



The Virtual Shelf

Devin Blong

Jonathan Breitbart

UC Berkeley School of Information
Master of Information Management and Systems
Final Report, 7 May 2009

Introduction

Abstract

The Virtual Shelf is the product of a two year partnership between the Open Library project (<http://openlibrary.org>) and the authors. In working with the Open Library, the authors' goal has been to help enable users of this massive digital library initiative to interact with and utilize the Open Library's collection in new and innovative ways. The Virtual Shelf idea initially grew out of interviews with and observations of physical librarians and library patrons, from which it became apparent that the bookshelf itself plays an integral role in successful information seeking and retrieval experiences. These initial observations inspired subsequent in-depth literature reviews concerning information seeking behavior (both online and offline) and scholarly communication and collaboration, as well as existing digital library catalogue systems. As a result of this research and analysis, the authors propose a visualization and collection building interface for the Open Library collection that combines many of the features and benefits of both physical and digital information systems. We call this tool the Virtual Shelf (<http://virtualshelf.org>).

The Open Library

Founded by Brewster Kahle, creator of the Internet Archive (<http://archive.org>), the Open Library project is a not-for-profit organization that seeks to make information about all published work (including high-quality full-color scans of works in the public domain) available to anyone online, free of charge. In the service of the Open Library's goal of universal access to information, the Open Content Alliance was founded to help bring together supporters and donors of materials and funding. This organization has partnerships with hundreds of libraries across the world, as well as major corporations and organizations such as Adobe Systems Incorporated, HP Labs, Microsoft, O'Reilly Media, Yahoo!, Xerox Corporation, and the William and Flora Hewlett Foundation¹.

The Open Library's mission is to create "One web page for every book ever published."² To this end, it takes the form of an enormous wiki³, in which each book (and author) in the collection has a unique webpage. From these book pages, users are able to obtain catalogue information and book metadata, as well as links for finding the books at local libraries or for purchase through online merchants. For public domain works that have been scanned and added to the collection, users can read the full text scans online, download copies of the books, and even print physical copies on demand.

Three parallel tracks are in place for adding content to the Open Library catalogue. First, the Open Library is constantly obtaining book information from various different sources, including hundreds of libraries from around the world, as well as the Library and Congress and Amazon. Books are also being

¹ See <http://www.opencontentalliance.org/contributors/> for a more complete list of Open Content Alliance contributors.

² <http://openlibrary.org/about>, retrieved 3 May, 2009.

³ A wiki is a software application that allows users to create, edit, and organize webpage content using any standard web browser. Wikis are unique in that they allow non-technical users to easily contribute information and content to a website or collaborative online community. For more information, see <http://wiki.org/wiki.cgi?WhatIsWiki>. See <http://openlibrary.org/about/tech> for information about the Open Library technology.

scanned into the system as funding and copyright permit. Finally, because the Open Library is a wiki, anyone can contribute to the book and author records of the collection. Users can add new books to the system and can freely edit and add to any book and author page on the site. In this sense, the Open Library is a truly unique endeavor in that it is a digital library that anyone can participate in and contribute to. It is a truly “public” library that promotes open information sharing. All the software is open source and all of the data in the collection is freely available to anyone at any time.⁴ In addition, the Open Library is planning a number of user content based functionalities, including book reviews and annotations, as well as the ability to save custom user-created book collections. At the time of writing, the Open Library provides records of over 22 million books, with over 1 million full-text scans.⁵

Project Objectives

Our partnership with the Open Library began in the fall of 2007. Our objective in this partnership has been to help provide ways for users to better explore, interact with, and utilize the Open Library collection. Throughout this endeavor, we have also worked closely with the Open Library’s staff to ensure that our application helps promote their primary goals of open information sharing and collaboration. As our primary stakeholder in this project, it has been especially important for us to ensure that our development supports and works in tandem with the development efforts of the Open Library team and to this end, we have been meeting weekly with members of the Open Library staff for the past year. As a result, we believe that the Virtual Shelf interface and architecture fit extremely well in this environmental context and can add significant value for a wide variety of Open Library users, including (but not limited to) basic information seekers and Open Library website visitors, other developers working with the Open Library collection, individuals and collaborative teams conducting academic research, as well as people hoping to engage in social dialogue and exchange around books and book collections.

Initial Research and Findings

As mentioned briefly above, the initial idea for the Virtual Shelf grew out of observations of librarians and library users. To better understand the nature of information seeking behaviors and practices in the library and online, we conducted contextual inquiries and interviews of librarians and library patrons. Our findings in these initial research endeavors directly influenced the design and development of the first Virtual Shelf prototype and its features. A summary of this research follows.

Contextual Inquiries and Interviews

As a starting point for the Virtual Shelf project, we conducted a series of contextual inquiries (in which we observed library patrons and library workers in their natural work context) and interviews. These interviews mostly took place at the University of California, Berkeley main library, with a few observations also taking place at the Berkeley, CA Public Library. In this research, with respect to library patrons, we focused on people who were using the online library catalog to search for relevant resources. The library workers we interviewed consisted of reference librarians, as well as a subject cataloguer and

⁴ See <http://openlibrary.org/about> to learn more about the Open Library project and mission.

⁵ <http://openlibrary.org/>, retrieved 3 May 2009.

the Integrated Systems Manager, who was in charge of the migration effort to the next generation digital catalog system for the library. From these inquiries, we hoped to gain some perspective on common search strategies of information seekers utilizing library catalogs and collections. The following sections outline our main findings from this initial research.

One of the overarching themes resulting from the interviews and contextual inquiries was that librarians find classification systems such as Library of Congress subject headings extremely powerful and nuanced, providing wonderful resources for users who can master it. Far more important, however, is the fact that very few library patrons actually utilize these systems. We were told by librarians that the only users who truly take advantage of this system are highly skilled researchers who began honing their search skills long before the digital age. Instead, the great majority of library users approach research by either performing a general subject keyword or title search on their topic, or by searching for a title of a book they know falls within the field they are researching.

This type of behavior is especially pronounced with younger and less experienced users (often referred to as the 'Google Generation' by library workers in our interviews) who simply type a query into the main search bar and hope for the best. These users seemed to treat library searches in a similar manner to popular online search engines from Google, Yahoo!, and the like – simply typing in a relevant keyword. Specifically, we found that users did not tend to use the extensive filtering options provided by most library systems when performing searches. On the University of California, Berkeley library system, the default search is by title. As a result, users often end up performing searches that return books with the words they are looking for in the titles, rather than books with a specific subject matter.⁶

The library workers we interviewed also expressed concerns that keyword search engines and the increasing reliance on information from online sources is creating a situation in which information seekers can become “lost”. Because they do not have frames of reference for the information they find online, they have trouble dealing with the different types of information and the requirements for finding them in different systems. The library workers argued that with the ability to just pull parts from a text, one can lose the context of that knowledge. They said that because users can access a variety of information types through the same medium, they often do not understand or recognize the medium or format in which the information was originally presented.

Related to this concern, we found that users often become confused or overwhelmed by search results if they vary significantly from what they expected to receive. For instance, a novice user was searching for periodicals on Hansel and Gretel and was confused when he received academic journal articles on psychological analyses of fairy tales. In providing user feedback, some patrons suggested that including more information about how specific books are categorized within the library collection and the subjects they fall under would be helpful.

Related to this suggestion, one of the main library strategies suggested by both advanced library users and librarians themselves was to use search features in the online catalog to get a general sense of the call numbers or sections of the library where relevant resources could be found. With this knowledge, users

⁶ This phenomenon of changing expectations and information seeking behaviors with the rise of online keyword search engines is well documented by many scholarly researchers and is discussed more extensively in our literature review below.

can go and physically browse the shelves of books in those sections of the library. Once in the stacks, users not only look for the specific book they searched for, but can also browse other books nearby. This has the added benefit of exposing searchers to other books that might not fit exactly what they expected to find but could also be useful.⁷ This system is not seen as a work-around. Rather, librarians regularly suggest it to patrons and one of the librarians interviewed emphasized that this is the strategy he employs when using the library's collection.

In addition, we repeatedly noted that both patrons and librarians identified books in a variety of different ways. Rather than remembering the title, author, or call number, users often asked for books by color, size, or location. Librarians confirmed that it was a common occurrence for them to receive requests for items like the “big grey book” on a certain subject. These findings suggested that an online interface that attempts to integrate some of these physical library browsing and searching behaviors might be extremely beneficial and could help remedy some of the problems and challenges many users face when attempting to negotiate large and often unfamiliar digital collections of information.

Specifically, we formulated the idea of a navigational and exploration tool that presents resources as if on a bookshelf. We envisioned a virtual bookshelf, complete with images simulating book spines from a library collection. As a result of our own discussions as well as consultations with the Open Library staff, we decided to move forward in exploring this sort of interface for the Open Library's digital collection of books. A summary of the development process for this first iteration of the Virtual Shelf is outlined in the Design and Development section.

Literature Review and Theoretical Grounding

In addition to our initial interviews and observations, we conducted an in-depth review of relevant literature concerning the theoretical basis for the Virtual Shelf's design and features. In this review, we aimed to gain a better understanding of our initial findings in the context of more empirical research that has been conducted on the subjects of library and online information search and retrieval. We focused our investigation on three main sections: 1) The influence of online search engines on library user expectations and experiences, 2) Digital resource evaluation and collaboration techniques, and 3) The role serendipity plays in information seeking endeavors. In addition to the literature review and analysis, we also administered an online survey concerning the usage of technology tools for individual and collaborative research endeavors. We received approximately 60 responses to this survey from a varying population that included undergraduate university students, graduate students, university professors, and professionals from a number of different fields. Anecdotes from the survey responses are referenced in the review.

The Influence of Online Search Engines

One of the most studied subjects in examining the library and its transition into the digital world is the library catalog in light of the expectations users acquire from other areas, specifically web search technologies on the internet. This is an important area to examine, as it reveals both how the current state

⁷ The importance of “serendipity” has also been examined extensively by researchers investigating information discovery. An in-depth review and analysis of this research is also presented in the literature review section below.

of the online library catalog was influenced by the technologies that it evolved from, but also provides insight as to the directions that its design may have to take to maintain relevance in the information world.

As library catalogs made the transition from the physical card catalog system to a digital catalog system, search functionalities remained highly constrained by the design choice to mirror the earlier system. Similar to card catalogs, these systems were built with controlled vocabularies and required Boolean search to navigate effectively (Antelman et al., 2006). These design decisions were not inherently wrong. Boolean searching allows very fine grained control over a user's search and when paired with an extensive knowledge of the system used to set up the subject classification, a trained user can focus in a great deal on the area of literature he or she needs. However, unpacking the previous statement helps explain many of the problematic aspects of library catalogs for the Google generation.

First, Boolean searching requires training to do it well (Antelman et al., 2006). This is something that a majority of users do not have. Numerous studies show that few users in online search contexts use Booleans, and even fewer use them in library catalog searches (Jansen & Pooch, 2001; Novotny, 2004). These results might be considered even more telling when considered in light of the fact that the studies reviewed by Jansen and Pooch (2001) occurred in the late 1990s, before post-Boolean search engines completely dominated online searching (Markey, 2007).

Secondly, controlled subject headings are a major stumbling block for users. Institutional classification systems provide advantages, including allowing records to be standardized, sorted, interlinked, and increasing interoperability between systems. If a system features correctly created controlled subject classifications and a user knows them, he or she can be confident in being able to access the majority of relevant results (Glushko, Maglio, Matlock, & Barsalou, 2008). However, this supposes familiarity with these subject classifications. Traditionally, librarians acted as intermediaries by translating patron information requests into queries understandable by the catalog systems. Yu and Young (2004) review a number of other studies dealing with subject searching. Within this review, they cite Larson's (1991) report that a lack of user understanding of Library of Congress subject headings as one of the major problems with the library catalog system, as well as Hunter's (1991) findings that 52.6% of zero-result searches resulted from the use of vocabulary terms not included in library controlled vocabularies. The fact that users are not familiar with controlled subject headings can hardly be considered surprising. The trend in web searching has been a progression away from categorized search and browsing engines toward keyword full-text engines, with Google being the most prominent example.

Compounding the confusion of what terms a user should be using to search as issues of understanding how these terms should be applied. Most library systems offer a variety of fields can be used to control where terms are applied. This is quite a difference from Google's single search bar. Both Novotny (2004) and Yu and Young (2004) describe studies in which users are thwarted by a lack of familiarity with a system in which inputs in a field will only search specific values in the system. In some cases, these are caused by confusing field labels, such as searching for the names of journal articles rather than title of the journal in a field labeled as "journal titles". Another common practice is simply treating the top most search box in a Google-like fashion and using it for keywords, regardless of its actual designation. This was a pattern that we also found when observing library patrons.

Another issue is revealed when examining the process of query refinement. Query refinement is important in both library catalog and online search contexts. One reference librarian we interviewed

noted that she had been trained in a process called a reference interview, noting that the question that patrons ask when they first approach the reference desk is often quite different than the information that they are actually looking for. The purpose of the reference interview is to help patrons formulate this question in terms useful for finding information. When attempting to tailor one's search in a search engine or library catalog system, users must examine the search results and attempt to deduce the proper way to adjust the query.

This, however, depends on users correctly evaluating the relevance of the results returned. Numerous studies show that online search users will only look at the first page or two of results in most cases. This is not an irrational move, as search engines like Google operate on a generally strong ranking algorithm which returns the most relevant items first. This expectation has been translated into searching library catalogs as well, with users expecting the most relevant results to be placed at the top (Yu & Young, 2004).

Users often remark that web searching outlets are better suited for those who are beginning search tasks or are not domain experts. Antelman et al. (2006) point out that this is not an accident, stating that many search systems were designed with the assumption that they would mostly be used for searching for already known items, and systems for browsing have been lagging ever since. If users have knowledge of the domain in which they are searching, this is not a major issue, as they might already have names of influential authors and issues in the field. This makes the catalog useful. Without this knowledge, however, users are stuck trying to guess at subject terms. This is no secret to librarians. As mentioned above, the librarians we observed and interviewed suggested that, rather than trying to conduct browsing within the catalog, an often more fruitful strategy is to attempt best possible keyword searches, mark down relevant call numbers from the results, and physically walk to the library stacks to browse the subject area. This suggests that online catalogs can be ineffective for many patrons without the supporting collection available for browsing. This is not unreasonable for most libraries and can provide benefits such as increased possibility for serendipity in searching (see the Serendipity section for further discussion). Doing this, however, requires extra effort in cases in which it is possible (something that a searcher will have to factor in when making decisions about where to go for information) and can create problems for libraries that rely solely on the catalog for searching and browsing (such as libraries with digital only holdings or holdings to which access is restricted).

Technologies are informed by the processes that came before them and the institutional context into which they are built. Digital catalog systems of libraries were built to replace the card catalog, and therefore inherited many of its limitations. However, it should not be assumed that the tension between the expectations of the Google generation and the current catalog systems are going unnoticed. The subject is under heavy study, and calls for change to the library catalog continue to emerge from both within and outside library institutions. A university library cataloger interviewed mentioned the creation of a field 520 in the Library of Congress MARC records. Rather than a specific value, this field is meant to contain a general summary and other information like reviews, intended scope and content, abstracts, and subject about a resource⁸. It was the cataloger's opinion that once this field was full text searchable, much of the rest of the book record information would be rendered obsolete. The UC Berkeley library is

⁸ See Library of Congress MARC Standards (<http://www.loc.gov/marc/bibliographic/bd520.html>) for more details on field 520.

also in the process of switching catalog systems and, as part of this switch, they have implemented a Google-style single bar search box on the front page of the site tied into all of holdings and subscription systems. If a system this large and entrenched in bureaucracy can make such changes, it may herald larger adoption of search systems that more closely resemble the web searches users are becoming more and more attached to.

Resource Evaluation and Collaboration Online and in the Traditional Library

In the traditional library, the majority of patron interactions involve physical artifacts (documents, shelves, signs) and other human beings (librarians, other patrons). While physical library behaviors increasingly include interactions with digital systems such as electronic library catalogs and article databases, the nature of the physical library still encourages significant interaction with tangible artifacts and other people. The physical nature of the traditional library provides many different metrics upon which patrons can base their evaluations of the quality and relevance of the resources they find – many of which are unavailable to users in online information search and retrieval systems. Conversely, online search systems provide new and different ways for users to assess the relevance and importance of the documents they find.

As discussed briefly above, people frequently cite concern over the validity of information found online and believe that information found in libraries can often be often more reliable, useful, and more suited for certain types of information searches (Fast & Campbell, 2004; De Rosa et al., 2005a, b; Griffiths & Brophy, 2005; Stemalszewska & Blandford, 2004). This points to one of the most basic ways library users have traditionally evaluated resource quality. Many users report that because libraries are collections developed and organized by “experts” in various domains and the fact that a resource is included in a library means that it meets a certain baseline level of quality and reliability (Blandford, Rimmer, & Warwick, 2006; Fast & Campbell, 2004; Griffiths & Brophy, 2005; Rimmer et al., 2007; Tenopir et al., 2003).

Many researchers also cite the importance of interacting with other library users in locating resources and assessing their quality and relevance. Professors and academic researchers routinely identify trusted colleagues and other researchers as one of the most important sources for resource evaluation (Adams & Blandford, 2005; Barrett, 2005; Blandford et al., 2006; Harley, 2007). In the physical sense, libraries have often been traditionally thought of as meeting places that allow this sort of rich interaction between colleagues and people with similar interest – meetings that may not occur in any other way than being physically co-located in the library (Blandford et al., 2006; Rimmer et al., 2007). Additionally, many patrons discuss the importance of the library in providing an inviting and invigorating environment for scholarship and exploration. The physical experience of being immersed in a scholarly atmosphere and literally surrounded by scores of information artifacts and other people engaging in similar activities can be inspiring and can create an environment conducive to productive research and analysis (Rimmer et al., 2007; Blandford et al., 2006).

In addition to the social affordances provided by the physical library space, the ability to interact with tangible information artifacts (books, pictures, articles, manuscripts, etc.) is another benefit of traditional libraries that many users identify as especially helpful for resource assessment and evaluation. Library users utilize physical cues from the condition and appearance of books and artifacts in the library to make

general conclusions or assumptions about various characteristics of certain resources and their relevance or importance. For example, age and popularity of an item can often be ascertained from a book's binding and how worn that binding is, as well as the book's checkout history (Blandford, 2006; Blanford et al., 2006; Makri et al., 2007). Also, if a library contains multiple copies of a certain work, users might reasonably conclude that the work of special importance or authority (Makri et al., 2007). For many researchers (especially those in humanities disciplines and working with historical documents), the importance of primary resources and original copies of books artifacts is deeply entrenched in their cultures (Rimmer et al., 2008). Many of these scholars report a feeling of excitement and authenticity that they can only get when interacting with physical items and that is often lost when viewing digitized versions (Blandford et al., 2006; Rimmer et al., 2007).

The physical layout of the library also provides patrons with clues to the relevance and utility of resources. Many patrons argue that browsing activities are much better supported by physical libraries than digital collections or search engines (Blandford, 2006; Blandford et al., 2006; Makri et al., 2007; Rimmer et al., 2007). Relevancy can often be judged by a resource's location in the physical library, which is organized according to subject (Makri et al., 2007). Because documents in the physical library are grouped according to relation and relevant documents are often co-located on surrounding shelves, users argue that browsing bookshelves in the library results in more productive browsing than trying to sort through long lists of search results (Rimmer et al., 2007). Online search results are often seen as overwhelming, in that many search systems provide users with large lists of results that are not necessarily organized in a meaningful way. Conversely, being surrounded by resources in the physical library is often seen as an opportunity for exploration, as the physical library organizes shelves by subject area or theme and often by resource type (Blanford, 2006; Blandford et al., 2006; Rimmer et al., 2007). This makes the task of browsing through a multitude of different documents much more manageable and provides users with a feeling of possibility in that they will eventually find something that is relevant – in contrast to digital catalog search results, which can frustrate users if they do not provide relevant resources in a timely manner or demand multiple query iterations.

When looking at academic scholarship in journals citation analysis is a commonly discussed method for evaluating the relevance and quality of a publication. Citation analysis is built on the premise that an article's or book's authority and importance can be largely determined by how many other writings cite the work in question. This is combined with the relative importance and reputation of those works and authors that cite the publication.

Another important avenue for resource evaluation is taking advantage of collaboration. Researchers often report that evaluating the quality of online and digital resources is a significant problem (Harley, 2007). Assessing the relevance of documents to a certain research or teaching endeavor is also an increasingly important and difficult task as the number of available digital resources online continues to grow at extremely high rates (Harley, 2007; Talja, 2002). In such an environment, recommendations and opinions of trusted colleagues become invaluable, especially with respect to information that may be outside a researcher's or team member's specific area of expertise (Harley, 2007; Poltrack et al., 2003; Spence et al., 2005).

However, this diversity of collaboration can be a double edged sword. Multidisciplinary endeavors are often especially fruitful environments for innovation because they allow for the juxtaposition of ideas

from multiple viewpoints and perspectives (Cummings & Keisler, 2005; Spence et al., 2005). Yet these collaborations often involve disciplines with different and sometimes conflicting organizational structures. This can lead to difficulties in coordinating research or design activities due to differing work practices, social and communication norms, etc. (Cummings & Kiesler, 2005; Poltrack et al., 2003). Communication and group coordination also becomes increasingly difficult for research and project teams that are distributed geographically, as traditionally, much of the productive inter-group discourse has occurred in face-to-face meetings (Cummings & Keisler, 2005; Poltrack et al., 2003; Spence et al., 2005). For example, Cummings & Keisler (2005) found that single university projects involving researchers from multiple disciplines tend to be highly successful, while multi-university projects tend to be less successful on average. The authors postulate that providing technology tools to manage and track task trajectories, to support ongoing conversation online, to support group decision making, and to support scheduling and presentation tasks would be helpful for geographically dispersed groups.

In addition to helping to evaluate information, collaborative systems can also be engaging for users. In their study of IBM's Beehive social network, Geyer et al. (2008) found that similar to other forms of media traditionally shared on social networking sites such as photos and status posts, lists could be shared and were embraced as a form of self expression. Even more interesting was the fact that shared lists generated more interaction and discussion than photos did.

Digital libraries and online information systems also provide functionalities that can, in some ways, mimic certain evaluation techniques common in traditional libraries, as well as provide new opportunities for resource evaluation and discovery. For example, online sharing and collaboration tools such as links e-mail resources or search results to colleagues provide opportunities for online collaboration between trusted colleagues and friends (Cooksey, 2004; Blandford, 2006; Rimmer et al., 2007). These remote collaboration options also provide the ability for colleagues who are geographically distributed to more easily interact with one another and share opinions about the resources they encounter. Integration of additional social network-type collaboration options (such as instant messaging, file transfer, collaborative resource collections, syndicated feeds, etc.) could also provide additional opportunities for productive dialogue and discourse. Additionally, some libraries are instituting electronic communication options for interacting with librarians and collections specialists. These services could provide users of digital library systems with opportunities to receive similar types of advice and guidance that they receive with query formulation and resource evaluation at traditional library reference desks (Stelmaszewska & Blandford, 2004). Finally, many digital databases and online search engines have begun providing citation linking services and options to read resource abstracts, summaries, and tables of content online, in addition to an increase in available full-text articles (Cooksey, 2003).

Serendipity

Serendipity, in the traditional literature sense, is often discussed as the occurrence of happy accidents or discoveries by chance (Cooksey, 2004; Foster & Ford, 2003). In terms of information seeking, serendipity is thought of as making unexpected discoveries of useful information. Foster and Ford (2003) review relevant literature concerning the importance of serendipity has played in the creative process for researchers in all different disciplines. They cite the work of Cobbledick (1996), for example, who argues that serendipity is a major factor in artistic stimulation, and Delgadillo and Lynch (1999) who discuss serendipity's value for historical researchers. They also report on work by Cory (1999), who argues that

serendipity helps reveal “hidden analogies” between information resources, which in turn help enable creative connections. Foster and Ford also discuss studies that support the role of serendipity in similar “connection building” roles in the social sciences (Merton, 1968), as well as serendipity’s contribution to scientific discovery (Myerly, 1980; Senoff, 1990).

While the benefit of serendipity for information seeking endeavors has been espoused by many researchers, most argue that the sort of serendipity related to information seeking is a bit different from the traditional notion of serendipitous discovery that occurs purely by chance. Theorists and researchers often argue that some degree of control or predisposition (whether conscious or not) on the part of the researcher plays a role in supporting serendipity. Often, this control is discussed in term of the notion a “prepared mind” – one which is open to new discoveries and unanticipated avenues of investigation (Cooksey, 2004; Foster & Ford, 2003). Scientists argue that discoveries like that of penicillin (Cooksey, 2004; Rosenman, 1988, cited in Foster & Ford, 2003) and X-rays (Cooksey, 2004) would not occur if those involved are not only been open to examining multiple perspectives and unanticipated avenues of investigation, but also have the ability to recognize serendipity when it appears.

Cooksey (2004) echoes Louis Pasteur’s belief that “in the fields of observation, chance favors only the mind that is prepared” (Bartlett, 1980) in her assertion that “It is not the observation or the bit of information *per se* that makes knowledge, but the mind that sees the significance and makes connections” (pg. 25). According to most theorists, preparing oneself to recognize and embrace the unexpected can help in enabling the fortuitous results of serendipity. Library users themselves also discuss the importance of certain personal cognitive strategies that can help induce serendipitous discovery. Foster and Ford (2003), for example, found that users reported embracing such strategies as employing a measure of “deliberate randomness” in their search and browsing activities in the library, as well as “making their own luck” through persistence and deliberate effort. Other users echo the importance of an open mind and willingness to recognize possible new or unanticipated avenues for exploration in their browsing activities (Blandford et al., 2006; Foster & Ford, 2003). Some even go as far as arguing that considering oneself as part of a specific discipline can actually hinder serendipity and that it is especially important to step back and take a broader view of the information search context, which can help in suggesting new angles of investigation that the researcher had not previously considered (Foster & Ford, 2003).

In addition to a cognitive or personality predisposition to serendipitous discovery, many researchers have argued that the ways in which resources are organized, categorized, and presented to information seekers can influence the possibility for serendipitous discovery. While library patrons often think of serendipity as mostly a product of chance, librarians and sophisticated library users recognize that the system in which resources are arranged and collected “prepares” those collections for the “happy discoveries” patrons often report (Cooksey, 2004; Rimmer et al., 2007). Cooksey (2004) discusses serendipity with an analogy to metaphor. She argues that “The metaphor does not tell us how to interpret itself, but it does suggest possibilities” (pg. 25). In a metaphor, the linking of concepts to one another “stimulates the reader to make connections...but the real power of a good metaphor is its ability to open the possibility of connections that have not previously been anticipated” (pg. 25). The same rings true for many in discussing the role of the library and the librarian in providing conditions necessary for serendipity (Cooksey, 2004; Foster & Ford, 2003; Rimmer et al., 2007). As Cooksey eloquently articulates, “Serendipity is the happy convergence of the mind with conditions. The scientist provides the mind, the librarian provides the conditions” (pg. 25).

Users studied often report that the organization of traditional libraries provides an environment conducive to serendipity. Most researchers discuss serendipitous discovery in the context of library browsing activities. Many users studied believe that serendipity is supported by the fact that library shelves are organized in a manner that places resources with strong potential for being related or relevant in close proximity to each other (Blandford, 2006; Blandford et al., 2006; Foster & Ford, 2003; Rimmer et al., 2007; Stelmaszewska & Blandford, 2004). From our interviews, it is apparent that librarians also support this belief and lend credence to it, as they often suggest that users record the call numbers for the most relevant search results from the library catalog and then go physically browse the stacks surrounding those resources for other relevant items. Users often recognize that unexpected discovery is often greatly influenced by librarians and those in charge of library classification schemes and that serendipity is also often related to a jogging of one's memory of relevant items that may have been forgotten:

... next to the art journals were the Garden History journals which I hadn't even thought of looking for, and they proved to be very valuable to me... Yes, it isn't simply luck that I found garden history because libraries are organised logically so that garden history journals are next to the art history journals and then next to the architecture journals... (Interviewee 31)... discovery is remembering really isn't it, so it is probably something that I knew once and there are traces in one's mind. Serendipity can happen if you go along a book shelf – you can find something that you ought to have known about, if books are organised thematically and that is serendipity... (Interviewee 32). (Foster & Ford, 2003, pg. 15).

As discussed in the section above, library users often report a positive feeling being surrounded by resources in the physical library yet find long lists of electronic search results daunting for effective resource evaluation. Similar opinions are reported with respect to serendipity, as many scholars feel that being surrounded by books on the library shelves presents an opportunity for serendipitous discovery, yet a similar experience with digital libraries and search engines can result in an experience that is viewed as unmanageable (Blandford et al., 2006; Rimmer et al., 2007). Some researchers also argue that digital collection searching and browsing can limit the possibility for serendipity through filters and rankings (Cooper & Prager, 2000, cited in Foster & Ford, 2003). Others posit that online and digital catalog search systems can serve to limit the number of available paths to relevant resources, which is another key factor in the possibility for serendipity (Cooksey, 2004; Huwe, 1999, cited in Foster & Ford, 2003).

The opportunity for serendipitous information discovery is often one of the main benefits that users cite for traditional libraries when compared to their digital equivalents (as well as online search engines). Still, many researchers and users also argue that digital search and browsing technologies might also be able to offer functionalities that could increase the possibility for serendipity. For example, digital catalogs and online databases are increasingly providing cross-links to resources for related works by the same author or in the same series, as well links to resources that are traditionally found nearby on library shelves (Cooksey, 2004). Google Scholar, for example, also provides links to "related articles" and articles that have been cited the current resource being browsed.⁹ Citation indices are also becoming more common in digital library systems (Cooksey, 2004). Additionally, many resource databases (and online search engines) are beginning to provide thesauri or term suggestions to help improve search queries (Cooksey, 2004). Digital collections also often provide access to a wider variety of information sources than traditional physical libraries (Rimmer et al., 2007). Also, the ability to sort search results or digital

⁹ See "Google Scholar Help," <http://scholar.google.com/intl/en/scholar/help.html> for more details.

collections based on a variety of different metrics could provide new ways for users to experience resource collections that the constraints of the physical library do not afford, which could theoretically lead to increased opportunities for serendipitous discovery.

Digital information systems also offer numerous new opportunities for collaborative information seeking and sharing, which is often cited as another source of serendipitous opportunity (Blandford, 2006; Cooksey, 2004). As mentioned above, many libraries are introducing electronic reference services where patrons can interact with librarians and collection specialists. Additionally, options to share resources, records, and search results with colleagues and friends via e-mail or social network-like functionalities can perhaps support a form of collaborative serendipity in which users can expand the variety of information sources they are exposed to. While social interactions in the traditional library are often enabled simply by virtue of co-location in a physical space, utilizing social features of online systems requires significantly more conscious effort, both in learning how to interact with the features and taking the additional time to use them. Accordingly, a significant challenge for these types of social interaction mechanisms in digital library systems will be encouraging users to utilize these functionalities enough to make them worthwhile.

Discussion

Declining usage of the traditional library and a transition to rapidly increasing online interactions has resulted in significant changes in the way that people search for information. The transition from physical to digital library interfaces means that many of the benefits users traditionally receive from physical libraries (such as community interactions, authoritative classification systems, personalized assistance, physically browsing of library stacks and tangible artifacts, opportunities for serendipity, etc.) are realized less often. Still, digital environments provide certain possibilities that the restrictions of the physical world cannot (including increasing opportunities for distributed collaboration, new ways of ranking and organizing search results and resource collection, access to a wider range of resources, immediate retrieval of resources and the ability to obtain many full text resources from anywhere, etc.). By focusing on improving online library interfaces and the searching and browsing tools they provide, we believe libraries (both physical and digital) can adapt to better serve changing user expectations. We hope that the Virtual Shelf interface serves as a step in this direction.

The Virtual Shelf: Design and Development

Declining library usage does not necessarily mean the end of the library as a central societal institution (as some would argue). To the contrary, digital libraries have the potential of maximizing the success of information seeking behaviors by introducing features and functionalities that integrate benefits from both traditional libraries and online information search systems – providing users with an experience that capitalizes on the best of both worlds. By designing our interface to take advantage of traditional library information seeking behaviors we hope to provide users with a digital environment that can better support serendipitous information discovery. By integrating tools for personal information organization and multi-user collaboration, we aim to encourage the interactive nature of research as well as promote discourse around information resources. The theoretical review summarized above, combined with our own previous research, serve as the foundation that our system is built upon, and informs the design

trajectories in both the model and the final envisioned design. Discussion of the Virtual Shelf prototype's development and features follows.

Target Population

When starting to design the Virtual Shelf, we began by considering the target user population for our application. The Open Library aims to serve as a starting point for all book searches on the internet,

The adventures of Huckleberry Finn (Tom Sawyer's comrade) ...
by Mark Twain
Published in 1896, Harper & Brothers (New York, London)

THE ADVENTURES OF HUCKLEBERRY FINN (TOM SAWYER'S COMRADE)
By MARK TWAIN
Illustrated
New York and London HARPER & BROTHERS PUBLISHERS

By statement: By Mark Twain [pseud.] illustrated.
Pagination: xi p., 1 L., 388 p.

LCCN: 10005880
LC: PZ5.C59 A6
OCLC: 2826199

BUY: Not available for this book.
BORROW: Your local library
BROWSE: Internet Archive:
- View
- Details
- Download
Google Book Search (partial)

CHANGE COVER
READ ONLINE

DESCRIPTION
TABLE OF CONTENTS

No table of contents available ▶ Add it!

Figure 1: Sample Open Library book page. Features include all the available metadata for the book (which any user can add to and edit), a cover image if available, links to buy, borrow, or browse the full text, and options for users to add to the description and table of contents fields.

effective ways for its users to explore and engage with the collection, especially in terms of both individual and collaborative information search and retrieval. With this context in mind, we developed three potential user personas to frame our design, keeping in mind the relevant tasks they would be likely to engage in while using a digital library. Briefly, these included:

- Joseph Chan, a technologically savvy graduate student researcher who is a veteran of the computer and information sectors. He handles most of the technical aspects of research for a university professor, including finding relevant documents on specific subject areas and organizing and presenting them to other members of the research group. He is experienced in Boolean search techniques, as well as the advanced search features of most search engines and is a veteran library catalog searcher.
- Professor Hacksworth, a tenured professor whose habits are deeply rooted in the traditional library setting. He only started using his computer regularly in the last couple years, as e-mail has gradually become the most common form of communication for other professors in his field. Although his experience with computer technologies is limited, he is an expert researcher and is intimately familiar with both the Library of Congress subject heading and the Dewey Decimal

providing metadata for the books in their catalog, as well as full text of some books, and links to other places online where the physical book may be found (including local libraries and online book merchants). (See Figure 1 for a sample Open Library book page.) The audience of a digital library is wide and varied and includes those interested in conducting academic research, as well as general users hoping to find books for pleasure reading. The Open Library's main hope for our project has been to enable new and more

classification system. He is also comfortable dealing with complex collections of resources and citation linking.

- Victoria Hutchens, a high school freshman who is a novice information searcher. She has never done any real research for school papers, but this year she has been assigned to do so for her history class. She is a bit nervous about this, as she has been to the library only a few times in her life and this has mainly been to find novels she might like to read for fun. She is active online, though, specifically on social networking sites and is comfortable using keyword search engines.

These contrasting personas and their information seeking needs are representative of some of the different skill sets and comfort levels our project's user base will likely possess.

Initial Virtual Shelf Prototype

After developing target personas and task scenarios, we embarked on creating an exploratory prototype of the Virtual Shelf application. This process was an iterative one, beginning with lo-fi mock-ups of very early conceptions of what a tool like this might embody. After a few mock-up revisions, we created a

semi-functional hi-fi prototype in Adobe Flash¹⁰. This early version featured an almost full-screen bookshelf full of spine images that users could scroll through. When a spine was selected, the book's cover image appeared in the center of the screen and some basic book metadata was presented in the bottom section of the screen. We envisioned a link on the cover image that would open the book's Open Library page in a new window. We also developed the idea of allowing users to add books they are especially interested in to a "Book Bag," analogous to the physical bag or briefcase people often carry with them in the library. Since this was a very early demonstration version, all of the data for

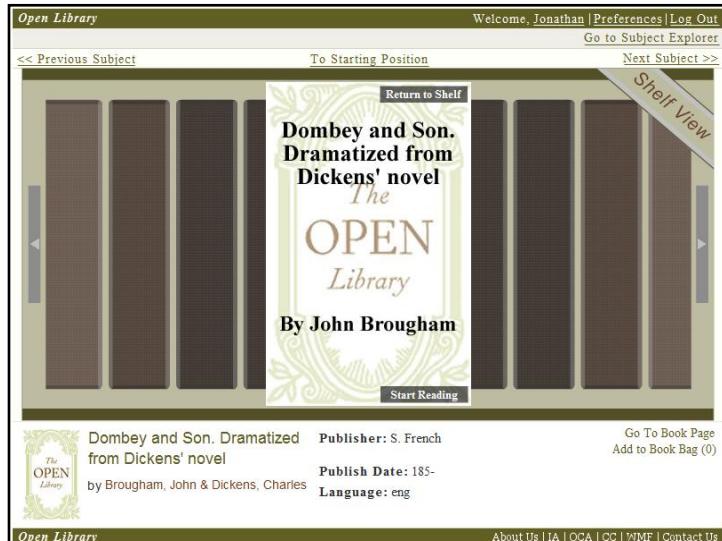


Figure 2: First iteration of the initial Virtual Shelf prototype.

the books included was hard-coded. (A screenshot of this first iteration can be found in Figure 2).

After a short period of user testing and heuristic evaluations, we settled on a few major changes and additions for the next iteration. (The second iteration was also a Flash-based application, but was built in Adobe Flex¹¹ – a Flash development environment that is well-suited for rapid prototyping because of its relatively quick learning curve and rich set of built-in visualization components.) The most significant design change we made was to shrink the shelf itself into a navigation-type panel at the bottom of the page. In confluence with this modification, we introduced the Book Details panel, which is a larger section above the bookshelf. When a book spine is clicked, the corresponding Open Library book page

¹⁰ <http://www.adobe.com/products/flash/>

¹¹ <http://www.adobe.com/products/flex/>

opens in the Book Details panel. (See Figure 3.) A complaint we often heard from users of traditional library and keyword search engines is that to explore detailed information about resources in a list or collection, users must follow links to the resources' individual webpages. This necessitates navigating away from the original collection being explored and can result in a disjointed experience, removing the book from its larger context. This experience contrasts greatly with the experience of browsing physical library stacks, in which one can pull books off of the shelf and peruse their contents, while still maintaining awareness of the other books on the shelves that surround them. Presenting the shelf visualization on the same webpage as the Book Details panel allows users to explore an entire collection of books without ever leaving that collection. Users are free to spend as little or as much time as they like reviewing individual books on the shelf and can quickly switch between books to review in greater depth – much like they can in a physical library setting. We also designed the bookshelf portion of the interface to be collapsible, so that users with limited screen real estate could temporarily hide the bookshelf contents and view the Book Details panel in full screen.

In the physical library context, basic information such as title and author(s) is printed on a book's spine. We also wanted to provide users with a quick way to gain similar basic information about each book on the Virtual Shelf. To accomplish this, we added tooltips that appeared when the mouse was placed over each spine. These tooltips displayed basic metadata (including title, author, year published, and number of pages) about the specific book (see Figure 3).

For this second iteration, we also wanted to capitalize on some of the affordances or cues provided by physical book artifacts. In our first iteration, book spines were all of uniform shape. For the second iteration, we hoped to have spine dimensions correlate to the actual physical dimensions of the books in the collection being browsed. However, dimension data was not available for the data sets we used to populate this prototype, so instead we approximated width based on the number of pages each book contained. Height was assigned randomly.

Finally, we expanded on the idea of the “Book Bag” for the first iteration by implementing functionality for users to add and remove books to and from a virtual book bag. Clicking on a button in the interface allowed users to toggle between viewing their custom collection and the original shelf. Books that had been added to the book bag would change color to indicate that they had been saved (see the red book spine in Figure 3). We also added a simple annotation option in which users could type text into a “Book Notes” field.

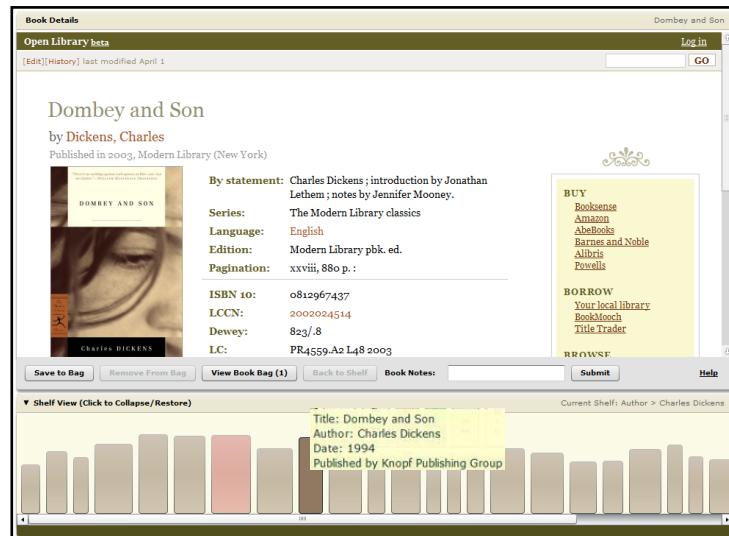


Figure 3: Second iteration of our initial prototype.

After completing the second iteration of the prototype, we conducted a small pilot test with 15 users. The participants were a mix of high school students, university undergraduate and graduate students, as well as a few post-university professionals. For this testing, we created two versions of the Virtual Shelf – one in which book spines were a single color and another in which spine color was randomized from a set of 10 colors. For the colorized version, we also attempted to add book titles to the spines. Due to time constraints, we were not able to rotate the titles to mimic the way titles are presented on real books, so the titles were oriented horizontally (see Figure 4.) We used a think-aloud protocol and also had users fill out a short post-testing survey. In these testing sessions, we asked users to perform simple hypothetical research tasks using one or both versions of the Virtual Shelf prototype (populated with sample search datasets from the Open Library) and a set of similar search results from the UC Berkeley library online catalog.



Figure 4: Third iteration of our initial prototype.

results placed on a continuously scrollable shelf, rather than as a paginated set of results. This finding served to lend a bit of credence to our hypothesis that users will be more likely to explore beyond the first 10 or 20 books in a collection in the Virtual Shelf interface, than they would when viewing text-based lists of keyword search results. In general, testers also reported that being able to save custom collections of relevant books from the shelf would be very useful for research tasks. One user also requested the ability to create multiple different book bags – a feature we had already been planning to implement in future iterations.

Opinion was nearly uniform as well in that the different sizing of the book spines helped to visually separate the books and made the shelf much easier to navigate. In addition, nearly every user assigned to the non-titled instance of the prototype requested the addition of book titles to the spine images. They stated that adding titles to the spine images would help users understand that the images were meant to represent book spines, that titles would help them keep track of the books they had already clicked on, and that titles would enable them to scan the collection more quickly. Correspondingly, users who tested the version with spine titles believed they were helpful, but the majority of these users disliked the manner in which they were presented. They disliked the horizontal orientation and word truncation. Based on these results, it was clear to us that the spine titles were an essential feature of the Virtual Shelf

Several conclusions about the various features of the Virtual Shelf arose from these pilot tests. First, a great majority of testers was enthusiastic about the bookshelf representation of a book collection and thought that an interface like this would improve the digital library experience. Another specific benefit mentioned by testers was the fact that they did not have to click back and forth between a search page and individual book pages. One specific user excitedly noted that he could not wait for an application such as this, as he was sick of having to open so many separate tabs in his browser. Many testers also preferred seeing search

interface, but that care should be taken to present them in a visually appealing and understandable manner.

Testers who saw both versions of the prototype generally preferred the one with multicolored spines, believing that color coding the spines made the books easier to remember and differentiate. From this, it became apparent that spine color is an important feature for this type of interface. Deciding what exactly color should represent in this context is a rather complex matter, though, and we recognized it would require significant thought, as color might be used for a wide variety of different functions (including, for example, grouping books based on various metadata fields, popularity data, etc.).

Testers also recognized the utility of the tooltips in presenting quick overview information about each book. There were a few aesthetic issues users found problematic though. Excessively long book titles could create tooltips that spanned almost the entire length of the bookshelf. Also, a few users complained that the tooltips could make browsing more difficult because they sometimes obscured the other spines nearby. Accordingly, we gathered that possible improvements to the tooltip implementation would be to truncate long book titles and to increase the transparency of the tooltip window. Additionally, most users were sufficiently confused by the annotation feature. Lack of visual feedback when a note was saved was cited as a major problem. Users also vocalized the importance of an indicator that identified which spines had already been annotated.

Initial Prototype: Final Version

Based on the feedback we received from the user testing sessions, we continued to revise the Flash-based prototype. During this time, we also presented our work at the CARL-IT North (<http://carlnit.blogspot.com/>) meeting at Santa Clara University in July of 2008. We also continued to discuss the Virtual Shelf project with the Open Library staff. For the final version of our initial prototype, we implemented or attempted to improve upon several Virtual Shelf features, including:

- Icon added to a book spine indicating annotation
- Possibility for multiple annotations per book
- Notes represented by yellow “post-it” notes at the bottom of the book panel
- Truncated titles in the tooltips
- Increased tooltip transparency
- Open-book icon to replace spine image when a book is clicked
- Shelf sorting functionality (implemented, but not fully functional)
- Saving custom shelves (implemented, but not fully functional)

An overview of the final version of our initial prototype can be seen in Figure 5. We presented and demonstrated our prototype at the Open Content Alliance's¹² annual meeting in San Francisco in October, 2008. The prototype was extremely well-received and, thus we decided to push forward with building a fully-functional production version of the Virtual Shelf.



Figure 5: Final version of the Flash-based Virtual Shelf prototype.

Virtual Shelf Requirements and Basic Design Features

Before embarking on the development of a production version of the Virtual Shelf, we outlined a set of basic requirements that our application should address, along with corresponding design features meant to support the various goals and tasks related to each requirement. These requirements follow directly from our experience building the first stage prototype, user testing feedback, as well as our theoretical research endeavors. Throughout this development process, we made sure to constantly consider the context in which our work was being created and our goal (as well as the Open Library's) to promote open access to information and information sharing. A summary of these requirements follow:

1. Integrate benefits of the physical library environment into the digital environment

As discussed extensively in our research above, stacks of books in the physical library provide information seekers with a great wealth of information, including physical attributes of books on the shelves, contextual organization and location cues, as well as the ability to quickly peruse books on the shelf for relevance and save them/check them out for later in-depth exploration. Much of this information, especially physical characteristics of books themselves, is lost in the online environment. Many researchers (especially those in the humanities and social sciences) also express a preference for immersion among shelves of books in the library than navigating long lists of electronic search results (the organization and presentation of which are often incomprehensible or unknown). Accordingly, we aim to bring some of these benefits of the physical library to our interface by providing the following features:

¹² <http://www.opencontentalliance.org/>

- A virtual bookshelf visualization, complete with images that represent the spines of books stacked on the shelf
- Quick preview of basic book metadata for each spine (in the form of mouseover tooltips)
- Allow users to browse records and full text scans without navigating away from the shelf
- Clear indication of what each bookshelf contains and how these contents are organized

2. Capitalize on the organizational affordances of the digital environment

While the library book stacks provide much information about the book collections they contain, they are still restricted by their physical nature in several ways. For example, re-organizing books on the shelves is an enormous endeavor for most library collections. Also, only one system of organization is possible in the library (usually some call number system) at one time.

Research has shown that providing information seekers multiple pathways to information can provide fruitful opportunities for discovery and multiple perspectives on a collection of resources can enable connections that were previously unseen. Users also conceptualize of information in different ways and preferences for how to visualize and experience collections of resources vary. The digital environment can eliminate many of the organizational restrictions inherent in the physical world, as electronic data can often be reordered and reorganized in a huge variety of different ways almost instantaneously. Therefore, we aim to capitalize on this affordance through the following features:

- Option to reorder books on a shelf by a wide variety of metadata fields, as well as physical attributes (book spine height and thickness, for example)
- Option for users to individually rearrange their bookshelves in whichever ways they choose

3. Enable users to easily create and organize custom collections of resources

In the physical library, patrons can collect and check out books or resources they find interesting or relevant to their search task. The ability to collect and organize relevant resources is an essential aspect of

productive research endeavors. Many existing digital library systems offer options for building lists of relevant books. For example, UC Berkeley's Pathfinder online catalog allows users to choose resources to save from lists of search results and then e-mail these results (see Figure 6.) In many of these

The screenshot shows the UC Berkeley Pathfinder online catalog interface. The search term 'tom sawyer' was entered. The results page displays 170 items found, sorted by Date, Author, Title. Each result entry includes a 'SAVE' button. The second result for 'The adventures of Tom Sawyer / <2007>' has its 'SAVE' button circled in red. At the bottom of the results page, there is a summary message: '2 Records Currently Saved' followed by a 'SAVED' link.

Figure 6: Save options in UC Berkeley's Pathfinder online catalog.

systems, though, users have no way to differentiate or group different resources in meaningful ways. They are provided with a single list of saved records, usually presented in the order they were saved. These lists also often don't persist across search sessions. A number of library and community websites have recently begun offering much more robust and useful collection building utilities. Goodreads¹³ and WorldCat¹⁴ are examples of sites let users create custom lists or "shelf" collections that persist and are tied to a specific user account (see Figure 7). In a similar manner, we aim to provide the following features in the Virtual Shelf:

- Option to save any book to any number of user-created custom bookshelves
- Opportunities to customize each bookshelf with at minimum a name and description
- Option to add comments or notes to bookshelves and the individual books on those shelves

4. Provide multiple options for collaboration

As discussed in our literature review, the physical library has traditionally been thought of as a space that supports and encourages rich interaction between colleagues and patrons who have similar interests or are pursuing related research endeavors. Interactions with trusted colleagues are often essential components of productive resource location and evaluation. Additionally, multidisciplinary and multi-institution endeavors represent great potential for innovative and unique scholarship. Often, though, the distributed nature of these teams presents a significant challenge in research coordination, dialog, and resource organization. Also, varied levels of technical proficiency and system familiarity among collaborators can also hinder productive

Rank	Title	Author	Rating	Comments	Votes	Date Added
1.	Blogging America : the new public sphere	Aaron Barlow	3.96	0	0	May 10, 2008
2.	Routledge handbook of Internet politics	Andrew Chadwick, Philip N Howard	4.08	0	0	May 10, 2008
3.	Civic engagement in a network society	Kai-Fu Yang, Erik Berndt	3.40	0	0	May 10, 2008

Figure 7: Collection building options on the Goodreads and WorldCat websites.

¹³ <http://www.goodreads.com>

¹⁴ <http://worldcat.org>

pe to mitigate some of these barriers to productive collaborative research with the following Virtual Shelf features:

- Unique and persistent universal resource identifiers (URIs) for each user-created bookshelf in (which allows for simple URL linking and e-mail or instant message sharing)
- Multiple privacy levels for user-created shelves (at the bare minimum, the ability to designate whether a bookshelf can be viewed by anyone visiting the Virtual Shelf website or whether access is limited to the bookshelf creator)
- Unique and persistent URIs for each user in the system – each user page will list that user's public bookshelves and provide direct links to them
- Option to create collaborative (multi-user) custom bookshelves

5. Provide a simple yet powerful user interface

As mentioned above, the Virtual Shelf's potential user base is varied – both in terms of technical proficiency and experience, as well as purpose. For our application to be beneficial across this user population, it must be easily navigated while still providing a wide enough set of useful tools that add value over traditional online library systems. Example features we hope will achieve this are:

- Keyword search bar (similar to all familiar online search engines) that returns search results as a virtual bookshelf – ensuring a low entry barrier for all user populations
- Options to iteratively refine search queries
- Simple, uncluttered user account pages

6. Produce a product that follows the philosophy of the Open Library

The Open Library is an organization with a definite ideological bent. They have a significant commitment to openness and accessibility of information. As the Virtual Shelf was built in collaboration with the Open Library, our goal was to produce a product that they would want to adopt. Ways we sought to accomplish this include:

- Supporting extensibility and repurposing as much as possible through proper data-modeling and
- Offering multiple options for sharing and repurposing bookshelf data outside the Virtual Shelf website
- Using open and non-proprietary technologies to build the application

Along with these requirements, we identified four specific types of bookshelves that we want to support in our application. These lenses served to frame our project concept and have proved useful as reference points throughout the process. They include:

1. The General Shelf

This is the shelf that the original concept was based on. The general shelf refers to placing a book in its context among all the records in the greater collection (sorted according to a single field).

As the purpose of this shelf is to mirror the stacks in a library, likely the sorting metric would be Library of Congress, Dewey Decimal, or some other call number system. The general shelf can be called an individual book's Open Library page and presents the context surrounding that book. When viewing this shelf, a user would theoretically be able to keep scrolling through the entire collection. This shelf poses a great number of problems for implementation and design, but it is a major goal for a complete implementation of an application like the Virtual Shelf.

2. Search Results

For a search results shelf, when a user performs a search and receives a set of books, a shelf containing those results is built. The user can sort these books in various ways and select books they find useful to save to their custom shelves. This is the type of discovery shelf that we have had the most success implementing in our prototypes.

3. User-Created Custom Shelves

Custom shelves are shelves that are created by users or groups of users out of books that they have found while browsing other virtual shelves. Users can order these shelves as they see fit, either by sorting based on various metadata fields or by dragging and dropping books into position. They can add titles, descriptions, and set privacy levels for these shelves. Public shelves can be found and possibly contributed to by other users. As mentioned above, user-created shelves add significant value to the research process by allowing collaboration and user created collection building, as well as letting providing valuable organizational and recall options for individual users.

4. Institutionally Created or Curated Shelves

Curated shelves are specialized shelves that might be created by groups of experts or institutions. They may contain a specific collection from a library, a collection of books on a certain topic or discipline, or the like. They are essentially custom shelves, except that they may have some custom sorting methods, and will likely receive some sort of marker that designates them as curated shelves.

Application Design and Features

The following section outlines the design and development process for the final production version of the Virtual Shelf prototype. The design features we included in the Virtual Shelf (<http://virtualshelf.org>) have been informed by the aforementioned research and initial prototype iterations, operating within the situational constraints of building a system that connects to an external content provider and time. This production version of the virtual shelf no longer utilizes static datasets or hard-coded spine images, but pulls all of its data from the Open Library's servers and dynamically generates bookshelves and book spines using this data.

In keeping with the philosophy of the Open Library, we are releasing this application under the GNU GPL open source software license¹⁵. Under this license, the code can be modified, extended, and repurposed. We hope to provide the Open Library with a product that they can show off as work of developers working with their collection and data services, as well as a tool that can be adopted by others, if they so choose, to be improved, adapted, or remade.

Languages, Environments, Platforms

The production version of the Virtual Shelf has been developed completely with all open source software that is freely available online. The application is built on a Ruby on Rails¹⁶ and MySQL¹⁷ database backend. The frontend takes the form of a web browser application using client-side Javascript and Ajax¹⁸. We moved away from a Flash-based application (on which our initial prototype was built) and decided to develop the Virtual Shelf as a web-based Javascript/Ajax application for a number of reasons. First, Flash and the Flex development environment are commercial applications and we want to allow anyone who wishes to interact with or build upon our work to be able to do so for free. Building the Virtual Shelf as a web application will also allow for greater openness and extensibility. The Javascript application is lighter weight and likely lowers the minimum processing power and memory demands for computers running the application. This is an important design consideration if the Virtual Shelf is to be accessible for institutions or users with machines that have limited power. Examples might include libraries, school systems, or users in developing regions.

We are using the JQuery¹⁹ Javascript library because other Open Library Javascript development efforts are also using it (including the Ajax Bookreader²⁰). Using JQuery as our Javascript library of choice ensures that our development is relatively consistent with similar development efforts being undertaken by the Open Library itself and allows for more effective collaboration. There is also a extremely active JQuery development community, many existing plugins, and a rich user interface library (JQuery UI²¹). JQuery plugins and applications are all licensed under either MIT or GPL licenses, which fits well with our goal and the Open Library's mission of creating an open source and easily modifiable and repurposable application.

For spine images, we worked with RMagick²², a ruby gem (plugin) for the ImageMagick²³ graphics program. This offered us the ability to dynamically generate spine images, and was chosen due to its ability to perform server-side text to image conversions. While generating images on the fly increases the processing time and overhead, the problem of creating spines for any book is non-trivial. Since we are dynamically generating the spines, we (or future designers extending our code) can change appearance

¹⁵ <http://www.gnu.org/copyleft/gpl.html>

¹⁶ <http://rubyonrails.org/>

¹⁷ <http://www.mysql.com/>

¹⁸ <http://www.w3schools.com/Ajax/Default.Asp>

¹⁹ <http://jquery.com/>

²⁰ <http://openlibrary.org/bookreader/tomsawyeradv00twairich#0>

²¹ <http://jqueryui.com/>

²² <http://rmagick.rubyforge.org/>

²³ <http://www.imagemagick.org/script/index.php>

attributes of the spine appearance to fit with various visualization schemes. This issue is discussed at greater length below.

A difficult design issue was whether to host the Virtual Shelf and its database ourselves or whether to attempt to utilize the Open Library's development server. Though the eventual goal is to integrate with the Open Library, the time required to perform such an integration in a relatively short timetable (approximately 3.5 months) would have been prohibitive and difficult to achieve. Important to note here as well is fact that neither of the authors was experienced at all in web or backend programming going into the project. We had to teach ourselves the languages as we went along. Thus, this slowed down the production schedule and limited some of the features that had originally been slated for implementation. Still, by designing the system to function as a standalone application that utilizes the Open Library's application programming interface (API) to retrieve collection data, it lends itself more easily to integration with other institutions or digital content providers. There are, however, drawbacks to this approach which are discussed at greater length in our Next Steps section.

Searching and Exploring

The procedure for searching for books in the Virtual Shelf system is straightforward – a single search bar can be found on the front page (see Figure 8) and in the top right corner of the site header. We chose this as our search interface because it follows the findings of our literature review, observations of library patrons, and interviews with librarians. As mentioned above, users have become accustomed to simple keyword-style search dialogs and may be stymied by or misuse complex field-based searching interfaces. In this, we hope to provide a low barrier to entry for exploration of the Virtual Shelf system.



Figure 8: Virtual Shelf home page (<http://virtualshelf.org>)

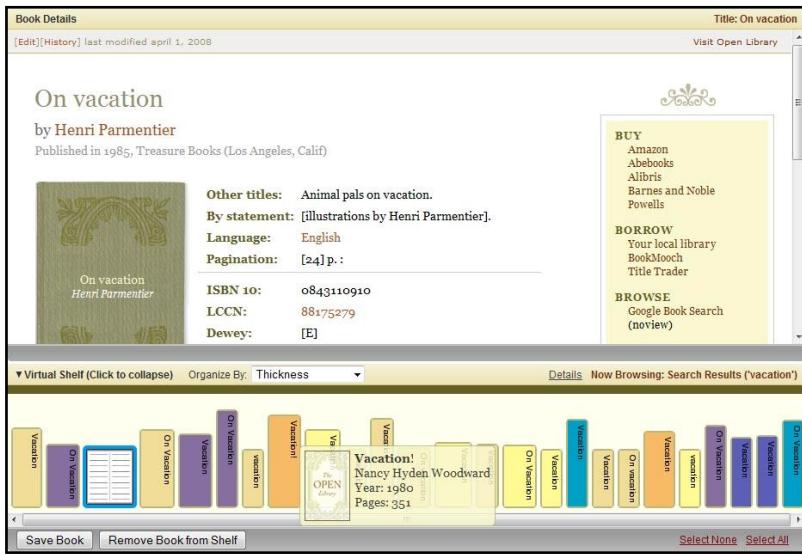


Figure 9: The Virtual Shelf bookshelf interface.

image has not yet been scanned into the Open Library collection) (see Figure 9). First and foremost, the tooltips offer identification information in circumstances in which the spine size is too small to display the full title. The tooltips also offers users much of the information that might be available at a glance in the physical library, the contents of which might help users quickly judge the potential usefulness of a resource. We have slightly increased the transparency of the tooltips compared to the Flash-based prototype so that adjacent book spines are not obscured when the tooltip is visible.

Also following the iterated design of our initial prototype, users can click on a book spine to see that book's Open Library in the Book Details panel above the shelf visualization. If there is full text available for the book, it is directly accessible from this page. This follows our design metaphor of browsing a shelf in a library or bookstore. Users can identify a book that might be useful or interesting and metaphorically open it to review its contents. This is done without navigating away from the shelf itself, so going from book to book to find what you need is easy and requires fewer clicks than other online catalogs and search systems.

Shelf Appearance

Search results are presented as spines arranged on a shelf. Again, the rationale for this is to support the types of physical browsing discussed in our research above. We aim to support a consistent browsing metaphor with users viewing all results at a glance and absorbing metadata such as size and color. Further, we want to promote serendipitous discovery, encouraging users to see a specific item they may be not necessarily be looking for. We hope that presenting search results in the context of the bookshelf might also encourage users to explore a greater number of results than they normally would when reviewing the traditional text-based lists characteristic of traditional online search interfaces.

The book spine images are generated based on the metadata available in each book's record. To determine the size of the spine, the Virtual Shelf application first checks whether the book contains dimensions metadata and calculates the size for the spine image based on that information. (We

The results of a search are displayed as book spines on dynamically generated virtual shelf that is presented at the bottom of the page, much like the design we settled upon for our initial Flash-based prototype (see Figure 9). When users place to mouse over a spine, a tooltip appears that shows (providing this metadata is present in the book's Open Library record) the book's title, author, year published, and number of pages. Unique to this version of the Virtual Shelf, the tooltip also displays a small image of the book's cover (or a default image if a book's cover

developed some relatively simple algorithms for transferring actual dimensions into spine pixel height and depth. These calculations are by no means comprehensive nor is the scaling completely accurate, but this method at least provides us with meaningful differentiation between the spines based on their actual physical characteristics.) If there is no dimensions data available or the dimensions data only contains height and width, the application next checks for page number data. If this is found, the number of pages the book contains is used to roughly calculate the thickness of the book spine. If height information is also unavailable, a simple ratio is applied to estimate the spine's height. If none of the above information is available, height and depth are randomized within the constraints of shelf space. Finally, we set minimum and maximum spine dimensions to ensure they no spines are too small to display at least a portion of the book's title or too tall to fit on the shelf. Titles are presented vertically on the spines, in the same orientation as physical books. The Virtual Shelf obtains the title from each book's Open Library metadata and uses dynamically attaches the title to the book's spine image. Titles are automatically truncated and wrapped according to the available space on the corresponding spine image (See Figure 9).

Currently, spine color is chosen by examining a book's Library of Congress call number or assigned subjects (if any exist in the record). The Library of Congress number is examined first, and a color is assigned based on the top level category²⁴ the number falls under. If there is no Library of Congress string, the values of the subject string are compared to the top level Amazon.com subject headings²⁵. The first subject that produces a match is used to assign color. If no matches can be made through the above system, a default color is assigned. It should be noted that we employed some degree of condensation with the subject heading to avoid having an excessive number of spine colors. Amazon has 32 top level categories, and once we removed colors that conflicted with our aesthetic scheme it was difficult to find 32 colors that were suitably easy to distinguish. We fully acknowledge that spine color is an aspect of the Virtual Shelf that would benefit from significant further exploration and is something we struggled with considerably. Spine color could be leveraged for a wide variety of different purposes and this treatment is meant to be a simple demonstration of one of its possible applications. (See the Next Steps section for further discussion).

Shelf Organization

As a default, search results are sorted by relevance (in the order they are received from the query to Open Library's API). We feel this is logical, as users are accustomed to online search engines that present the most "relevant" (according to their algorithms) results presented first. While this may be a logical starting point, we feel that providing users with a multitude of different organization options will result in greater opportunity for meaningful exploration of bookshelf collections and will allow users to explore the bookshelves in whichever ways they find most comfortable or fruitful. Accordingly, we provide a dropdown menu on the Virtual Shelf header that allows users to sort the bookshelf according to a wide array of metadata fields, including title, author name, year published, number of pages, Library of Congress call number, and Dewey decimal number. (Other sorting options can certainly be added in the future as well). Unique to the Virtual Shelf, though, and a feature that we feel sets this interface apart from other digital library collections is the ability to order the virtual bookshelves by physical

²⁴ See <http://www.loc.gov/catdir/cpso/lcco/>.

²⁵ See http://www.amazon.com/Subjects-Books/b/ref=sv_b_1?ie=UTF8&node=1000.

characteristics of the books. For example, we currently support shelf sorting by both spine height and thickness. We hope that some users who tend to identify with the physical characteristics of a resource might find these options productive. The decision to use a dropdown menu for sorting is deliberate, as it allows users identify the method by which a shelf is organized at any point in time. In addition to automatically sorting the shelf, users can easily create custom orderings in a simple drag-and-drop manner. To do so, the user must simply click and hold down the mouse over the spine to move, drag that spine to the desired location, and let go. The Virtual Shelf automatically creates an empty space for the book being moved. We feel that this is a simple, intuitive, and fast method for users to customize bookshelves in any manner they so choose.

Users can also save the state (order) of any custom bookshelf. Each custom shelf possesses a default order – the order in which the books are presented when the shelf is initially loaded. This is stored in the database. By default, this order is assigned according to the sequence in which books are added to the shelf, but users can save any order as their default preference. To alert users of this option, a clickable link appears on the shelf header whenever the shelf is reordered in a manner that differs from the default. Users are also prompted as to whether they would like to save their custom order if they have not already done so when they attempt to navigate away from the current shelf, to ensure that careful sessions of custom re-ordering are not lost by accident.

Custom Shelf Creation

Each time a book is clicked, that spine becomes selected and a blue border appears. Users also have the option to select (and deselect) multiple books at a time by holding down the control or shift buttons on the keyboard while clicking on each spine. Once at least one book has been selected, users can press the “Save Books” button. This brings up a dialog box that offers users a checklist of shelves they can save these books to (See Figure 10). Any shelf that is connected to that user’s Virtual Shelf ID is shown. One design feature that we considered was to have shelves that already contain the book(s) being saved to be selected and grayed out. We reasoned that this would be extremely confusing when attempting to save multiple books at once. Therefore, users are allowed to add a book to a shelf even if it already contains that book. Confirmation is sent from the server as normal, but books that already exist on the target shelf will not be duplicated. Below the checklist of shelves is a button to create a new shelf. If a user clicks this button, the dialog box expands to display the form for creating a new custom shelf (See Figure 10). When a new shelf is saved, its title is automatically added to the checklist of available shelves.

In addition to individually-created custom shelves, we plan to allow users to collaboratively build bookshelves by creating groups of users. While this feature has not been implemented at the time of writing, we plan to include it in the future and we have constructed our data model to support it (see the Backend Design section, which follows).

In the spirit of collaboration and easy repurposing of data outside the Virtual Shelf application, every bookshelf is represented as an Atom news feed which contains all the metadata and book contents for that shelf. This means that people can subscribe to bookshelves that interest them and will be alerted whenever a book is added or removed. The Virtual Shelf will also provide Atom feeds for each user account.

These are essentially feeds of feeds – which contain a feed of all of the bookshelves created by the user, as well as all of each of the book contents of those shelves. Whenever a user creates or deletes a shelf or alters the contents of a shelf, everyone who is subscribed to that user’s feed is alerted of the changes. We have also provided simpler and more traditional ways for users to share their bookshelves. For example, each bookshelf has a unique resource identifier (URI) or web address that is readable and persistent. Users can simply copy the URI (URL) for a shelf and e-mail it anyone they wish to share the shelf with – a very common practice of academic researchers. The Book Details panel also displays the Open Library address for the book currently being viewed. In addition, users can even point directly to a specific book on a specific Virtual Shelf, as the shelf’s URI changes to reflect whichever book is currently active. Not only does this let us handle browser “back” button clicks more effectively, but it makes more precise linking possible from outside the Virtual Shelf system.

Backend Design

The backend design of our system aims to strike a balance between trying not to duplicate the Open Library’s full data structure while still supporting export and extensibility. A representation of our underlying data model can be found in Figure 11 and we briefly discuss a few essential features of it here.

In our model, author objects have a raw name field (that the Open Library provides), as well as first and last name fields. We attempted to parse out last name and first using regular expressions so that book spines could be sorted by author last name. We recognize, though, that names can appear in a number of different formats (which vary widely depending on language and nationality), and that our method is biased towards a Western naming structure. A more robust algorithm could certainly be employed in future versions that attempts to account for different cultural patterns. We keep the full (raw) name field for display on the tooltips and for export in the shelf Atom feeds.

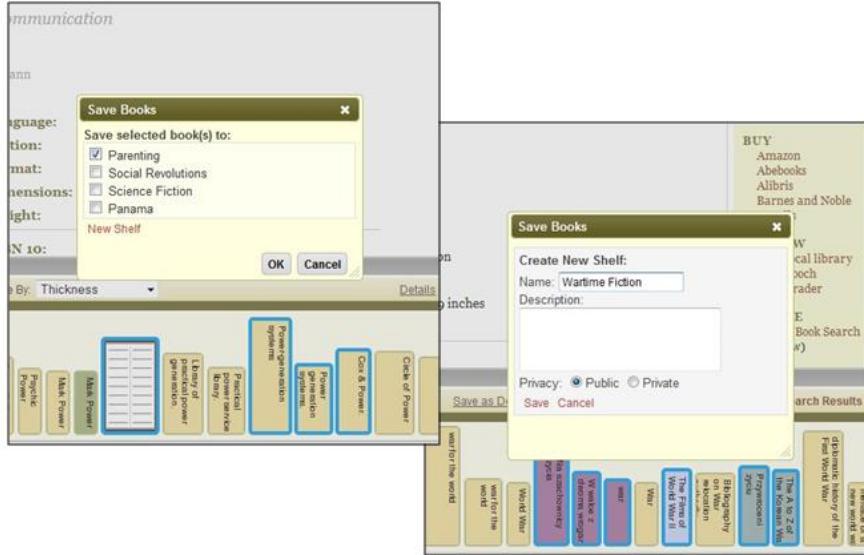


Figure 10: Custom shelf creation and book saving dialogs.

Book objects in the Open Library have an impressive number of possible metadata fields. As mentioned, it is not our intention to duplicate all of the information in the Open Library records, so our book model only captures data fields that we need to display books on our shelves or those that we see a potential use for in the Virtual Shelf application or for developers who might

want build applications based on our data.

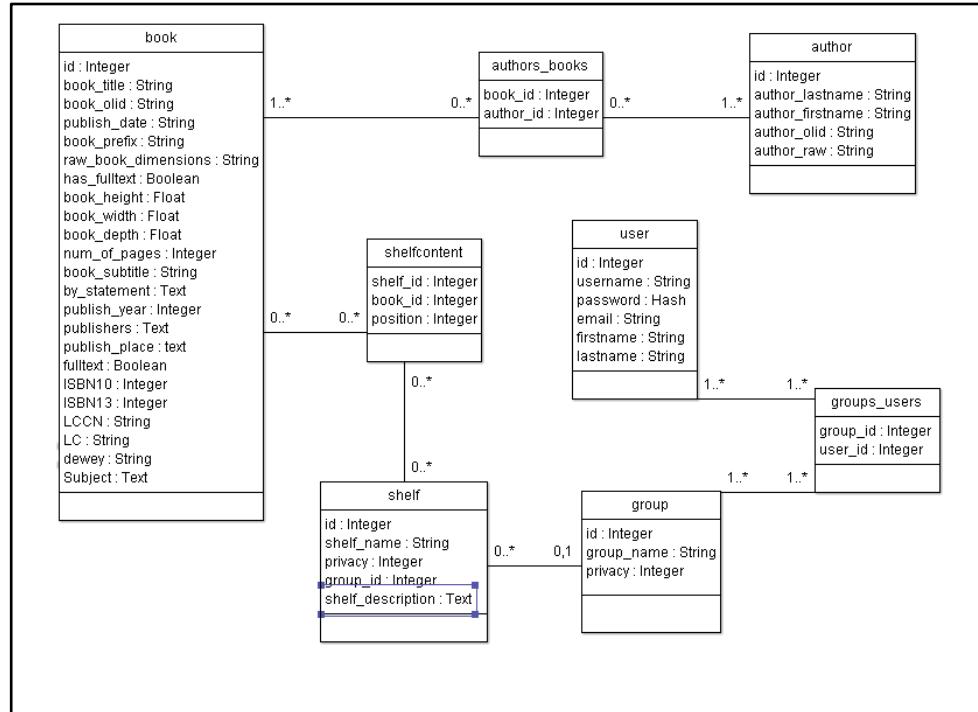


Figure 11: Virtual Shelf data model.

As mentioned briefly above, a modeling decision that is important for ensuring extensibility is the inclusion of groups. In our application, shelves are not actually created by users, but by groups. When a users signs up for a Virtual Shelf account, he or she is automatically assigned a group that only contains themselves. It is through this group identification that they create shelves. While in this version of the application all this does is add an extra step, it is essential for allowing for the creation of collaborative shelves in the future. (See the Next Steps section for more information.)

We have attempted to build the application to follow RESTful²⁶ web communication principles as much as possible. In this vein, each bookshelf, book, author, user, and group object has a single, unique, readable, and persistent (non-state dependent) URI. Having resources properly formatted and accessible makes pointing to and requesting these resources simple, which is of great importance to our project, as supporting extensibility and repurposing is high on our list of priorities. Also, all the client-server communications and data transfers in our application are accomplished with standard HTTP CRUD (create, edit, update, delete) requests.

Discussion and Next Steps

Currently, there several features that we feel either need improvement or we have not yet had the opportunity to implement fully. One current major stumbling block in the Virtual Shelf design, for example, is a limitation on shelf length. Currently, it is necessary to limit the number of books returned

²⁶ See <http://tomayko.com/writings/rest-to-my-wife> and <http://www.infoq.com/articles/rest-introduction> for more information about Representation State Transfer (REST) principle.

by each search in our system. (At writing, we are currently limiting searches to 25 records.) This is due to the fact that the processing time needed to dynamically create spine images not already in our database is rather large. Requesting too many books can actually cause a server timeout and the bookshelf can take a very long time to load. (Building a shelf of 25 search results currently takes approximately 20-50 seconds on a high-speed internet connection, depending on server traffic and bandwidth). Moving forward, this problem could be solved different ways – either through multiple successive calls to the Open Library’s server with small set lengths and rendering each set as they are returned, or by integration with the Open Library’s system which is the best solution and our eventual goal. Currently, an application such as ours is pushing the boundaries of what is possible in web-based, distributed systems. Increasing bandwidth, as well as tighter coupling between content providers and external developers in the future will likely make systems like the Virtual Shelf more feasible.

Another major limitation of the Virtual Shelf at the time of writing is that it currently only runs correctly in the Firefox web browser. We will be working to ensure better cross-browser compatibility in the near future. We also anticipate that more uniform web format standardization, like the widespread adoption of CSS3²⁷ for example, will help make cross-browser development much simpler. (CSS3 will also likely make aesthetic enhancements that currently rely on Javascript, for example, straightforward and may also make dynamic image creation in an application such as ours easier).

Perhaps the feature most lacking in this design is the ability to view a book as it would appear in a library, namely in the context of all other books, sorted by Library of Congress call number or some other classification system. The plan for this feature was to be able to view a book’s Open Library page and choose to view that book on the general library shelf. The book would appear in the middle of the shelf with the closest records populated on either side. Not only would this provide much of the functionality of browsing the stacks in a physical library, but by allowing users to choose their method of sorting, it would be the equivalent of reordering all the books in the library on demand. As this is the original idea that inspired this project, implementing it as soon as possible is a high priority. The reason it is not present in the current version is due to the difficulty of performing such a search from an external content provider’s system. Searching by ranges is not supported yet in Open Library’s API, and the processing time associated with the multiple requests, parses, and sorts required is beyond the capabilities of our current system. However, as we hope to see a version of the Virtual Shelf implemented within the Open Library, we definitely see the inclusion of this feature occurring in the future.

Due to the fact that the Open Library draws its records from so many sources, the quality and completeness of records is highly variable. The only field that is actually required for a book to exist is a title, and there are many with nothing but. On the other hand, there are a number of records with a great deal of data. Most records fall somewhere in between. Further compounding the problem, however, is the fact that records are pulled from a variety of sources. This means that there are no guarantees of consistent formatting. This created challenges in a number of areas, as discussed above. The fact that the Open Library is truly an “open” library (to which anyone can contribute), though, creates a wonderful opportunity for an application like the Virtual Shelf, in that it can encourage users to edit or add book information they might possess that is lacking for a specific record without having to leave the context of

²⁷ <http://www.w3.org/TR/css3-roadmap/>

their information searching behaviors. The collaboration options afforded by the Virtual Shelf will also hopefully encourage meaningful and increased dialog around the Open Library collection, which can serve to inspire more users to contribute content and help build this amazing resource.

One of our original ideas for displaying the shelf was book spines being assigned the color of the actual book's binding or a scanned image of the actual spine. This was not possible because spine images are not currently available in the Open Library's collection (although there are plans to begin including this in their scanning activities). However, we still believe that if this were possible, it might be a valuable option. As discussed in our research section, appearance is one of the ways users remember books, and this could provide an extremely effective tie to the books as real, physical objects. While it was not possible to have this level of appearance differentiation in our current version, it might be possible in the future. Smaller curated collections might be also able to provide the requisite data or spine scans, and since these collections often exist physically as well as digitally, they would be doubly valuable.

We also feel that the application would benefit from further examination of how to assign spine color. Color is one of the most powerful tools available to a visualization designer and therefore would benefit from more attention. One direction that could be explored is expanding the mapping of colors in such a way that subjects that are conceptually similar could have similar colors. Also, varying color intensity based on other metrics such as popularity data or user ratings, for example, are other option. These are two values that do not currently exist in the Open Library, but their inclusion has been discussed. This data could also be obtained by examining usage patterns of our own users. Another idea that was discussed with the Open Library staff and other advisors was simply turning control over to the users and letting them pick their own colors to represent personal methods of classification.

In addition to the above ideas, extensive user testing of the Virtual Shelf application is needed. We would like to refine the aesthetics and tweak variables such as maximum and minimum size, title font, and title size to ensure maximum usability. While we were able to do some of this testing between the current and previous prototypes, we are keenly aware that our design would benefit from user feedback in all areas, which we have not yet had time to seek.

While our literature review and research indicated that people are most comfortable with a Google-style keyword search bar, providing options for more advanced query refinement and filtering would most likely be beneficial. For example, the Open Library has a relatively effective faceted filtering system already in place. Unfortunately, we are not able to currently access this system through remote communications. The lack of filtering features is currently a significant shortcoming of our shelf system, but it can be fixed either through integrating with the Open Library's architecture (as is our plan) or by implementing our own faceted system in the future.

Another watershed feature that we were unable to finish at the time of writing is the account and user authentication system. The system is envisioned with users maintaining accounts that allow shelves they create to be associated with their group and user IDs. Ideally, these would be the same accounts that they have in the Open Library, or whatever institution is providing content. The reason setting up this system is a watershed is that it allows the implementation of a number of features that are framed in the data model, but not yet present. We plan to finish implementing user accounts in the near future.

Account systems would also make the collaboration features mentioned in the design section possible. Users are not associated with shelves directly, but instead are members of some number of groups, which are in turn associated with shelves. While every user has at least one group (of which they are the only member), users will eventually be able set up and join multi-users groups as well. The members of the group would have editing control over shared shelves. As mentioned in our research, collaborating around shelves can be useful for accomplishing group goals, spurring productive discussion and dialog, and create significant community value. Once this system is in place, venues for interaction around a shelf could be expanded, with shelf pages becoming mini forums or discussion boards.

An account system would also make possible the implementation of privacy controls. Shelves could be assigned an access level. A scale for this might range from shelves restricted to their creator group, to shelves that can be seen but not edited, to shelves that can be edited by any user on the system.

Finally, having the above two systems in place would allow us to capitalize on the value created by users producing custom shelves. For example, as the collection of user-created shelves grows, searches on the system could be expanded to include the titles, notes, and descriptions of bookshelves themselves, instead of simply books. This would add value in a number of areas. By allowing users to discover shelf collections that are relevant to their search, they can take advantage of and engage with the ordering and sorting done by others. Users finding themselves on these shelves are likely to find other relevant resources, and may take the time to leave a comment or add to the shelf if it strikes their interest and it is marked as open. Book info pages could also display a list of all the bookshelves each book has been added to in the system.

A feature present in the Flash-based prototype that we have not yet have time to implement at <http://virutalshelf.org> is the creation of book notes. While the Open Library allows users to edit a book's record, book notes would be specific to the user who created them. These would have to be stored in the same database as the users, and be tied to the local book records. This system would only improve once the Open Library finishes development on their book reader, which would also allow users to annotate specific pages of full-text books.

As noted, one of the major goals of this project is to create a framework that can be extended and repurposed. While the system is currently built to supplement searches on the Open Library, we envision other possible uses for it. First, it could be implanted in other libraries in either its current form as a general search visualization tool, or as a way of displaying specific collections. It would provide a distinctive advantage in circumstances in which users cannot easily browse the physical holdings. Examples of this include library holdings are not open to the public or any collection with restricted access. Further, altering the spine appearance is not a difficult task, meaning that the Virtual Shelf could easily be repurposed to provide shelf-like browsing for non-book collections such as museum collections or archeological artifacts. Other interesting uses can result from the Virtual Shelf's data export capabilities. Shelves could serve as a way of building collections that could be exported and mashed up in interested ways, including for example matching books to geographic locations or locations in time using mapping and or timeline software. Bookshelf feeds could also be embedded in blogs or social networking profiles, providing valuable identity and social capital for people to engage with.

Finally, there are a number of visualization features we would be interested in exploring for future development of the Virtual Shelf. One of the highest on the list and most requested by reviewers is the

inclusion of multi-level shelves. We have considered a number of ways this could be executed. One idea is allowing the shelf to be “maximized” to take up the whole window and display multiple rows. Another is allowing users to drag the whole shelf to the side of the screen, which would transform it into a vertically-oriented multi-level shelf that would span a portion of the browser’s width and stretch the full height.

Conclusions

The Virtual Shelf application (available at <http://virtualshelf.org>) is the product of two years of extensive research, development, and design iteration. Beginning from a simple observation that users do not always use online library catalogs in quite the way those search systems seem to support, the project became an exercise in user interface design, data modeling and backend architecture design, user testing and design iteration, academic research, and market review. Over the course of the project we learned a plethora of design techniques and practices, three coding languages, and gained a solid sense of what is required to move a project from the idea stage to production.

There are a great number of steps we would like to take should we continue the project. Even if we cannot address them all, it is our hope that by releasing our software in an open fashion to the Open Library and related design communities, it can inspire, and be adopted, extended, and re-imagined by others. We believe that an integral component of resource collections like libraries (especially the powerful bookshelf metaphor) has largely been ignored by digital interfaces until now. We hope that the Virtual Shelf will serve to reinforce some of the value physical resource environments contribute to information seeking and searching behavior and will inspire further exploration into how features of these environments might benefit online interfaces in the future.

Acknowledgements

We would like to deeply thank the many people who provided guidance, help, and encouragement along the path of this project. These include, but are not limited to our Masters Project Advisor, Eric Kansa, Rebecca Malamud, Aaron Swartz, and Anand Chitipothu at the Open Library, School of Information Professors Tapan Parikh, Jenna Burrell, Marti Hearst, Coye Cheshire, Kimiko Ryokai, and Erik Wilde, our fellow classmates, as well as our families for their unending support.

References

- Adams, A. & Blandford, A. (2005). Digital libraries' support for the user's information journey. In *Proceedings of the ACM/IEEE JCDL* (pp. 160-169). ACM.
- Antelman, K., Lynema, E., & Pace, A. K. (2006). Toward a 21st Century Library Catalog. *Information Technology and Libraries*, September, 128-139.
- Barrett, A. (2005). The information seeking habits of graduate student researchers in the humanities. *The Journal of Academic Librarianship*, 31(4), 324-331.
- Bartlett, J. (1980). In E. M. Beck (Ed.), *Familiar Quotations, 15th Ed* (pp. 591). Boston: Little Brown.
- Blandford, A. (2006). Interacting with information resources: Designing libraries for learning. *International Journal of Learning Technologies*, 2(2-3), 185-202.
- Blandford, A., Rimmer, J. & Warwick, C. (2006) Experiences of the library in the digital age. In *Proceedings of the Third International Conference on "Cultural Convergence and Digital Technology"*. Athens.
- Cobbledick, S. (1996). The information seeking behavior of artists: exploratory interviews. *Library Quarterly*, 66(4), 343-372.
- Cooksey, E. B. (2004). Too important to be left to chance – Serendipity and the digital library. *Science & Technology Libraries*, 25(1/2), 23-32.
- Cory, K. A. (1999). Discovering hidden analogies in an online humanities database. *Library Trends*, 48(1), 60-71.
- Cummings, J., Kiesler, S. 2005. Collaborative Research Across Disciplinary and Organizational Boundaries. *Social Studies of Science*, 35(5), 703-722.
- Delgadillo, R. & Lynch, B. P. (1999). Future historians; their quest for information. *College and Research Libraries*, 60, 245-259.
- De Rosa, C., Cantrell, J., Cellentani, D., Hawk, J., Jenkins, L., & Wilson, A. (2005). *Perceptions of Libraries and Information Resources*. . Dublin, OH: OCLC Online Computer Library Center, Inc. Available at: <http://www.oclc.org/reports/2005perceptions.htm>.
- De Rosa, C., Cantrell, J., Hawk, J., & Wilson, A. (2005). *College Students' Perceptions: Libraries & Information Resources*. . Dublin, OH: OCLC Online Computer Library Center, Inc. Retrieved May 10, 2008, from <http://www.oclc.org/reports/perceptionscollege.htm>.
- Fast, K., & Campbell, G. (2004). "I still like Google": University student perceptions of searching OPACs and the web. *Proceedings of the American Society for Information Science and Technology*, 41(1), 138-146.
- Foster, A. & Ford, N. (2003). Serendipity and information seeking: An empirical study. *Journal of Documentation*, 59(3), 321-340.
- Glushko, R.J., Maglio, P.P., Matlock, T., & Barsalou, L.R, (2008) Categorization in the Wild, *Trends in Cognitive Science*, 12(4), Pages 129-135.
- Griffiths, J. R. & Brophy, P. (2005). Student searching behavior and the web: use of academic resources and Google. *Library Trends*, Spring, 539-554.
- Geyer, W., Dugan, C., Dimicco, J., Millen, D. R., Brownholtz, B., & Muller, M. (2008). Use and reuse of shared lists as a social content type. *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (pp. 1545-1554). New York, NY, USA: ACM.

- Harley, D. (2007). Use and Users of Digital Resources: A survey explored scholars' attitudes about educational technology environments in the humanities. *EDUCAUSE Quarterly*, 30(4), 12-20.
- Hunter, R. N. (1991). Success and Failures of Patrons Searching the Online Catalog at a Large Academic Library: A Transaction-Log Analysis. *RQ*, 30: 395-402.
- Huwe, T. K. (1999). New search tools for multidisciplinary libraries. *Online*, 23(2), 67-70, 72-74.
- Jansen, B. J., & Pooch, U. (2001). A review of web searching studies and a framework for future research. *J. Am. Soc. Inf. Sci. Technol.*, 52(3), 235-246.
- Larson, R. R. (1991). The Decline of Subject Searching: Long-Term Trends and Patterns of Index Use in an Online Catalog. *Journal of the American Society for Information Science and Technology* 42(3), 197-215.
- Makri, S., Blandford, A., Gow, J., Rimmer, J., Warwick, C., & Buchanan, G. (2007). A Library or Just Another Information Resource? A Case Study of Users' Mental Models of Traditional and Digital Libraries. *Journal of the American Society for Information Science and Technology*, 58(3), 433-445.
- Markey, K. (2007). The Online Library Catalog: Paradise Lost and Paradise Regained? *D-Lib Magazine*, 13(1/2). Available at: <http://www.dlib.org/dlib/january07/marky/01marky.html>.
- Merton, R. K. (1968). *Social theory and social structure*. New York: Free Press.
- Myerly, R. C. (1980). Real world of industrial chemistry: serendipity and discovery. *Journal of Chemical Education*, 57(6), 437-438.
- Novotny, E. (2004). I Don't Think I Click: A Protocol Analysis Study of Use of a Library Online Catalog in the Internet Age. *College & Research Libraries*, November, 525-537.
- Poltrock, S., Grudin, J., Dumais, S., Fidel, R., Bruce, H., and Pejtersen, A. M. (2003). Information seeking and sharing in design teams. *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work* (pg. 239-247). New York: ACM.
- Rimmer, J., Warwick, C., Blandford, A., Gow, J., & Buchanan, G. (2007). An examination of the physical and the digital qualities of humanities research. *Information Processing and Management: an International Journal*, 44(3), 1374-1392.
- Rosenmann, M. F. (1988). Serendipity and scientific discovery. *Journal of Creative Behaviour*, 22, 132-138.
- Senoff, C. V. (1990). The discovery of [Ru(NH₃)₅N₂]²⁺⁺: A case of serendipity and the scientific method. *Journal of Chemical Education*, 67(5), 368-370.
- Spence, P. R., Reddy, M. C., & Hall, R. (2005). A survey of collaborative information seeking practices of academic researchers. *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work* (pp. 85-88). New York: ACM.
- Stemalszewska, H. & Blandford, A. (2004). From physical to digital: A case study of computer scientists' behaviour in physical libraries. *International Journal of Digital Libraries*, 4, 82-92.
- Talja, S. (2002). Information sharing in academic communities: Types and levels of collaboration in information seeking and use. *The New Review of Information Behavior Research*, 3, 143-160
- Tenopir, C., Hitchcock, B., & Pillow, A. (2003, August). Use and Users of Electronic Library Resources: An Overview and Analysis of Recent Research Studies. Report for the Council on Library and Information Resources. Available at: <http://www.clir.org/pubs/reports/pub120/pub120.pdf>.
- Yu, H., & Young, M. (2004). The impact of web search engines on subject searching in OPAC. *Information Technology and Libraries*, December, 168-180.

APPENDIX

The purpose of this appendix is to outline the steps necessary to replicate, extend, or modify the Virtual Shelf application. This document walks through the steps involved in creating the Ruby on Rails framework for the program and outlines a list of existing JQuery plugins being utilized in the applications user interface. All the other client-side code, stylesheets, and custom Javascripts the authors created can be found at the project website (<http://virutalshelf.org>).

Program Components

The underlying architecture and backend structure of the program is built in Ruby 1.8.6, Rails 2.2, and MySQL version 5.0. For the creation of spine images we worked with the RMagick gem for ImageMagick. This offered us the ability to dynamically generate spine images, and was chosen due to the ability to perform text to image conversions.

We are using the JQuery (<http://jquery.com/>) Javascript library because other Open Library Javascript development efforts are using it (including the Ajax Bookreader). Using JQuery as our Javascript library of choice will ensure that our development is consistent with similar development efforts being undertaken by the Open Library itself and allows for more effective collaboration. There is also a very active JQuery development community, many existing plugins, and a rich user interface library (JQuery UI, <http://jqueryui.com/>) of which we are utilizing specifically for the drag and drop functionality of the book spines, as well as styling of the dialog boxes. JQuery plugins and applications are all licensed under either MIT or GPL licenses and our application is released under the GNU GPL license version 3 (<http://www.gnu.org/copyleft/gpl.html>). A copy of our license information and disclaimer can also be found at <http://virtualshelf.org>.

Following is a list of JQuery plugins we are using in Virtual Shelf application:

- Was initially using JQuery Corner plugin (<http://www.malsup.com/jquery/corner/>) to do rounded corners on the interface, but switched to the JQuery Corners plugin (<http://www.atblabs.com/jquery.corners.html>) because the corners are much smoother, provide better anti-aliasing, is simple and lightweight, and can be applied to any DOM element including the tooltips.
- The UI.Layout plugin (<http://layout.jquery-dev.net/index.html>) is being used to created the two panels (Book and Shelf) and also the collapsible function of the shelf. The plugin allows us to specify a center content div with header, footer, left, and right panels that are toggleable and completely customizable. Iframes as panel content work perfectly, which is essential as we are loading the Open Library book pages in an Iframe. Resizing/toggling are incredibly smooth. Toggling can be assigned to custom buttons, etc.
- We are using JQuery Tooltip plugin (<http://jquery.bassistance.de/tooltip/demo/>) to create the spine mouseover tooltips. We experimented with and considered using other tooltip plugins (clueTip, <http://plugins.learningjquery.com/cluetip/demo/>; BeautyTips, <http://www.lullabot.com/files/bt/bt-2latest/DEMO/index.html>; Simpletip, <http://craigsworks.com/projects/simpletip/>). The Tooltip plugin seemed most appropriate after reviewing all of the features. It has many different options, can easily create tooltips with complex content (including images which will allow us to include cover images in the tooltip itself), custom backgrounds, opacity levels, easy positioning, mouse tracking, etc.

- Wil Linssen has made some simple modifications to the JQuery UI library that allows for more advanced sortable lists and for automatic updating of stored list order on list item reordering (<http://www.wil-linssen.com/jquery-sortable-lists-with-drag-drop-handle/>). This plugin helped illuminate how to capture the order of DOM elements upon reordering.

- Using \$.get (JQuery.get) to load JSON files, with type specified as text, then using the json2.js library from json.org (<http://json.org/json2.js>) to parse the resulting non-json text. Getting the JSON object as a string ensures that no automatic processing of the JSON file is done by JQuery. Other JQuery AJAX methods (JQuery.getJSON() always, JQuery.ajax() if type is 'json', JQuery.get() if type is 'json', JQuery.post() if type is 'JSON') automatically do an eval and parse the JSON without checking the file contents - this can be a security risk, as malicious scripts could be embeded in the JSON file (see <http://json.org/json2.js> for more details). This technique is a secure way of processing JSON objects, as the JSON.parse() method checks to make sure the JSON file only contains valid JSON data. (See <http://funkatron.com/site/comments/safely-parsing-json-in-javascript/> for details.)

- Using the JQuery validation plugin (<http://bassistance.de/jquery-plugins/jquery-plugin-validation/>) for form validation in the new shelf dialog - will also use it for user account creation/login/update/lost password.

Creating the program

This walkthrough assumes that you have working versions of the above programs running on your server.

Creating /configuring the program

Rails *virtualshelves* -d mysql

Creates the program and necessary file structure

Edit database.yml to include the database passwords

Otherwise your rails app cannot communicate with the database and nothing works

Scaffold Commands:

Run:

ruby script/generate scaffold author author_raw:string author_olid:string author_firstname:string, author_lastname:string

ruby script/generate scaffold book book_title:string book_prefix:string book_subtitle:string book_olid:string by_statement:text publish_date:string publish_year:integer publishers:text publish_place:text raw_book_dimensions:string fulltext:boolean num_of_pages:integer ISBN10:string ISBN13:string LCCN:string LC:string dewey:string subject:text

ruby script/generate scaffold spine book_id:integer name:string data:binary book_height:float book_width:float book_depth:float

ruby script/generate scaffold shelfcontent shelf_id:integer book_id:integer position:integer

```
ruby script/generate scaffold shelf shelf_name:string privacy:integer group_id:integer  
shelf_description:text
```

```
ruby script/generate scaffold group group_name:string privacy:integer
```

```
ruby script/generate scaffold user username:string password:string email:string first_name:string  
last_name:string
```

```
Ruby script/generate controller home
```

The above commands create the models and controllers that make up most of the content of the backend. For explanations of why the above data model was chosen, see the section on data models below.

Migrations:

The above scaffolds create migration files for each. However, we also require two custom migrations for join tables. For more explanation, see the data model section

Run:

```
ruby script/generate migration create_authors_books
```

Then alter the migration file to read as follows:

```
class CreateAuthorsBooks < ActiveRecord::Migration  
  def self.up  
    create_table :authors_books, :id => false do |t|  
      t.integer :author_id  
      t.integer :book_id  
    end  
  end  
  
  def self.down  
    drop_table :authors_books  
  end  
end
```

end

Run:

```
ruby script/generate migration create_groups_users
```

Then alter the migration file to read as follows:

```
class CreateGroupsUsers < ActiveRecord::Migration  
  def self.up  
    create_table :groups_users, :id => false do |t|  
      t.integer :group_id  
      t.integer :user_id  
    end  
  end  
  
  def self.down
```

```

drop_table :groups_users
end
end

end

```

--join table naming conventions are the two tables being joined in alphabetic order separated with underscore. These do not need to be fully modeled objects. Rather they can just be added to the database via migrations

--Note that the shelfcontent table is more than a join table. Rather it exists as a full object. This is because join tables cannot have more attributes, and we want to be able to include position. Because of this, the “through” methods must be added to the model.

Models:

Models contain the relationships between objects as well as methods associated with the creation those objects. The text of the following models should be modified to read:

Author

```

class Author < ActiveRecord::Base
  has_and_belongs_to_many :books

  def self.fullauthorcreate(parsed_json_string, current_book)

    current_book.authors.delete_all
    if parsed_json_string['authors']
      parsed_json_string['authors'].each do |author_pair|
        existing_author = Author.find(:first, :conditions => { :author_olid => author_pair['key'] })
        if existing_author
          then current_book.authors << existing_author
          current_book.save
        else
          current_author = Author.new(:author_olid => author_pair['key'], :author_raw =>
          author_pair['name'])

          if current_author.author_raw =~ /,/
            current_author.author_lastname = $~.pre_match
            current_author.author_firstname = $~.post_match
          else
            raw = current_author.author_raw.split(/\s/)
            if raw[1]
              then
                current_author.author_firstname = raw[0]
                current_author.author_lastname = Author.lastname(raw)
            else current_author.author_lastname = raw[0]
              end
            end
          end
          current_author.save
        end
      end
    end
  end
end

```

```

    current_book.authors << current_author
    current_book.save
  end
end
end
end

#Author.find(:all, :conditions => {:author_id => au.author_id})

#A couple of name edgecases. These should be added to be expanded.
def self.lastname(ns)
  ln = ns[-1]
  if ns[-1] =~ /(\A(jr|sr)\.*$)|(\\AI+$|\\AI+V$)/
    ns.delete[-1]
    ln = ns[-1]
  end
  if ns[-3] == ("van" or "von" or "de")
    then
      ln = ns[-3] << " " << ns[-2] << " " << ns[-1]
    else
      if ns[-2] == ("de" or "van" or "von")
        ln = ns[-2] << " " << ns[-1]
      end
    end
  end
  ln
end

end

```

Book

```

class Book < ActiveRecord::Base

  has_and_belongs_to_many :authors
  has_many :shelfcontents
  has_many :shelves, :through => :shelfcontents
  has_one :spine

  def self.trunc(string)
    if string.length > 255
      nstring = string.slice(0..252) << "..."
      nstring
    else
      string
    end
  end

  def self.fullbookcreate (parsed_json_string)

```

```

current_book = Book.new(:book_title => parsed_json_string['title'] , :book_olid =>
parsed_json_string['key'])
  current_book.raw_book_dimensions = parsed_json_string['physical_dimensions']
  current_book.publish_date = parsed_json_string['publish_date']
  if current_book.publish_date
    if current_book.publish_date =~ (/d{4}/)
      current_book.publish_year = $~.to_s
    end
  end
  current_book.by_statement = parsed_json_string['by_statement']

  current_book.num_of_pages = parsed_json_string['number_of_pages']

  if parsed_json_string['ocaid']
    current_book.fulltext = true
  end

  cdlbreak = ","

# Make sure that the comma remover is pulling the last one

# Saves all publishers as a comma delineated string
  if parsed_json_string['publishers']
    parsed_json_string['publishers'].each do |publisher|
      current_book.publishers = current_book.publishers.to_s << publisher << cdlbreak
    end
    current_book.publishers.sub!(/#\{cdlbreak\}\z/ , "")
  #   current_book.publishers = trunc(current_book.publishers)
  end

# Saves all publish places as a comma delineated string
  if parsed_json_string['publish_places']
    parsed_json_string['publish_places'].each do |places|
      current_book.publish_place = current_book.publish_place.to_s << places << cdlbreak
    end
    current_book.publish_place.sub!(/#\{cdlbreak\}\z/ , "")
  #   current_book.publish_place = trunc(current_book.publish_place)
  end

# Saves all subjects as a comma delineated string
  if parsed_json_string['subjects']
    parsed_json_string['subjects'].each do |sub|
      current_book.subject = current_book.subject.to_s << sub << cdlbreak
    end
    current_book.subject.sub!(/#\{cdlbreak\}\z/ , "")
  #   current_book.subject = trunc(current_book.subject)
  end

# Saves all ISBN10 as a comma delineated string
  if parsed_json_string['ISBN10']
    parsed_json_string['ISBN10'].each do |is10|

```

```

    current_book.ISBN10 = current_book.ISBN10.to_s << is10 << cdlbreak
end
    current_book.ISBN10.sub!(#{cdlbreak}\z/ , "")
#   current_book.ISBN10 = trunc(current_book.ISBN10)
end

# Saves all ISBN13 as a comma delineated string
if parsed_json_string['ISBN13']
parsed_json_string['ISBN13'].each do |is13|
    current_book.ISBN13 = current_book.ISBN13.to_s << is13 << cdlbreak
end
    current_book.ISBN13.sub!(#{cdlbreak}\z/ , "")
#   current_book.ISBN13 = trunc(current_book.ISBN13)
end

# Saves all LCCN as a comma delineated string
if parsed_json_string['LCCN']
parsed_json_string['LCCN'].each do |lccn|
    current_book.LCCN = current_book.LCCN.to_s << lccn << cdlbreak
end
    current_book.LCCN.sub!(#{cdlbreak}\z/ , "")
#   current_book.LCCN = trunc(current_book.LCCN)
end

# Saves all LC as a comma delineated string
if parsed_json_string['LC_classifications']
parsed_json_string['LC_classifications'].each do |lc|
    current_book.LC = current_book.LC.to_s << lc << cdlbreak
end
    current_book.LC.sub!(#{cdlbreak}\z/ , "")
#   current_book.LC = trunc(current_book.LC)
end

# Saves all Dewey as a comma delineated string
if parsed_json_string['dewey']
parsed_json_string['dewey'].each do |dew|
    current_book.dewey = current_book.dewey.to_s << dew << cdlbreak
end
    current_book.dewey.sub!(#{cdlbreak}\z/ , "")
#   current_book.dewey = trunc(current_book.dewey)
end
current_book

end

end

```

Group

```

class Group < ActiveRecord::Base
  has_and_belongs_to_many :user

```

```
end
```

Shelf

```
class Shelf < ActiveRecord::Base
  belongs_to :group
  has_many :books, :through => :shelfcontents
end
```

Shelfcontent

```
belongs_to :shelf #foreign key - shelf_id
belongs_to :book #foreign key - book_id
```

Spine

```
class Spine < ActiveRecord::Base
  belongs_to :books
```

```
require 'RMagick'
include Magick
def self.randgen(dimension)
  sign = rand(1)
  dim = 0
  if dimension == "height"
    coe = rand(25)
    if sign == 1
      dim = 90 + coe
    else
      dim = 90 - coe
    end
  end
  if dimension == "depth"
    coe = rand(10)
    if sign == 1
      dim = 40 + coe
    else
      dim = 40 - coe
    end
  end
  dim
end
```

```
#Spine Image Creation
def self.spinecreate(current_book)
  finishedspine = Spine.new
```

#The values are set to average dimensions. If no data is found to change them, they will remain.

```
  finishedspine.book_height = Spine.randgen("height")
  finishedspine.book_depth = Spine.randgen("depth")
```

```

if current_book.raw_book_dimensions or current_book.num_of_pages
#Looks if there is a dimensions string
  if current_book.raw_book_dimensions
    #Pull height from dimensions and set to tmp
      tmp_book_height = (current_book.raw_book_dimensions.slice!(/\d+.*(\d+)/).to_f * 9)
    #Pull width from dimensions and set to tmp
      finishedspine.book_width = current_book.raw_book_dimensions.slice!(/\d+.*(\d+)/).to_f
    #Pull depth from dimensions and set to tmp if it is there
      if (current_book.raw_book_dimensions =~ /(d+)/)
        then tmp_book_depth = (current_book.raw_book_dimensions.slice!(/\d+.*(\d+)/).to_f * 20)
      end
    end

#If there is not a dimensions string, looks for page number
  if current_book.num_of_pages and tmp_book_depth == nil
    #If there is a page numeber use it to set depth
      then tmp_book_depth = (current_book.num_of_pages * 0.075)
    #If there is page num, see if there was a height not assigned earlier
      if tmp_book_height == nil
        #If not, make a guess on the assigned height based on the depth
          tmp_book_height = (tmp_book_depth * 3)
        end
      end

# Tests to make sure that the calculated heights are not outside the display boundries
if tmp_book_height > 115
  then finishedspine.book_height = 115
else
  if tmp_book_height < 65
    then finishedspine.book_height = 65
  else finishedspine.book_height = tmp_book_height
  end
end
if tmp_book_depth > 60
  then finishedspine.book_depth = 60
else
  if tmp_book_depth < 20
    then finishedspine.book_depth = 20
  else finishedspine.book_depth = tmp_book_depth
  end
end
end

```

rawtit = "caption:" << current_book.book_title

```

canvas = Image.new(finishedspine.book_height, finishedspine.book_depth){self.background_color =
'transparent' }
rect = Draw.new
# rect.fill('rgb(' << Spine.colorpicker(current_book.LC, current_book.subject) << ')')
rect.fill("blue")
rect.fill_opacity('60%')
rect.rectangle(0, 0, finishedspine.book_height, finishedspine.book_depth)
rect.draw(canvas)
canvas.write("canvas.gif")

titarray = Image.read(rawtit) do
self.size = "" << finishedspine.book_height.to_s << "x" << finishedspine.book_depth.to_s << ""
self.pointsize = 9
self.font = "Tahoma"
self.background_color = 'transparent'
end

tit = titarray[0]
tit.write("testimage.gif")

spineimage = canvas.composite(tit, NorthWestGravity, 5, 5, OverCompositeOp)
spineimage.rotate!(90)
filename = "/images/spines/" << current_book.book_olid.sub(/\b/, ".") << ".gif"
spineimage.write(filename)

finishedspine.update_attributes(:book_id => current_book.id, :name => filename, :data =>
spineimage)
finishedspine.save
finishedspine
end

def self.colorpicker (lc, subjects)
# These are the Amazon top-Level subject headings. Some of the most broad subject areas have been
moved to the end. Other than that, so ordering has been done.
if lc
then
  lcssubject_array =
["A","B","C","D","E","F","G","H","J","K","L","M","N","P","Q","R","S","T","U","V","Z"]
  lccolor_array = [199, 209, 245,"208, 250, 197","150, 171, 171","150, 171, 171","150, 171,
171","150, 171, 171","149, 208, 214","183, 127, 94","145, 166, 245","84, 164, 144","183, 127, 94","162,
174, 115","162, 174, 115","255, 250, 147","92, 94, 183","229, 127, 43","144, 249, 133","200, 202,
202","92, 94, 183","92, 94, 183","199, 209, 245"]
  sub = lc[/^w/]
  color = lccolor_array[lcssubject_array.index(sub)]
else
  if subjects
    then
      sub =
subjects[/Art|Photography|Biograph|Memoirs|Business|Investing|Children|Christian|Comic|Graphic|Comp

```

```

uter|Internet|Cooking|Food|Wine|Crafts|Hobbies|Entertainment|Gay|Lesbian|Health|Mind|Body|History|Home|Garden|Mystery|Thriller|Outdoor|Nature|Parenting|Families|Politics|Professional|Technical|Puzzles|Games|Reference|Religion|Spirituality|Romance|Science Fiction|Science|Fantasy|Self-Help|Sports|Teens|Travel|Literature|Fiction/]

  if sub
    amasubject_array =
      ["Art", "Photography", "Biograph", "Memoirs", "Business", "Investing", "Children", "Christian", "Comic", "Graphic", "Computer", "Internet", "Cooking", "Food", "Wine", "Crafts", "Hobbies", "Entertainment", "Gay", "Lesbian", "Health", "Mind", "Body", "History", "Home", "Garden", "Mystery", "Thriller", "Outdoor", "Nature", "Parenting", "Families", "Politics", "Professional", "Technical", "Puzzles", "Games", "Reference", "Religion", "Spirituality", "Romance", "Science Fiction", "Science", "Fantasy", "Self-Help", "Sports", "Teens", "Travel", "Literature", "Fiction"]

    amacolor_array = ["162, 174, 115", "162, 174, 115", "162, 107, 153", "162, 107, 153", "135, 171, 112", "135, 171, 112", "134, 110, 161", "208, 250, 197", "91, 123, 164", "91, 123, 164", "200, 202, 202", "200, 202, 202", "173, 92, 108", "173, 92, 108", "173, 92, 108", "0, 160, 198", "0, 160, 198", "0, 160, 198", "177, 164, 116", "177, 164, 116", "229, 127, 43", "229, 127, 43", "229, 127, 43", "150, 171, 171", "119, 198, 81", "119, 198, 81", "255, 127, 0", "255, 127, 0", "144, 249, 133", "144, 249, 133", "235, 124, 124", "235, 124, 124", "145, 166, 245", "200, 202, 202", "200, 202, 202", "0, 160, 198", "0, 160, 198", "199, 209, 245", "208, 250, 197", "208, 250, 197", "233, 51, 121", "251, 134, 179", "92, 94, 183", "251, 134, 179", "2, 160, 113", "0, 160, 198", "171, 16, 210", "247, 186, 101", "255, 250, 147", "255, 250, 147"]

    color = amacolor_array[(amasubject_array.index(sub))]
  else color = "241, 220, 152"
  end
else
  color = "241, 220, 152"
end
end
color
end
end

```

User

has_and_belongs_to_many :groups

Run the Migrations

Rake db:create:all

This creates the tables. This only has to be run the first time a migration is run.

Rake db:migrate

This populates the tables

Updating Controllers

Controllers tell rails how to direct page requests. Your controllers should be updated to read as follows:

Books

```
class BooksController < ApplicationController
  # GET /books
  # GET /books.xml
  def index

    @books = Book.find(:all)

    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @books }
    end
  end

  # GET /books/1
  # GET /books/1.xml
  def show
    @book = Book.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.xml { render :xml => @book }
    end
  end

  # GET /books/new
  # GET /books/new.xml
  def new
    @book = Book.new

    respond_to do |format|
      format.html # new.html.erb
      format.xml { render :xml => @book }
    end
  end

  # GET /books/1/edit
  def edit
    @book = Book.find(params[:id])
  end

  # POST /books
  # POST /books.xml
  def create

    @book = Book.save(params[:book])
  end
end
```

```

respond_to do |format|
  if @book.save
    flash[:notice] = 'Book was successfully created.'
    format.html { redirect_to(@book) }
    format.xml { render :xml => @book, :status => :created, :location => @book }
  else
    format.html { render :action => "new" }
    format.xml { render :xml => @book.errors, :status => :unprocessable_entity }
  end
end
end

# PUT /books/1
# PUT /books/1.xml
def update
  @book = Book.find(params[:id])

  respond_to do |format|
    if @book.update_attributes(params[:book])
      flash[:notice] = 'Book was successfully updated.'
      format.html { redirect_to(@book) }
      format.xml { head :ok }
    else
      format.html { render :action => "edit" }
      format.xml { render :xml => @book.errors, :status => :unprocessable_entity }
    end
  end
end
end

# DELETE /books/1
# DELETE /books/1.xml
def destroy
  @book = Book.find(params[:id])
  @book.destroy

  respond_to do |format|
    format.html { redirect_to(books_url) }
    format.xml { head :ok }
  end
end
end

```

Shelves

```

class ShelvesController < ApplicationController
  require 'net/http'
  require 'uri'

```

```

require 'json'

# GET /shelves
# GET /shelves.xml
def index
  @shelves = Shelf.find(:all)

  if (params[:command]) == "user"
    user = User.find((params[:user_id]))
    groups = user.groups
    @usershelves = Array.new
    groups.each do |group|
      if Shelf.find(:all, :conditions => {:_id => group._id})
        @usershelves << Shelf.find(:all, :conditions => {:_id => group._id})
      end
    end
  end

  respond_to do |format|
    format.html # index.html.erb
    format.js # index.js.erb
    format.xml { render :xml => @shelves }
  end
end

# GET /shelves/1
# GET /shelves/1.xml
# GET /shelves/1.atom
def show
  @shelf = Shelf.find(params[:id])
  @books = @shelf.books

  respond_to do |format|
    format.html # show.html.erb
    format.js # show.js.erb
    format.xml { render :xml => @shelf}
    format.atom #show.atom.builder
  end
end

# GET /shelves/new
# GET /shelves/new.xml
def new

  api_query = URI.escape('/api/search?q={"query":"' + params[:search_string] + '"}&rows=25')
  @olresult = Net::HTTP.get 'openlibrary.org', api_query
  @olresultparse = JSON.parse(@olresult)

```

```

new_shelf = Shelf.new(:shelf_name => "search:" << params[:search_string], :privacy => 3, :group_id => 12)
new_shelf.save
@olresultparse['result'].each do |bookid|
  book_api_path = '/query.json?query={"type":"/type.edition","authors":{"name":null,"*":null,"key":""}<< bookid << ""}'
  book_json_data = Net::HTTP.get 'openlibrary.org', book_api_path
  book_data_parsed = JSON.parse(book_json_data)[0]
  finished_book = Book.fullbookcreate(book_data_parsed)
  existing_book = Book.find(:first, :conditions => { :book_olid => bookid })
  if existing_book
    then
      existing_book.update_attributes(finished_book.attributes)
      if existing_book.spine == nil
        existing_book.spine = Spine.spinecreate(existing_book)
      end
      new_shelf.books << existing_book
      new_shelf.save
    else
      finished_book.save
      Author.fullauthorcreate(book_data_parsed, finished_book)
      finished_book.spine = Spine.spinecreate(finished_book)
      finished_book.save
      new_shelf.books << finished_book
      new_shelf.save
    end
  end

  @dashelf = new_shelf.to_json
  @dabooks = new_shelf.books.to_json
  @shelf = new_shelf

  respond_to do |format|
    format.html { redirect_to(@shelf) }
    format.xml { render :xml => @shelf }
  end
end

# GET /shelves/1/edit
def edit
  @shelf = Shelf.find(params[:id])
end

# POST /shelves
# POST /shelves.xml
def create

```

```

@shelf = Shelf.new(:shelf_name => (params[:shelf_name]), :group_id => (params[:group_id]),
:privacy => (params[:privacy]), :shelf_description => (params[:description]))

respond_to do |format|
  if @shelf.save
    @alert = 'Shelf was successfully created.'
    format.html { redirect_to(@shelf) }
    format.xml { render :xml => @shelf, :status => :created, :location => @shelf }
    format.js
  else
    @alert = 'Shelf not created'
    format.html
    format.xml { render :xml => @shelf.errors, :status => :unprocessable_entity }
    format.js
  end
end
end

# PUT /shelves/1
# PUT /shelves/1.xml
def update
  @shelf = Shelf.find(params[:id])
  bookstring = JSON.parse(params[:book_ids])

  if (params[:command]) == "add"
    bookstring.each do |bookid|
      bo = Book.find(:first, :conditions => {:book_olid => bookid})
      if @shelf.books.include?(bo)
        else
          @shelf.books << bo
        end
      end
    end

  end

  if (params[:command]) == "remove"
    bookstring.each do |bookid|
      @shelf.books.delete(Book.find(:first, :conditions => {:book_olid => bookid}))
      @shelf.save
    end
  end

  if (params[:command]) == "update"
    @shelf.books.delete_all
    bookstring.each do |bookid|
      @shelf.books << Book.find(:first, :conditions => {:book_olid => bookid})
    end
    @shelf.save
  end
end

```

```

    end
  end

  if (params[:command]) == "shelf_info_update"
    @shelf.update_attributes(:shelf_name => (params[:shelf_name]), :group_id =>
  (params[:shelf_name]), :privacy => (params[:privacy]), :shelf_description => (params[:description]))
    end

  respond_to do |format|
    format.html { redirect_to(@shelf) }
    format.xml { head :ok }
    format.js
  end
end

# DELETE /shelves/1
# DELETE /shelves/1.xml
def destroy
  @shelf = Shelf.find(params[:id])
  @shelf.destroy

  respond_to do |format|
    format.html { redirect_to(shelves_url) }
    format.xml { head :ok }
  end
end
end

```

Users

```

class UsersController < ApplicationController
  require 'RMagick'
  include Magick

  # GET /users
  # GET /users.js
  def index
    @users = User.find(:all)

    respond_to do |format|
      format.html # index.html.erb
      format.js
    end
  end

  # GET /users/1
  # GET /users/1.xml
  def show

```

```

@user = User.find(params[:id])
@groups = @user.groups
@shelves = []
@groups.each do |gr|
  gr.shelves.each do |sh|
    @shelves << sh
  end
end

respond_to do |format|
  format.html # show.html.erb
  format.xml { render :xml => @user }
end
end

# GET /users/new
# GET /users/new.xml
def new
  @user = User.new

  respond_to do |format|
    format.html # new.html.erb
    format.xml { render :xml => @user }
  end
end

# GET /users/1/edit
def edit
  @user = User.find(params[:id])
end

# POST /users
# POST /users.xml
def create
  @user = User.new(params[:user])
  @group = Group.create(:group_name => (params[:user][:username]))
  @user.groups << @group

  respond_to do |format|
    if @user.save
      flash[:notice] = 'User was successfully created.'
      format.html { redirect_to(@user) }
      format.xml { render :xml => @user, :status => :created, :location => @user }
    else
      format.html { render :action => "new" }
      format.xml { render :xml => @user.errors, :status => :unprocessable_entity }
    end
  end
end

```

```

end

# PUT /users/1
# PUT /users/1.xml
def update
  @user = User.find(params[:id])

  respond_to do |format|
    if @user.update_attributes(params[:user])
      flash[:notice] = 'User was successfully updated.'
      format.html { redirect_to(@user) }
      format.xml { head :ok }
    else
      format.html { render :action => "edit" }
      format.xml { render :xml => @user.errors, :status => :unprocessable_entity }
    end
  end
end

# DELETE /users/1
# DELETE /users/1.xml
def destroy
  @user = User.find(params[:id])
  @user.destroy

  respond_to do |format|
    format.html { redirect_to(users_url) }
    format.xml { head :ok }
  end
end

```

Updating the layout

From here it is just a matter of creating and updating any view information you need. All of these files, along with the XHTML and Javascript code can be found in the attached .zip package of the program. This version will be subject to changes before the final demonstration.