2011

# A Path to Providing User-Centered Subject Guides

Tammy Stitz, *University of Akron Main Campus*
Shari Laster, *University of Akron Main Campus*
Frank J Bove, *University of Akron Main Campus*
Casey Wise, *University of Akron Main Campus*

# A Path to Providing User-Centered Subject Guides

TAMMY STITZ, SHARI LASTER, and FRANK J. BOVE

University Libraries, University of Akron, Akron, OH, USA

CASEY WISE

Information Technology Services, University of Akron, Akron, OH, USA

**Abstract**

The experience at the University of Akron illustrates a transitional path from static Web sites to dynamic, user-centered Web design.  Starting with Web-based subject resource guides that were difficult to maintain and navigate, University Libraries migrated to a content management system that solved existing maintenance issues and created the opportunity for interactive resource lists.  Subsequent usability testing found that users needed librarian-designed subject guides to understand and access resources.  Consequently, University Libraries redesigned its Web site functionality.  In the course of this article, it is shown that ease of maintenance and ease of use need not be mutually exclusive.

## 1.  Introduction

Subject and course guides have a long history at University Libraries (UL) at the University of Akron (UA).  With the rapidly increasing availability of electronic resources for every discipline, Web-based subject guides lead users to critical resources that support their research or help them complete course-specific assignments.  These guides also provide support information that leads to effective use of the available resources, and contact information for subject- or course-specific librarian assistance.

The history of our Web-based subject and course guides illustrates a trend toward easier maintenance for librarians.  In developing and populating a system that ensured easier maintenance, however, we initially lost functionality that served some of our users' needs.  Following the

implementation of a campus-wide content management system (CMS), our subject and course guides were transformed into long lists of resources presented in alphabetical order without sufficient guiding text or groupings.  While undertaking usability testing to discover opportunities for improving the Library Web site, we identified the functionality needed to transform a list of resources into a subject guide.  The usability results prompted a redesign that was implemented by UA Information Technology Services (ITS).  Our final design shows that usability testing combined with technical expertise can deliver user-centric subject and course guides that are easy to maintain.

## 2.   From Static Pathfinders to FrontPage-CMS-Generated Subject Guides

In the early days of the library Web site, the library developed HTML versions of standard print pathfinder handouts for its Web pages.  These pages included a series of static hyperlinks to library databases and other electronic resources along with extensive textual narrative advising users how to access print and physical materials owned by the library.  As with many early library Web sites, a site that started as a few static pages grew into thousands of files consisting of hundreds of megabytes.  As in comparable institutions and departments on campus, library personnel have varying levels of HTML expertise, so the responsibility of creating and maintaining Web pages was left to interested library personnel.  The volume and complexity of these files can make maintenance very time-consuming (Kmetz & Bailey, 2006; Goans, Leach, & Vogel, 2006).  Policy often dictates that Web authors use templates; however, the authors are free to edit the HTML Web pages based on their preferences and skill levels (Kmetz & Bailey, 2006).

It is commonly known that a CMS can alleviate these problems (Kmetz & Bailey, 2006; Goans *et al.*, 2006).  UL built its own Microsoft FrontPage CMS in 2003-2004 to allow librarians to build and edit Web pages for specific subjects and courses or other subject subdivisions.  As happens at many libraries, the first step in migrating to a new system was to examine the structure of the existing system in order to find patterns.  Standardized categories such as Article Databases, and Journals and Newspapers, were

developed by examining the existing subject and course guides.  To accommodate differences among

subjects, librarians were also able to create their own categories.  In the UL FrontPage CMS,

introductory text accompanied the standard categories to promote uniformity and provide instruction.

Library subscription resources could be added to a page using a dropdown list, providing a consistent

description for the resource and facilitating maintenance of the URL through a single process.

Additionally, librarians could include custom resources or instructions on the subject page using a Web-

based form that allowed the inclusion of embedded HTML tutorials.  By default, each page included

contact information for the subject librarian and each subject guide contained a hyperlink to the

collection development policy.  The categories, librarian-added resources, and custom instruction

displayed in the order that the librarian specified, an important concept in the literature (Dupuis, Ryan,

& Steeves, 2004, p. 275; Greene, 2008, pp. 215-216).  Users will often try links that appear earlier in a

category first.  For example, depending on the topic, an electrical engineering student might need to

search *IEEE Xplore* first because its scope is limited to electrical engineering resources.  A novice might

select *Engineering Village* interfacing with the *Compendex* database, which covers engineering from an

interdisciplinary perspective, first when the options are displayed in alphabetical order.  Listing the best

options first in the Article Database section can assist users in making a selection.  The resulting page

presented a long list of hyperlinks with integrated instructional text and anchor hyperlinks to navigate to

different sections of the page.

     This was a great leap forward for both the librarians and the users; however, the UL FrontPage

CMS did not address some important issues.  While the CMS eliminated duplicate content for

subscription resources, custom content added by the subject librarians could not be reused, resulting in

hyperlinks that were correct on one page and incorrect on another.  This system also resulted in an

unmanageably large database that negatively impacted performance for users.  While the guides

automatically added anchor hyperlinks to navigate the pages, users disliked scrolling through the content.

Many users also complained that it could be difficult to locate the subject and course guides on the Web site. A Web page listing all the subject guides could be accessed from the library home page; however, the "Resources by Subject" hyperlink was one hyperlink among thirteen under a heading of "Guides and Links to Library Resources" and was easily overlooked by the user. The hyperlink to the Web page that listed all the course guides was even more elusive. It was located at the top of the Web page that listed all the subject guides, making it even easier to overlook. The location of subject guides on the Web site is a common theme libraries need to address, because Web resources require effective location in order to promote their use (Morris & Del Bosque, 2010, p. 181).

After several years using the FrontPage CMS, UL realized that its entire Web presence needed to be grounded on a single platform with a CMS that would allow all library personnel to contribute content to the Web site. In 2006, the library began to explore CMS options, just as a similar process was underway at the university level. The UA Webmaster invited library personnel to participate in planning for a university-wide solution.

A survey by Connell (2008) indicates that UL was not alone in the decision to move to a CMS. Connell surveyed a sample of Web developers from academic libraries in the United States. Among the results from the 94 respondents, she found that 25.5% were using CMS software and 26.6% were using a database-driven system. In addition, she found that larger institutions tend to have more complicated needs regarding Web presence, which could signify a need for a database-driven system (Connell, 2008, p. 126). Many libraries have created database-driven Web pages to deliver their Web guides (Goans *et al.*, 2006; Greene, 2008; Whang, 2005). At Kent State University, the entire Web site for University Libraries was developed in house using MySQL and PHP (Wiggins, Remley, & Klingler, 2006). Some libraries decide to write their own Web applications due to a need for specific features not available in

packaged content management systems, the high cost of commercial content management systems, and the ability to communicate with existing library systems.  Generally, software tools are chosen when they have been successfully employed on another project and the expertise is already available within the institution.  Libraries have developed database-driven Web sites using databases including MySQL, SQL Server, and PostgreSQL, and using languages, such as PHP and ASP to interface with them.  Connell (2008, p. 126) postulated that smaller libraries generally do not have the staff to support a home-grown database driven solution.

Fulton (2010) surveyed ARL member and Illinois institutions to determine which content management systems were used most by academic libraries.  Albeit within a small result set, Plone and Drupal were found to be used most frequently.  Existing literature supports this result, particularly with respect to Drupal (Fennell, 2007; Garza, 2008; Wiersma, 2009).  A list of libraries that use each CMS can be found at the following Web sites:

- Drupal – http://groups.drupal.org/libraries/libraries

- Plone – http://plone.org/support/sites/sites_listing?industries%3Alist=Libraries

In the Fulton (2010) survey, one of the commercial content management systems used by an Illinois institution at the time was Ektron CMS400.net.  The Camden-Carroll Library at Morehead State University was part of a campus-wide implementation of Ektron CMS400.net (Kmetz & Bailey, 2006).

Ultimately, UA selected a firm called dotMarketing to customize dotCMS Enterprise to provide additional functionality to the UL Web site, and in January 2009, the Web site was launched on the production server.

## 3. DotCMS Enterprise

DotCMS is an enterprise Java-based, open source CMS using an Apache Tomcat application server and a MSSQL, PostgreSQL, MySQL or Oracle database.  Although dotCMS is compatible with multiple databases, the PostgreSQL database yields the superior implementation (Fienen, 2010, p. 19).

DotCMS utilizes the Velocity Templating Language and the Lucene™ Search Engine.  DotCMS has

considerable system requirements, which can result in greater startup cost when purchasing hardware.

Since dotCMS is cross-platform, it is possible to save money with server software.  Another commercial

CMS provider, Ektron, is a Microsoft partner, a fact that limits Ektron CMS400.net to being compatible

with only Microsoft products, which can also cause additional software costs.  DotCMS is bundled with

and employs several other open source tools, such as Apache Struts and Hibernate.  Officially, extensive

knowledge of these tools is not required for implementation; however, in the event they are not

functioning properly the administrator might need to gain this knowledge.  DotCMS has a community

edition; however, the enterprise version has more extensive capabilities.  In addition to the price of the

enterprise software, there is a licensing fee, and some of the code in the enterprise version is

proprietary.  In contrast with Drupal and Plone, DotCMS does not have a great deal of documentation

and is not widely used in libraries.  DotCMS training is available, but it is costly.

Notable strengths of dotCMS include multi-site capabilities, where one instance of dotCMS can

support multiple Web sites, and granularity in assigning content and administrative permissions.  Other

standard features of dotCMS include a structured content repository, workflow, content versioning and

rollback, streaming video, and Web form-based content entry.  Web 2.0 features, such as user ratings,

mediated comment entry, and tag clouds, can be included with the use of dotCMS macros that contain,

in some cases, hundreds of lines of Velocity code.  To develop a Web site using dotCMS, knowledge of

Velocity is a must.  With the use of dotCMS macros and Velocity, functionality  seen in Drupal or Plone

can be replicated (Fienen, 2010, p. 7).  It is not uncommon for some programming to be required when

using a CMS.  It is possible that you would need to write Python code when using Plone or PHP when

using Drupal.

The enterprise site search for dotCMS is Lucene™, which is a solid and fast search interface;

however, not all search schemes can be achieved with this tool.  DotCMS can be configured to use the

Google Search Appliance (GSA).  Drupal has an integrated search, but it is more suitable for small Web

sites.  For larger Web sites, a module can be used to implement the GSA with Drupal as well (Redding,

2010, p. 178).  Plone uses the Zope built-in search engine, ZCatalog, which includes several indexes, by

default (Delmonte & McKay, 2009, pp. 147-158).  Using the default search, a user can search with a

single text box or perform an advanced search.  The LiveSearch feature in Plone uses AJAX to deliver a

list of results while the user is typing (Delmonte & McKay, 2009, p. 79).  Stemming is not available, but

the user can use a truncation character to achieve this effect.  There are other value added features,

such as suggested spelling, that are available using the GSA and are not available when using ZCatalog.

There is a package that can be added to Zope to enable Plone to use the GSA for the single search box

and LiveSearch.  ZCatalog and GSA index differently; therefore, all indexes cannot be replaced by the

GSA.  Plone's advanced search template cannot be used by the GSA, but the GSA advanced search can

be used instead.  A caveat about using ZCatalog is that indexing is extremely resource intensive and can

take a great deal of time (Delmonte & McKay, 2009, p. 148).

Each Web page in dotCMS is based on a template to standardize the look and feel of the Web

pages typically consisting of navigation and aesthetics.  Containers containing advanced functionality

and structured content are embedded in the templates.  Structures consist of three types: widgets,

content, and form data, the last of which is available in version 1.9 and later (Fienen, 2010, p. 47).

Widgets provide a simpler method of embedding complex functions.  They are reusable and can be

included in any container.  When a widget is selected for a container, the developer only needs to enter

parameters instead of many lines of code.  DotCMS packages several structures with the initial install,

including events, web page content, and news.  Structures are created much like database tables, with

many field types available such as text, HTML form elements, keyword tags, WYSIWYG, images, or any

other type of Web-based media.  DotCMS also allows the creation of categories that can be used as

fields.  Categories, which are used to facilitate the collocation of resources, can consist of a single layer

or a multi-level hierarchy.  In addition, a field can contain Velocity or JavaScript to run when a content author creates or edits content.  To enable CMS users to relate pieces of content, relationships can be established between structures in dotCMS.  Nodes in Drupal (Redding, 2010) and content types in Plone (Delmonte & McKay, 2009) are comparable to the structures in dotCMS.  In Drupal, additional nodes and field types including those that relate content can be added by installing modules or by using the Content Construction Kit (Redding, 2010; Miles & Miles, 2009).  In Plone, the ability to relate content is available with the base application (Delmonte & McKay, 2009, p. 50).  Additional content types can be added by using the Dexterity or Archetypes frameworks (Delmonte & McKay, 2009, p. 226).  With dotCMS, more field types to construct the structures are packaged with the system without downloading modules or using external programs.

All the systems discussed have a separation of content and presentation; however, this separation is more visible in the back-end of dotCMS than with content management systems that consist of a directory tree alone.  In dotCMS, there is a section of virtual folders containing Web pages, external files loaded by CMS users, and files to support functionality of the Web site (CSS, Velocity, JavaScript, etc.).  There is a completely separate section for CMS users to enter content.  Content authors access structures in this section of the CMS to enter content using a Web-based form.  The content entered is ultimately displayed dynamically on the Web pages.  In Plone, content is placed in virtual folders (Delmonte & McKay, 2009, pp. 40-43).  To produce a dynamic list consisting of content from different folders, a content type of "collection" must be selected with a field, such as category, selected as the source of collocation (Delmonte & McKay, 2009, pp. 72-76).  At this time, it appears the ability of a multi-level hierarchy of categories is not available in Plone.  In Drupal, the taxonomy core module provides the ability to produce a multi-level hierarchy of categories (Redding, 2010, pp. 157-172).

The structures configured in our dotCMS implementation are outlined here to clarify the functionality that initially replaced the subject and course guides, along with the redesigned guides that currently exist. Resources in our new system are called "Research Tools," and data is contained in the structure, UL-ResearchTools. These resources can be a subscription article database, such as *Academic Search Complete*, *ASME Digital Library*, or *Web of Science*. A Research Tool can also be a Web site that is helpful to the users and freely accessible to everyone, such as *USGS.gov*, *Senate.gov*, or *OhioLINK's Electronic Thesis and Dissertation Center (*http://etd.ohiolink.edu/*)*. Even materials available in tangible formats can be made available as a Research Tool by linking to the static URL for a record in the library's online public access catalog (OPAC). The UL-ResearchTools structure contains fields such as title, overview, link, keywords, research tool type, and subjects. The research tool type and subjects are categories within our dotCMS implementation. "Services" are pieces of content that describe any active service provided to library users. These Services include InterLibrary Loan, research consultations, and the availability of tutoring labs. The UL-Services structure contains fields such as title, overview, service type, and number of visits, where number of visits is automatically tracked by the dotCMS software. "Help" currently has three content structures with fields similar to those of UL-Services. One is a frequently asked question, or FAQ, such as: "How do I renew books and other materials?" The second is "Citation Guide," which can direct users to resources to assist in creating citations using systems such as APA or MLA. Finally, "Tutorials" may consist of HTML pages, embedded videos, downloadable PDF handouts, or hyperlinks to external online tutorials. Relationships can be established in the CMS among Research Tools, Services, and Help in order to provide users with hyperlinks to related materials on the detailed Web pages of each Research Tool. The supporting information for the resource is available in one location. Relationships are added by selecting existing Services and Help using the "relationships" tab in the Web Form for editing the UL-ResearchTools structure. Staff directory information is contained in the structure UL-Staff, which plays a role in the redesigned subject and course guides. UL-

Staff contains fields such as first name, last name, description, and standard contact information.

Services and Departments (UL-Departments) can also be related using the "relationships" tab.

In addition to collocating content by assigning categories and relationships, content can be collocated by assigning keywords tags. Because the Research Tools structure has entry fields for subject categories, research tools categories, and keyword tags, the initial design utilized each of these to collocate research tools to provide a new interpretation of the subject and course guides.

## 4. A New Interpretation of Subject and Course Guides in a CMS Environment

The titles of the subjects guides in the legacy CMS became "subjects" categories in the initial design of the new system. The legacy categories (Article Databases, Encyclopedias, etc.) became "research tools type" categories in the new system. A few new types (Musical Scores, Patents, etc.) were added, following a review of the existing Research Tools. The library's site administrator could add new research tools types or subjects as needed. The results list therefore contained all of the resources on a subject guide when the user selected the subject using "Finding Research Tools by Subject" (browse functionality).

In our initial launch, the titles of the legacy course guides were added as keyword tags to the corresponding Research Tools in the CMS. To generate the course content, the user entered the course title, such as "Electromagnetic Compatibility," using the "keyword or tag" tab. A wireframe of the results list that was intended to replace the subject and course guides in the initial design is shown in Figure 1. The user could refine the results list or change the sort order of the Research Tools to user ratings, alphabetical order, or by Research Tool type.

Driven by feedback after the launch of the new Web site, formal usability testing was performed by a consulting company, decimal152, in April and June/July of 2009 in order to make informed decisions about fixes and enhancements the site needed. Details of the usability findings that are not directly related to the subject and course guides are beyond the scope of this article; however, the main

findings pertaining to the subject and course guides are illuminating.  Essentially, we discovered that the new version of the subject and course guides did not successfully replicate the functionality of its predecessor.  There was a sense that some of the users needed additional information literacy skills, which cannot be completely addressed in Web design (University of Akron Libraries, The University of Akron, 2009, p. 1).  However, the integrated instruction librarians were able to include in the previous version of the subject guides had helped to inform the users about where to start and how to best approach each tool based on the needs of their discipline.  Our new site design sought to use interactive features to help users, but this approach was not successful, according to the usability testing (University of Akron Libraries, The University of Akron, 2009).

For example, users had difficulty locating the "Finding Research Tools by Subject" tab, because the default tab was the keyword tab (University of Akron Libraries, The University of Akron, 2009, p. 4).  When users were directed to the correct screen, they did not understand the meaning of many of the Research Tool types, and were overwhelmed by the number of subjects available as filtering options in the results screen (University of Akron Libraries, The University of Akron, 2009, p. 4).  Instead of an easy-to-use list of available subjects, users were confronted with a confusing list of terms they did not understand.

Another serious issue was the way in which results could be sorted.  In the initial design, sorting results by user ratings was intended to direct users to highly-used or preferred resources.  Because the users were not, in fact, rating Research Tools, the sort option did not display results in any meaningful order.  All participants in the usability testing indicated that librarian-recommended resources would be more effective in providing a starting point for users (University of Akron Libraries, The University of Akron, 2009, pp. 6-7,11).

Many users seek contact information for librarians when they have a question.  On the new version of the site, users experienced a great deal of difficulty locating the subject librarian contact

information (University of Akron Libraries, The University of Akron, 2009, p. 4).  Making matters worse,

the subject hyperlink on the detail screen of a Research Tool would take the user to the collection

development policy for the subject instead of the librarian contact information.  This was normally not

the information the user was seeking.

## 5.  Leading Our Users to the Subject and Course Guides

Based on the usability tests and the reports and recommendations developed by decimal152,

UA ITS redesigned the UL Web site, and relaunched it in August 2009.  As part of our new user-centric

design, we wanted navigation to the subject and course guides to be as intuitive as possible.  Therefore,

we separated the subject guides from the browse functionality and the course guides from the keyword

search functionality.  We left those functionalities intact, but developed specialized functionality for

subject and course guides intended to address issues raised in usability testing while allowing librarians

to control and customize content as they did in the legacy FrontPage system.

First, we addressed obstacles to locating the subject guides.  We provided access to the subject

and course guides along with a brief description directly from the new Research Tools landing page as

shown in Figure 2.

To make the list of subjects less daunting, we categorized them into broad groups.  In addition,

we added a course category to generate the course guides, transforming the subject category into a

three-tier hierarchical list consisting of subject clusters, subjects (usually disciplines or majors), and

courses (primarily for use in library bibliographic instruction).

Science, Engineering, and Medicine (subject cluster)

Computer Engineering (subject)

Electromagnetic Compatibility (course)

The subject guide and course guide sections each display a linked list containing the subject

hierarchy.  When a subject cluster is selected, the cluster's subjects are revealed.  For the subject guide

section, the user selects the appropriate subject and the subject guide will appear.  In the case of the

course guide, a subject is selected and the subject's courses appear.  When a course is selected, the

course guide displays.  ITS implemented the linked list using SQL and AJAX to query the database and

present the results to the user.  As of this writing, we are running dotCMS 1.7, but we plan to upgrade to

dotCMS 1.9.  In 1.9, this process will be achieved using the dotCMS API to improve performance and

elevate rewriting code in the event of database schema changes.

**6.    Eliminating Problems while Restoring Positive Functionality of the Legacy Guides**

The new design of the guides eliminated some of the complaints about the legacy subject and

course guides.  By providing content by type using the left navigation menu, some of the legacy version's

need to scroll has been eliminated.  The left navigation menu and corresponding research tools lists are

created using SQL, AJAX, and Velocity to query the database and display the results.  After upgrading, we

will no longer use SQL to query the database.  The dotCMS API, Lucene™, and Velocity will be used

solely.  All Research Tools listed on the subject or course guides have a consistent description and URL,

because each is a single piece of content that can be maintained in one place.  This is true whether the

Research Tool in question is a subscription database or a free resource, which is an improvement over

the legacy system.  Librarian contact information is located on the subject and course guides, just as in

the legacy version.  The new design surpasses the functionality of the legacy system in that the design of

the legacy subject and course guides did not include the addition of a photograph or image by the

librarian.  A photograph or image can be added to suggest individualized service (Morris & Del Bosque,

2010).  These features are shown in Figure 3.

To use the resource listed on the guide, the user selects the resource title.  When selecting the

"View More Details" hyperlink, the detailed record appears.

Content is generated dynamically for the subject and course guides with respect to the subject

category, which is also the foundation of the browse functionality.  Three data structures contain the

content displayed on the guides.  The appropriate category is selected in the CMS by the librarian for each piece of content in the Research Tools and Staff structures.

As shown in Figure 3, the Philosophy Librarian chose the category Philosophy for the *ATLA Religion Database (EBSCO Interface)* and the *Electronic Journal Center*, so these items appear on the Philosophy subject guide.  The category Philosophy has been chosen for the content "Frank Bove" in the staff structure, so his contact information appears on the subject page as well.

## 7. Flexible Default Page for the Subject and Course Guides

Some subjects and courses require specialized content.  Therefore, the subject and course guides have been redesigned to display either a list of resources designated as "librarian recommended," or custom content (Blank Slate).

**Displaying Librarian Recommended**

As seen in Figure 3, a librarian can select several Research Tools to display dynamically on the default page of the guide.  The librarian does so by selecting the librarian recommended checkbox for content in the Research Tools Structure.

*ATLA Religion Database (EBSCO Interface)*  will appear as librarian recommended on the philosophy subject guide (Figure 3) and the religion subject guide because both of these subjects were selected and the librarian recommended checkbox was selected.  In addition, this resource will appear as part of the "Librarian Recommended" list on any other guide for which the subject is selected on the Research Tool Structure, as long as the subject is using the "Librarian Recommended" method.

The "Librarian Recommended" method is not appropriate for all subjects and courses.  The order is always alphabetical, so the librarian cannot specify an order of databases to try.  In addition, the librarian cannot integrate instruction regarding these resources, beyond the customizable Research Tool descriptions that appear after clicking the "View more details" hyperlink.  There are cases when a Research Tool should be added to multiple subjects where the Research Tool should be one of the first

tried for some subjects but not others.  For example, beginning electrical engineering majors could use

*Academic Search Complete (ASC)* to find articles; however, this Research Tool would be much less

relevant to a fourth year electrical engineering student.  For other subjects, *ASC* would be the resource

used first and has been selected as "librarian recommended" accordingly.  When the checkbox is

selected for a Research Tool in the CMS, it forces the Research Tool to be displayed as a librarian

recommended resource on all selected subject and course guides that use the Librarian Recommended

style.

**Displaying Custom Content (Blank Slate)**

Custom content can be entered for any subject or course guide.  ITS used Apache Lucene™ and

Velocity to determine whether the Blank Slate Structure contains any content for the subject or course

guide.  If a piece of content exists, it is displayed; otherwise, the librarian recommended list appears.

Librarians can enter HTML directly or they can use a WYSIWYG ("What You See is What You Get") editor

to construct this content.  For example, the computer engineering subject guide contains custom HTML

content.  In order to display this content, the librarian selects the subject (computer engineering) in the

Blank Slate (UL-SubjectCourseContent) structure.

The custom HTML content appears to the users clicking on the subject or course guide instead

of a list of Research Tools.  As an example the computer engineering guide is pictured in Figure 4.  In

addition to the contact information that appears on all guides, the librarian included HTML code for a

Meebo widget to facilitate user questions, along with a section for library news.  The Frequently Asked

Question section enables users to find answers to common questions including information on which

article databases to try first, for those who prefer to do their own exploration before requesting

assistance.

Some librarians have created custom content to direct users to internal and external tutorials on using resources, or to direct users to subject-specific library services. Librarians can also direct users to custom Web pages that are tailored to fit a department's information needs.

## 8. Conclusion

Library users often rely on the guidance that librarians provide to navigate the library's resources and effectively use discipline-appropriate materials for both research and schoolwork. The role of the library Web site in mediating this relationship has never been greater. A CMS is an effective solution for library Web sites that eliminates many of the disadvantages of static Web pages. At UA, it succeeded in eliminating duplicate content, which resulted in easier maintenance of the content and database. It also facilitates content entry from librarians without HTML expertise, and effectively manages permissions to allow content entry by those with content to contribute.

An important lesson we learned about designing a new Web site is to keep what users demonstrate they already need and use, while eliminating functionality that is not useful to them. Usability testing revealed flaws in the initial design, which ultimately led to many improvements to the subject and course guides. Making the guides more visible on the Web site has rectified a problem that has existed from the beginning of our Web presence. Replacing user ratings with librarian recommendations has improved the user experience, particularly for novice users. Novice users now know where they should start when beginning research. The architecture of the guides has eliminated much of the scrolling, which was a common complaint of the legacy subject and course guides. In addition, contact information has been restored on the guides, which greatly aids users seeking help.

Essentially, users still need subject guidance that amounts to more than an alphabetical and filterable list of resources on a subject. With the redesign, the users gained subject and course guides with functionality that far exceeded that of the legacy system. By exploring the need for changes and then making these changes in an informed manner, we have created a Web site that helps us effectively

meet the needs of our users.  Deploying a CMS using informed Web design that is fueled by usability testing can provide a Web site that serves the library's eclectic user population more efficiently and effectively.

**References**

Connell, R. S. (2008). Survey of Web developers in academic libraries. *Journal of Academic Librarianship*, 34(2), 121-129. doi:10.1016/j.acalib.2007.12.005

Delmonte, M., & McKay, A. (2009). *The definitive guide to Plone* (2nd ed.). Berkeley: Apress.

Dupuis, J., Ryan, P., & Steeves, M. (2004). Creating dynamic subject guides. *New Review of Information Networking*, 10(2), 271-277. doi:10.1080/13614570500082931

Fennell, C. M. (2007). Content management and Web 2.0 with Drupal. *Medical Reference Services Quarterly*, 26, 143-167. doi:10.1300/J115v26S01_08

Fienen, M. (2010). *DotCMS from the ground up*. Available from http://www.lulu.com

9780931686108-0.

Fulton, C. (2010). Library perspectives on Web content management systems. *First Monday*, 15(8). Retrieved from http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/2631

Garza, A. (2009). Next Generation OPACs: Part 2 from OPAC to CMS: Drupal as an extensible library platform. *Library Hi Tech*, 27(2), 252-267. doi:10.1108/07378830910968209

Goans, D., Leach, G., & Vogel, T. M. (2006). Beyond HTML: Developing and re-imagining library Web guides in a content management system. *Library Hi Tech*, 24(1), 29-53. doi: 10.1108/07378830610652095

Greene, A. (2008). Managing subject guides with SQL Server and ASP.net. *Library Hi Tech*, 26(2), 213-231. doi:10.1108/07378830810880324

Kmetz, T., & Bailey, R. (2006). Migrating a library's Web site to a commercial CMS within a campus-wide implementation. *Library Hi Tech*, 24(1), 102-114. doi: 10.1108/07378830610652130

Miles, E., & Miles, L. (2009). *Drupal's building blocks: Quickly building Web sites with CCK, views, and panels*. Upper Saddle River, NJ: Addison-Wesley.

Morris, S., & Del Bosque, D. (2010). Forgotten resources: Subject guides in the era of Web 2.0. *Technical Services Quarterly*, 27(2), 178-193. doi:10.1080/07317130903547592

Redding, J. (2010). *Beginning Drupal*. Indianapolis: Wiley.

University of Akron Libraries, The University of Akron. (2009). *Usability testing: Results and recommendations working draft 5/7/09* (unpublished report).

Wiersma, G. (2009). Building online content and community with Drupal. *Collaborative Librarianship*, 1(4), 169-171. Retrieved from

http://search.ebscohost.com/login.aspx?direct=true&db=lih&AN=58034932&site=ehost-live

Whang, M. D. (2005). PHP and PostgreSQL Web content management systems at western Michigan University Libraries. In H. Yu (Ed.), *Content and Workflow Management for Library Web Sites: Case Studies* (149-166). Hershey, PA: Information Science Publishing.

Wiggins, R., Remley, J., & Klingler, T. (2006). Building a local CMS at Kent State. *Library Hi Tech*, 24(1), 69-101. doi:10.1108/07378830610652121