of Bill Gates as he stated it in his “open letter to hobbyists” as they distributed his implementation of the BASIC programming language in order to run programs they had written

against it. This perspective regarding software as something to be bought and sold ultimately led to the creation of Microsoft. At the time of Gates' writing of the 1976 "letter" computers were bought and sold. Software sort of just came along for the ride. While it is important to note that I, the author of this review, am certainly an advocate for open source software, the book takes no sides one way or another. Instead, throughout the book, Steven Weber tows a middle ground by simply asking questions and then does his best to answer them as objectively as possible. The book neither advocates nor condemns open source software. It simply observes the environment and makes generalizations accordingly.

OSS as a process not a thing

The chapters I found most interesting described the open source software process, how it works, and the motivations of its participants. I learned from the book that open source software is more about a particular process and less about a thing or product. Consider two statements a priori:

1. software can be copied an infinite number of times and not denigrate the original version, and
2. globally networked computers allow the tiniest numbers of like-minded individuals to find each other easily.

Given such an environment the open source software process flourishes, and that process is outlined here:

1. Someone has a computing problem they want to solve — an itch to scratch.
2. The person builds on the good work of others and writes a computer program.
3. The person "freely" shares their software with others under some sort of license agreement.
4. A community forms along with norms of behavior and guidelines (governance) for contributing back to the solution.
5. The software grows and matures, hopefully
6. Go to Step #1 until the software is "done" or until someone else wants to take on the leadership role.

The resulting open source software application, the book points out, is not necessarily better (or worse) than "closed source" software. Instead, it is simply different. It is more easily modified. It is vetted through the eyes of end-users — a set of self-described pragmatists wanting to scratch their own itch. Furthermore, since the process easily accommodates the philosophy of letting a thousand flowers bloom, the resulting software is not necessarily designed for the mass market: The software is not aimed at a lowest common denominator. What motivates open source software participants? Weber shies away from the altruistic motives espoused in Eric Raymond's *The Cathedral and the Bazaar*. Instead, Weber imagines that the motivations stem from a desire for artistry & craftsmanship, the desire to create better software, and the desire for recognition from peers ("ego-boosting"). Many of the people who participate in open source software seem to enjoy puzzle solving. Some of them like reading and writing beautiful code — software as poetry. They are engineers looking to build better solutions to known problems, to "build a better mouse trap". Like scholars and researchers in academia, peer-review is an important aspect of the work, and if you write something that is used (cited) by many people, then in the eyes of others your value is increased.

OSS and business

The sections and chapters regarding open source software and business are probably the most significant aspects of the book. They cover issues that are the least understood by the wider community including the definition of "free", the concept of property and the "commons", the differences between the BSD and GPL licenses, how these differences effect business opportunities, and business models in an environment where the primary thing to be exchanged for value is bound by rights of distribution as opposed to exclusion. In our increasingly commercial society, the concept of "free" software is something most people find confusing. Weber compares and contrasts "free" from the point of view of the Free Software Foundation (led by Richard Stallman) and the point of view of dot-com companies such as Red Hat (a commercial distributor of open source software). In both cases the concept of "free" should be equated with the word "liberty" as opposed to "gratis". However, the FSF approach has a moral slant while the Red Hat approach emphasizes

practicality and the ability to easily improve software solutions. The concept of property plays a big role in business models. How can you sell something that is “free”? How can you earn money on a thing that is a part of the community commons? Who owns this intellectual property? These questions can be answered, according to Weber, by interpreting the BSD and GNU (or General) Public License. In both cases the software is given away gratis. The differences lie in redistribution. Under the GPL license, new software based on GPL-licensed software must be re-distributed under the GPL or a less restrictive license. The BSD license does not have this stipulation; new software based on BSD licensed software does not have to be redistributed for “free”. Mac OSX is an example of BSD-based software which has been commercialized: Apple, Inc. started with BSD Unix, enhanced it, then redistributed their modified version for a fee. Weber outlines a number of possible business models for open source software:

1. support sellers,
2. loss leaders,
3. “sell it, free it”,
4. accessorizing,
5. service enablers, and
6. branding.

For each of these models Weber points to a number of examples.

The book’s summary

The last two pages of the book summarize much of Weber’s observations, and listing them here feels a bit like spoiling the ending of a mystery novel. The effective open source software process/project needs to take into account and support:

- disaggregated contributions,
- the need for a critical mass of users,
- peer review,
- the positive effects of the Internet,
- belief that a small group can generate something truly useful, and
- a voluntary community.

Weber also outlines the characteristics of effective agents (open source software participants). They:

- can judge the viability of the product,
- can make an informed bet their contributions will be used,
- are driven by things beyond simple economic gain,
- gain personal knowledge through the process, and
- hold a positive ethical valence towards the process.

OSS and libraries

Naturally, I read the book through my rose-colored glasses of librarianship. After reading it and combining it with more recent personal experiences, I am now less of a “believer” in open source software. I take away a more realistic perspective on the definition of open source, its process, and what motivates its participants. This does not in any way diminish my belief that the open source software process can benefit the library community and therefore library users. Throughout the book I kept comparing the kernel of the Linux operating system to the integrated library system (ILS) of libraries. Both provide fundamental interface functions between two entities. In the case of an operating system kernel, the interface is between hardware and people. In the case of an ILS the interface is between a library and patrons. Many library open source projects already exist. The extent they meet with continued success and wider adoption, I predict, will be measured by the extent that they can accomplish the following things. First, an easy-to-understand vision

statement needs to be outlined by one or more people who possess leadership qualities. Second, those leaders need to amass the resources required to make their vision a reality. Third, they need to put their vision into practice allowing as many people to participate as possible. Fourth, start small and work up. Encourage the building and re-use of existing core applications, such as databases, indexers, editors, and server platforms. Make sure the applications are modular and standards-compliant. Practice the Unix Way. Make it work first, then improve things. Don't even attempt to create the perfect system the first time. The process won't be quick. The process won't be easy. The process won't be "free". On the other hand, the process will empower and enable the profession. It will give it increased choice and opportunity. Weber's book, *the Success of Open Source*, can be used as a set of guidelines - a description of a framework - for building software solutions for the computing problems facing libraries.

About the Author

Eric Lease Morgan <emorgan@nd.edu> is the Head of the Digital Access and Information Architecture Department at the University Libraries of Notre Dame. He considers himself to be a librarian first and a computer user second. He and his fellow teammates have primary responsibilities for the University library's website, the campus-wide search engine, and a number of digital library projects. He is also on the editorial committee of the Code4Lib Journal. In his spare time he can be seen folding defective floppy disks into intricate origami flora and fauna.

COLUMN: 700 Dollars and a Dream : Take a Chance on Koha, There's Very Little to Lose

by BWS Johnson

One of the sentiments that I try to infect the Library Science populace with is the notion that innovation can come from anywhere at all. This is especially true of rural libraries. I truly believe that the meekest amongst us has a special duty and a special circumstance that fosters innovation. Ours is not the culture of red tape entrenched tradition, but rather the atmosphere of the pioneer. No one will notice a failed experiment in the middle of nowhere, but they'll certainly notice a cataloguer someplace in Edema making a dent in backwards standards. So much of this field is not about the money—technology is definitely in that basket, particularly with Open Source making a fierce showing of things.

This was the argument I took to my Board: let me take a small portion of our State Aid to Public Libraries money and try out this new fangled thing. The software's free. Yes really, we don't pay anything for it, I just go out and grab it. Nope, it won't be more than \$1,000 so we'll still have emergency money. If it doesn't work out, I can just reformat the drive, make it a regular old public access terminal, and we're good to go. Nope, it's not stealing. Everyone's got something to give back to the community, and when I set things up and figure out what does what, I'll write about it, and that'll be our share. It's like a barn raising.

We had nothing to lose—our Library wasn't automated yet. Ours was the perfect test environment. My Board was receptive, my Patrons weren't attached to any system whatsoever, and my Staff were behind the move. My loving husband was willing to install this for us, as well as physically assemble a custom server, although that's not necessary—Koha will run on just about everything. As with any other product, the more you put into your hardware, the better the result until you top out at a certain point. On Koha, that point is frighteningly low, making my \$700 box a Ferrari Testarossa. With the proliferation of Linux user groups out there, it oughtn't be too hard for just about anyone to approach the geeks that be and walk away with a functional server in a matter of a few hours.

We're mostly done with bibliographic input now—we've got just over 7,000 items catalogued of about 8,500. Our Patrons are in the database. We could circulate now if we wanted to. We've tested the basic features that we need and they work well enough for our purposes. When we started out, we were looking for the barest minimum of functionality. We got a whole lot more than we bargained for.

Koha is far more reliable than many commercial ILS options. This was certainly a factor with me. It seemed as though things would be down every other month for a few days of unscheduled time with a few of the commercial products I've had the displeasure of experiencing. Our server has been down twice in about 3 years of testing, with the box running 24/7. Once was when my roommate inadvertently unplugged the server to charge his mobile phone. The second time was a catastrophic hardware failure. The power supply essentially caught fire. I was terribly worried my data was toast. It wasn't. I had backups, but I didn't need to use them.

Koha is far better at keyword searching than anything I've ever seen. Something in the way it ranks search results really ends up giving you highly relevant items first. It also loots and pillages its way through a MARC record so that those notes fields everyone tires over are searched through, too.

The support is astounding. I have yet to pay money for support, yet I've had developers bend over backwards to program in a feature I've wanted, in a remarkably short span of time. A basic reports module came to me free of charge inside of a couple of days from across the ocean in France. An IRC channel dedicated to Koha tends to have someone on it most of the time. With heavily involved developers in the United States, France and New Zealand the project doesn't sleep. I can't imagine the results you would see if you had a few thousand dollars to give to a developer for your feature. At a recent demo, I was eating lunch and chatting with the other Librarians about which developer was responsible for what feature. One of the other Librarians stopped me and said of their product, "Wow, this is so great. You know the **names** of the developers. We're lucky to even get through to support on the telephone!"

With Koha you get something you don't get with any other product. You have complete control over what your catalogue looks like, you don't have to wrestle with a vendor to get your data to do what you want it to do, and if the product doesn't have a feature you need, you can programme it or pay someone to add it.

The rate of development and improvement over the past few years has been nothing short of astounding. When I started using Koha, it was very wooden and very ugly. It's come a long way since then. The current out of the box release is on par with at least a handful of commercial products. When the templates are customised for a given Library, the product can meld seamlessly and aesthetically with a Library's website. The Horowhenua Library Trust catalogue can give you a taste of the aesthetics: (<http://www.library.org.nz/cgi-bin/koha/opac-main.pl>) The upcoming Version 3 looks quite like the Athens County catalogue: (<http://search.athenscounty.lib.oh.us/>)

Since Koha was developed in New Zealand, connectivity issues caused the developers to make a product that would be very easy to access regardless of the speed of a person's connection. I was able to access the catalogue which resided at home in Albany, New York from my Library in Western Massachusetts with no noticeable wait time for searching and data input over an incredibly crummy connection. (It was allegedly a 56k connection, but the plain old dial up telephone line connections routinely ran faster.)

It's not for everyone, however.

Installation is still difficult. Unless you've someone in the area who is very comfortable with Linux administration, this project will be a difficult set up. On the other hand, one can pay for a preinstalled box.

Cataloguing for a large institution would be tough. Holdings information is a bit bodged at the moment. The cataloguing module is certainly clunky to use. The interface is tabbed with each MARC field getting its own text box. As a result, either a Librarian ends up sticking all of the fields in one tab for a really long screen of many, many boxes, or fields are missed by sloppy cataloguers that don't switch tabs. It is possible to set up frameworks that anticipate necessary fields for a given material type, but this entails a good deal of planning during setup. The good news in this department is that thanks to Google summer of code, a powerful new tool is being worked on to make things much nicer for cataloguers everywhere, and functionality should be vastly improved with Version 3.

Reports are also getting a massive workover thanks to sponsorship from the British National Health Service. These can be tricky from a programmer's perspective thanks to each client wanting a different data set. The new module will guide a user through the process of selecting which sets they'd like in order to produce the table or chart they'd like to pull from the raw data.

Because Koha came from the mind of a computer programmer, there are creature comforts that Librarians take for granted that could be absent or less fleshed out than one might like. Increasingly, this is less true as the developers address new feature requests and the project gathers fans, and thus steam, along the way. The positive side of this is that it rapidly assimilates neat new Web 2 innovations, for instance tag clouds are going to be featured in the new OPAC.

Like everything out there, there are bugs. Developers do work to keep this down to a minimum, but I don't want anyone to think I promised perfection. Users are encouraged, and yea, even thanked when they submit problems to the project's bug tracker, bugzilla (<http://bugs.koha.org/cgi-bin/bugzilla/index.cgi>).

It's far from perfect, but I can't name a commercial product that has it all.

Ask yourself—what does my Library have to lose? Why not run an open source catalogue redundantly to your current system to discover the differences for yourself? If you do like Koha, imagine how much you do have to lose in terms of that nasty annual license fee. You can choose to either have the product supported at an affordable rate or you can just set everything up yourself and never pay a thing except for the cost of the hardware.

The model that Koha is based on is very similar to National Public Radio or the Corporation for Public Broadcasting. Open source is out there waiting to be enjoyed by everyone, regardless of financial status. Just as local programming is developed in your backyard and contributed back to the National efforts, individual Libraries can customise their installation. When some flavours are contributed back, like the Nelsonville templates, they prove to be very popular and are widely accepted in turn, like Fresh Air or Nova. Not everyone supports their local affiliate in a fund drive,

and not everyone can afford to financially support the Koha project. When Libraries choose to pay for support or new features, everyone benefits since good reliable features can be selected out and then rolled into the product. Even small contributions of time and labour end up making large differences in making the product better through collective effort.

There is further information and a demo on the Koha web site: <http://www.koha.org>

and Nicole Engard has a blog entry about Koha: <http://www.web2learning.net/archives/1165>

About the Author:

BWS Johnson is a graduate of the Graduate School of Library and Information Science at the University of Illinois Urbana - Champaign and was the Director of the Hinsdale Public Library in Hinsdale, MA. She was recently honoured to serve as President of the Western Massachusetts Regional Library System.

