Beyond Information Architecture: A Systems Integration Approach to Web-site Design

Krisellen Maloney and
Paul J. Bracke

Users' needs and expectations regarding access to information have fundamentally changed, creating a disconnect between how users expect to use a library Web site and how the site was designed. At the same time, library technical infrastructures include legacy systems that were not designed for the Web environment. The authors propose a framework that combines elements of information architecture with approaches to incremental system design and implementation. The framework allows for the development of a Web site that is responsive to changing user needs, while recognizing the need for libraries to adopt a cost-effective approach to implementation and maintenance.

The Web has become the primary mode of information seeking and access for users of academic libraries. The rapid acceptance of Web technologies is due, in part, to the ubiquity of the Web browser, which presents a user interface that is recognized and understood by a broad range of users. As libraries increase the amount of content and broaden the range of services available through their Web sites, it is becoming evident that it will take more than a well-designed user interface to completely support users' information-seeking and access needs. The underlying technical infrastructure of the Web site must also be organized to logically support the users' tasks. Library technical infrastructures, largely designed to support traditional library processes, are being adapted to provide Web access. As part of this adaptation process, they are not necessarily being reorganized to meet the changing expectations of Web-savvy users, particularly younger users who are not familiar with traditional library organization methods such as the card catalog, print indexes, or other legacy tools.

Libraries must harness the power of the highly structured information systems that have long been a part of libraries and integrate these systems in new ways to support users' goals and objectives. Part of this challenge will be answered by the development of new systems and technical standards, but these are only a partial solution to the problem. An important part of making library systems and Web sites function as powerful discovery tools is to modernize the systems that provide existing services and content to support the changing needs and expectations of the user. Emerging concepts of information architecture (IA) describe the system requirements from the user perspective but do not provide a mechanism to conceptually integrate existing functions and content, or to inform the requirements necessary to modernize and integrate the current system architecture.

The authors propose a framework for approaching a comprehensive Web-site implementation that combines components of IA and system modernization that have been successful in other industries. Within this framework, those components are tailored for the unique aspects of information provision that characterize a library. The proposed framework expands the concept of IA to include functional and content requirements for the Web site. This expansion identifies points within the conceptual and physical design where user requirements are constrained by the existing infrastructure. Identification of these constraints begins an iterative design process in which some user requirements inform changes to the underlying system architecture. Conversely, when the required changes to the underlying system architecture cannot be achieved, the constraints inform the conceptual design of the Web site. The iterative nature of this approach acknowledges the usefulness of much of the existing infrastructure but provides an incremental approach to modernizing installed systems. This framework describes aspects of the conceptual and physical-design elements that must be considered together and balanced to produce a Web site that supports the goals and objectives of the user but is cost-effective and practical to implement.

Information Architecture and the Problem of Libraries

IA is both a characteristic of a Web site and an emerging discipline. A number of authors have attempted to develop a formal definition of IA. Wodtke presents a simple task-based definition, stating that an information architect "creates a blueprint for how to organize the Web site so that it will meet all (business, end user) these needs."[1] Rosenfeld and Morville present a four-part definition in which two parts focus on the practice, and two parts define IA as characteristic. The first characteristic defines IA as a combination of "organization, labeling, and navigation schemes" while the second describes it as "the structural design of an information space to facilitate task description and intuitive access to content."[2] There is general agreement that IA provides a specification of the Web site from the perspective of the user. The specification usually describes the organization, navigational elements,

Krisellen Maloney (maloneyk@u.library.arizona.edu) is Director of Technology at the University of Arizona Libraries, Tucson. **Paul J. Bracke** (paul@ahsl.arizona.edu) is Head of Systems and Networking at the Arizona Health Sciences Library, Tucson.

and labeling required to completely structure a user's Web-site experience. IA is not synonymous with Web-site design, but rather provides the conceptual foundation upon which a presentation design is based. Web-site design adds presentation and graphical elements to IA to create the user experience.

Library Web sites provide a display platform by which library content and services can be accessed through a common user interface. Most of the tools and services have been available for decades and, in response to user demand, are increasingly being made Web-accessible in digital formats (virtual reference, full-text databases). Despite this new access medium and format, the conceptual design of the underlying systems has not changed much. The library technical infrastructure is made up of many loosely coupled systems optimized to perform a single function or to support the work of a library department. Library Web sites do not present a sufficiently unified interface design or level of technical integration to match current users' mental models of information seeking and access.[3] The systems have not been integrated to support users' overarching goals or meet the expectation of seamless access that they have developed when using other Web sites (such as Google or Amazon). In many cases, users are still expected to understand aspects of the library that are now obsolete (card catalogs) in order to navigate the library's Web site.

For example, the process of finding a journal article using a typical library Web site is based on a print paradigm and has changed little despite the advent of online discovery tools. In a print environment, users first looked at an index to identify an article of interest, then wrote down the citation, went to the card catalog, and there looked up the journal containing the article. If the library owned the journal, the user would then write down the call number and go to the shelves to find the article. This process has not necessarily changed much for many libraries, even though indexes, card catalogs, and journals are often available online. Even more confusing is that the end result of some search processes within a library Web site is not necessarily content, but a metadata representation of content that must be entered into another search box. Although the first search is representative of the search of a traditional index and the second search is representative of the search of the card catalog, many of our users have no mental model for this multistep search process. Users accustomed to the simple keyword search available through Internet search engines may have great difficulty in understanding the need for the many steps involved in library use. There is an expectation that search systems and online content will be linked, regardless of the economic, legal, and technical factors that make these links difficult. While linking-options in vendor databases and OpenURL resolvers have begun to simplify the electronic version of the process by automat-

ing some of the steps, the multistep process is still valid in many instances in most libraries.

It is clear that library Web sites must undergo a fundamental change in order to be responsive to the needs of the user. Because library Web sites appear to be similar to conventional Web sites, it is tempting to adopt a general approach to IA to address users' needs. There are, however, several areas in which the general approach to IA does not adequately support the design needs for library Web sites.

Generalized IA approaches, such as those provided by Rosenfeld and Morville, do not provide adequate guidance regarding the organization and display of content from external sources. There is an unstated assumption that external sources will provide information in the format specified by the Web-site architect. IA approaches suggest methods to completely describe the user experience, from the time a user first accesses a site to the point at which a user task is complete, regardless of the origin of the content or service accessed. For example, the content from each of Amazon.com's commercial partners is packaged to operate like a part of the Amazon.com site. In contrast, libraries often only have control of a user's experience up to the point at which they leave a library's servers. Libraries guide users not only to local services and digitized collections, but to databases, journals, and more that are licensed from external sources and the appearances of which are controlled by external sources. Even when using a technical standard such as Z39.50 to provide a local look and feel to remote resources, libraries do not necessarily have full control over the data format or elements of the content that is returned. This lack of local control over content is a limitation to libraries adopting common definitions of IA.

Another design area that is not well supported by generalized approaches to IA is the integration of previously installed systems, such as library catalogs. These legacy systems provide important services that represent decades of development and collaboration, and are essential to the future of libraries. For example, libraries provide access to unique resources and systems ranging from online catalogs to abstracting and indexing databases to interlibrary loan (ILL) networks. Libraries are using Web technologies to provide new access methods to library content and services. These technologies provide a thin veneer on systems that function in a manner unfamiliar to many users. The challenge then becomes to change what lies beneath the surface, the underlying functionality of the site, to support the needs of the user. Using a generalized approach to IA, as applied in other settings, libraries would assess the needs of the user and develop a new, complete system that supports those needs. Such an approach ignores the extensive, existing infrastructure of legacy systems in libraries that is still useful and that serves purposes beyond the user's Web interface. What is

needed is a standard reference model for library services that provides a framework for access to services and content. This is a long-term goal that requires cooperation and agreement among libraries, and that would allow legacy systems to be repackaged in ways that are more flexible, meet changing user needs, and can be integrated into changing technology environments. Because there are currently no such reference models, librarians need to develop other approaches to integrate existing legacy systems into a modernized Web site.

Extending the IA Framework

In this paper, the general definition of IA that has been proposed by several authors has been extended to incorporate the additional constraints that characterize library Web sites.[4] Extended Information Architecture (EIA) is the first half of the framework, and provides a complete conceptual design of the Web site from the users' perspective. Figure 1 depicts the elements and relationships within EIA. The coordinating structure provides an overarching framework for the integration of the multiple service elements that provide much of the underlying functionality of the Web site. The relationship between the coordinating structure and the service elements is iterative, with service elements constraining the coordinating structure and the coordinating structure informing the design of the service elements.

The Coordinating Structure

The coordinating structure contains many of the design elements that are found in generalized approaches to IA, including the organization, navigational structure, and labeling. These are the elements of a Web site that, in concert, define the structure of the user interface without specifying the functionality and content underlying that interface. The framework emphasizes aspects of the generalized approaches that are most relevant to libraries and places them in relation to the service elements that specify the content and functionality of the site.

The first element of the coordinating structure is the organization of the Web site. Organization refers to the logical groupings of the content and services that are available to the user. These groupings are not necessarily representative of physical-system implementations, but may be task- or subject-based instead. For example, many academic library Web sites have primary groupings that include information resources, services, and user guides. Although the information resources may include information from a range of systems (for instance, the catalog, abstracting and indexing databases, full-text databases, locally-developed exhibits), the logical grouping of infor-mation resources unifies the concept for the user. A site's organization scheme will often serve as the foundation for the primary navigational choices on a site's main menu or primary navigational bar.

Another component of the coordinating structure is the navigational structure of the site. Navigational structures define the relationships between content and service elements of a site, and between groupings in the site's organization. These structures also include search tools and other link-management tools that help users locate needed content and services. There are usually two types of relationships that form a navigational structure. First is the definition of a global relationship scheme that outlines the primary navigational structure of the site. These often define relationships between sections of a site's organization, but may also provide access to key pieces of functionality from any point within a site. In addition to the overarching global relationship scheme, there are often several locally or functionally defined relationship schemes that are used throughout the site. These local relationship schemes are usually located within a service or content grouping and provide logical connections within their defined grouping. Both sets of relationships are designed to support a task and provide pathways for the user to move among the various elements of the site. Other relationship schemes may be topic oriented, allowing the user to move easily among similar content sources. These logical relationships are later implemented within a user interface as tools such as menus, navigation bars, and navigation tabs when combined with labels and a visual design.

Customization and personalization are navigational structures that have gained a fair amount of attention in the library literature. Both strategies allow a Web site to be displayed differently, based on user characteristics. Customization allows the user to create the relationships most suitable for his or her needs. This strategy has been explored by a number of libraries, although there is little convincing evidence that users implement such strategies in an intense or repeated manner.[5] Personalization allows a system designer to bring together a set of pages in a relationship that is meaningful for a user or a user group.

Labels, the third element of the coordinating structure, provide signposts that communicate an integrated view of a Web site's design to those who use it. It is important to define a labeling system that consistently and clearly communicates the meaning of the site to the user. Accordingly, the labels should be constructed in the user's language, not the librarian's. For example, a user may not understand that an abstracting and indexing database will provide them with information regarding journal articles that are relevant to a topic of interest. In that case, the label "Find an article" is more useful than "Indexes."
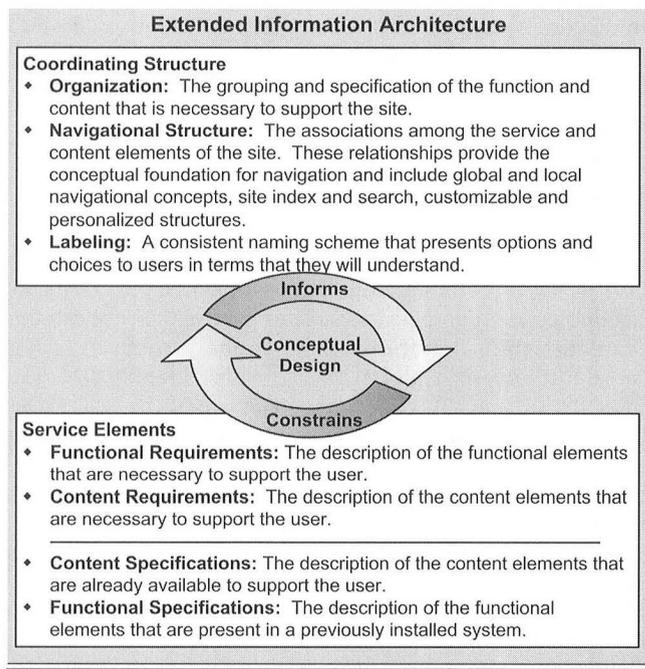
Extended Information Architecture

Coordinating Structure

- **Organization:** The grouping and specification of the function and content that is necessary to support the site.
- **Navigational Structure:** The associations among the service and content elements of the site. These relationships provide the conceptual foundation for navigation and include global and local navigational concepts, site index and search, customizable and personalized structures.
- **Labeling:** A consistent naming scheme that presents options and choices to users in terms that they will understand.

Informs

Conceptual Design

Constrains

Service Elements

- **Functional Requirements:** The description of the functional elements that are necessary to support the user.
- **Content Requirements:** The description of the content elements that are necessary to support the user.

- **Content Specifications:** The description of the content elements that are already available to support the user.
- **Functional Specifications:** The description of the functional elements that are present in a previously installed system.

Figure 1. An Extended Information Architecture for Developing a Conceptual Design of Library Web Sites

Labels are used to describe individual service or content units, but may also be used as headings to provide structural elements to augment the navigational scheme. The consistent use of labels as headings within the site not only increases user understanding of the site, but may also be explicitly constructed to support user tasks. An example of labeling to support tasks can be seen on the University Libraries Web site of the University of Louisville where, under the main heading for Articles, the first subheading is Step 1: Search article databases; and the second subheading is Step 2: Search (the catalog) by journal title.[6]

Service Elements

Service elements are the second major component of Extended Information Architecture, and represent the content and functionality of the Web site. In this framework, the service elements serve a dual purpose. The definition of service elements involves defining both the ideal requirements for functionality and content as well as the specifications of what is currently available. The definition process can then be used to identify points in the Web site where new functions and content need to be added, or where existing functionality must be modernized. These additions and modifications may be achievable immediately, but in many cases an incremental plan for change may need to be developed.

The service-element requirements, labeled as Functional Requirements and Content Requirements in figure 1, express the users' needs and expectations for the functional or content elements of the Web site. The purpose of the requirements definitions is to describe the service elements that are necessary to allow a user to meet his or her goals or objectives in using the site. These requirements are a representation of the ideal composition of a Web site, and inform not only the immediate implementation of the site but also the development of future systems and the modernization of existing systems. It is also important to note that the requirements should be developed to express user needs, not a particular implementation option. For example, it might be tempting to specify the implementation of a particular vendor's OpenURL resolver. This does not, however, describe how the system would function ideally from a user perspective. Instead, an appropriate requirement would be that users should be able to link to full text from all citations in an abstracting and indexing database.

More specifically, content requirements describe the content that is necessary to meet the users' goals and objectives. Access to content is often the primary emphasis of a library Web site, and the content requirements describe the intellectual content that should be accessible through a Web site. Examples of content that might be required are article citations, full-text articles, and multimedia objects. Normally, these requirements will be closely connected with library-wide collection-development policies and priorities, and should be driven by subject specialists rather than systems personnel. These requirements inform the development of systems to meet the needs of the users. The content specifications describe the content that is available within the current systems. There are many reasons why content requirements and content specifications do not match, including the inability or choice of a library to acquire a particular piece, the unavailability of specified content, or technical incompatibilities between content and the library's infrastructure.

Although content is sometimes viewed as the core component of a library Web site, there is also great deal of additional functionality that is provided to users. The functional requirements describe the users' needs and expectations of the functionality in the context of completing tasks on the Web site. For example, ILL forms found on many sites are easy for the user to fill out, although the most effective interface to ILL for the user might not involve a form-based user interface at all. It might be a direct system-to-system interface from an OpenURL Resolver to the ILL software in which all citation data are transmitted for the user. This requirement is not necessarily obvious when considering ILL in isolation, but is evident when considering it in the larger context of the users' goals and objective for the entire Web site. The functional specifications describe the functions

as they exist in the installed base of systems and expose the functionality that is available to the user. When the specifications do not match the requirements, the users' expectations regarding the system will not be fully achieved.

The economic and technical limitations of system implementation and modernization often reduce the speed at which the large base of previously installed systems can be modified to meet users' changing needs and expectations. It is thus critical to identify gaps between existing systems and desired systems and discover areas where a Web site will have characteristics that are not completely aligned with what the user needs or expects. When the service-element requirements do not match the service-element specifications of existing systems, an iterative design process begins. This process will be intertwined with the evaluation of the system architecture. Gaps that can be addressed immediately should be incorporated into an implementation plan for the new Web site. Longer-term migration or development plans can be developed to fill gaps that cannot be addressed immediately. It is also important to acknowledge that developing and meeting service-element requirements is an iterative process. They will need to be revisited over time as user needs change, and requirements that are met now become the specifications that are evaluated in the future.

Interrelationships within EIA

When the service-element requirements cannot be used to modify the service-element specifications, the service elements constrain the design of the Web site and influence the design of the coordinating structure. The upward arrow in figure 1 labeled Constrains indicates that the user experience is constrained by the specifications of content or functional elements that are not currently changeable. In such situations, the coordinating structure must be designed to provide additional context for the user to understand the purpose of the existing service elements. This explanatory role can be seen in the implementation of many Web sites as formal parts of the organizational structure designed to explain the idiosyncrasies of the Web site to the user. For example, many academic library Web sites have tutorials, FAQs, or sections labeled "How do I . . . ?" that provide tips on using aspects of the site that are not always evident to users.

It is necessary to acknowledge the usefulness of the explanatory role of the coordinating structure in the iterative and incremental processes of Web-site design. Just as bibliographic instruction and adequate signage have allowed the user to navigate aspects of the traditional library that were not intuitive, the coordinating structure provides the conceptual signposts and other guidance required for users to effectively navigate the Web site. At the same time, it is important to realize that the explanatory role would not be necessary if the Web site's architecture and design were intuitive to the user. As the design of the service elements changes to accommodate the larger goals of the user, the explanatory function of the coordinating structure will be diminished. The main goal of library Web site design should be to reduce the explanatory role of the coordinating structure and to develop service elements that seamlessly support the goals and objectives of the user. Until all service elements have been modernized to meet the needs of the user, the conceptual design of Web sites will represent a compromise between what users require and what it is possible for users to do within the current legacy information infrastructure.

System Architecture

While the conceptual design of the Web site describes the needs of the user apart from the technical details of the implementation, the system architecture is the description of the system as it exists. In the case of library Web sites, the system architecture is not limited to the functionality and data on the library's Web server. Instead, it is also inclusive of all core infrastructure, individual systems, and data access and storage mechanisms that provide the blueprint of the Web site's backend as it has been built. The individual systems in the architecture may include locally controlled ones, (for instance, an online catalog), but will also include remote systems such as abstracting and indexing databases mounted by a vendor. A definition of the design of the existing system plays a key role in the evolutionary specification of the system because it provides developers with a greater understanding of the possibilities and constraints of the existing infrastructure. In describing a system architecture, several formal representations can be used that capture various aspects of the system's capabilities at different levels of granularity. These include module views that provide static specifications of individual components; component and connector views that provide dynamic views of processes; and deployment views that incorporate hardware elements.[7] The selection of representations is beyond the scope of this paper.

Typical elements of a system architecture can be seen in figure 2. For the paper, three classes of components are being considered, although more may be introduced if applicable locally. The core-infrastructure components are fundamental services and information that support one or more systems or subsystems. In a typical library environment this includes authentication services, Web platforms, and the network. In some library environments, external

units may maintain some or all of these components. For example, many college campuses maintain an authentication infrastructure in the campus computing office. Overall, core infrastructure provides the glue for tying together the many applications that libraries attempt to integrate in their Web sites. The system architecture should include details regarding the standards and interfaces that are used within the library technical infrastructure.

Many of the applications in the library environment are off-the-shelf components that have been developed by external vendors. These off-the-shelf components may include the catalog, ILL modules, electronic-course reserves, and virtual-reference systems. Although individual libraries may have some control over configuration options in these applications, they are likely to have little influence over the basic functionality or data formats provided by these systems. Core functionality tends to change based on the demands of many libraries looking for similar functionality. Despite the lack of functional control over these systems, components developed by external vendors may provide standards-based system interfaces to their functionality. These usually take the form of industry-supported standards or vendor-supplied application programming interfaces and give libraries some flexibility in working with these components. Explicit descriptions of the available standard and proprietary interfaces should be included within the system architecture.

Other applications may have been developed within the library and so can be changed more easily. Examples of locally developed applications typically include subject pages, information about the library, and digital Web exhibits and collections. Although local development does provide more control over the appearance and functionality of a piece of software, it is not without problems. Local development is often conducted using a *bricolage* approach, solving specific problems singularly, without giving consideration to the larger networks of systems in which the solutions operate. When such approaches do not take into account larger issues of systems architecture, opportunities to solve a broader range of problems may be missed and subsequent repackaging of these solutions may be limited or impossible. Libraries frequently also have a limited number of programmers, often remedied by pulling librarians or staff from other duties. While this certainly can allow libraries to meet some user needs, the lack of software-engineering skills in libraries may result in local solutions that are inflexible and that do not support standards for data storage or interchange. Because the internal design of these applications is accessible and modifiable, the system architecture should include more extensive descriptions of the internal features and relationships that they contain. Although this will not completely alleviate the problems of software maintenance, it will provide a better foundation for decisions regarding future migration.
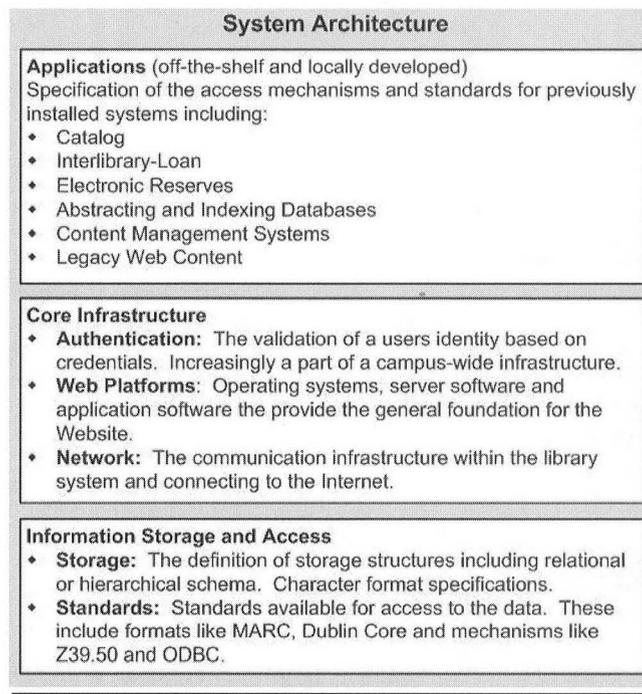
System Architecture

Applications (off-the-shelf and locally developed)
Specification of the access mechanisms and standards for previously installed systems including:
- Catalog
- Interlibrary-Loan
- Electronic Reserves
- Abstracting and Indexing Databases
- Content Management Systems
- Legacy Web Content

Core Infrastructure
- **Authentication:** The validation of a users identity based on credentials. Increasingly a part of a campus-wide infrastructure.
- **Web Platforms:** Operating systems, server software and application software the provide the general foundation for the Website.
- **Network:** The communication infrastructure within the library system and connecting to the Internet.

Information Storage and Access
- **Storage:** The definition of storage structures including relational or hierarchical schema. Character format specifications.
- **Standards:** Standards available for access to the data. These include formats like MARC, Dublin Core and mechanisms like Z39.50 and ODBC.

Figure 2. Elements of a System Architecture

Finally, typical library architectures consist of links to resources that are licensed or organized on behalf of the user. These include abstracting and indexing databases, full-text content provided by publishers outside of the library, and general vetted Internet sites. Linking the user to the system usually provides access to these systems, and libraries have no control over the technical implementations of such resources. Newer federated search technologies are integrating into the library infrastructure the users' access to the site and to results from the sites, and linking tools make the interrelationships between these systems more easily understood. Nevertheless, integrating these resources into a Web site in a manner that makes sense to library users is a challenge. The access mechanisms and information formats required to communicate with the site should be clearly documented within this system architecture.

Interrelationship of the Information and System Architectures

Reacting to the rapid pace of change can result in an ad-hoc or haphazard approach to Web-site design. The sections above describe a systematic approach to include and evaluate changes to the Web site. In order to imple-

ment the changes and create a Web site that is scalable and made of reusable components, it is necessary to evaluate, plan, and document all changes to the system. Figure 3 graphically depicts the interrelationship between EIA and system architecture. User needs, as described by IA, should inform the development of technical infrastructure. The Informs arrow indicating that EIA informs the design and development of the system architecture depicts this interrelationship. The Constrains arrow designates the reality that some aspects of the existing infrastructure cannot be changed within this planning cycle and will limit the library's ability to immediately change the underlying content and function of the Web site. When mapping the conceptual design to the physical design, there will be gaps that represent functionality that cannot be supported, either fully or in part, by the current system architecture and thus constrain the full implementation of the conceptual design.

If IA is then to be implemented as fully as possible, these gaps identify the modifications and additions that must be carefully evaluated, designed, and implemented within the underlying system architecture. Gaps can be addressed in a variety of ways. If there is a total gap in functionality, a system can be developed or implemented to provide the desired functionality as part of the larger system architecture. This may result in a complete development project or in the specification of an off-the-shelf application to meet the newly identified demand.

In the case where an existing system has some of the required functionality but is not completely suitable for the users' goals and objectives, an incremental approach of modernization can be adopted. Modernization surrounds "the legacy system with a software layer that hides the unwanted complexity of the old system and exports a modern interface."[8] This is done to provide integration with a modern operating environment while retaining the data and exposing the functions of the existing system, if desired. Techniques range from screen scraping to the implementation of Web services to export access to functions that are still relevant within the new context. All of these changes become part of the system architecture for future iterations of change. Gaps that cannot be immediately added or changed to meet the specified requirements become constraints in the next iteration of conceptual design.

In the absence of a plan, the underlying systems will continue to undergo constant evolutionary changes, ostensibly to meet the changing needs and workflows of both users and staff. Change comes from many sources, including local implementations and modifications, external vendors, and industry-wide changes in standards. This rapid but incremental change can produce a system that is very difficult to maintain and that provides few reusable modules. Having a well-documented implementation and integration plan will not guarantee that
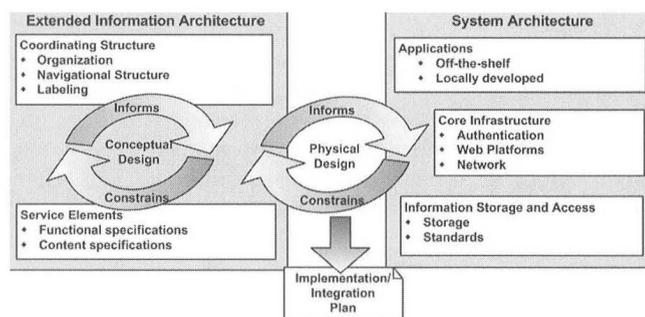

Figure 3. The Interrelationship between the Conceptual and Physical Design of the Library Web Site

the library will not experience the negative effects of technological change, but it does allow a library to better manage change in meeting the needs of its users. The more explicitly and clearly the modifiable features are documented within the system architecture; the easier it will be to plan to fill the gaps.

Conclusion

Library users' mental models of library processes have fundamentally changed, creating a serious disconnect between how users expect to use a library Web site and how the site was designed. In particular, user expectations regarding the number of steps that must be completed have changed. At the same time, library technical infrastructures are composed, in part, of legacy systems that provide great value and facilitate interlibrary resource sharing, but were not designed for the Web environment. It is essential that libraries develop new approaches to the conceptual design of Web sites that support current and future changes to both user behaviors and to library systems architectures. In the long run, these approaches should contribute to the development of a reference model for the description of library services.

The authors have proposed a complete framework for conceptual design and physical implementation that is responsive to changing user needs while recognizing the need for libraries to adopt an efficient and cost-effective approach to Web-site design, implementation, and maintenance. Functional and content needs of the user are identified and molded into a conceptual design based on a broadened perspective of the users' objectives. Mapping conceptual requirements to physical architectures is an important part of this framework, using an architectural representation in combination with descriptions of integration elements that have been developed to support the incremental and iterative change.

The ability to respond is essential, necessitated by the rapid change in the technical and user environments in which libraries operate. The framework is designed to allow logical and informed decisions to be made throughout the process regarding when to create new systems, when to replace or modernize existing systems, and when to improve the conceptual signage of the Web site.

References

1. Christina Wodtke, *Information Architecture: Blueprints for the Web* (Indianapolis: New Riders, 2003),

2. Louis Rosenfeld and Peter Morville, *Information Architecture for the World Wide Web*, 2nd ed. (Cambridge, Mass.: O'Reilly, 2002), 4.

3. Bob Gerrity, Theresa Lyman, and Ed Tallent, "Blurring Services and Resources: Boston College's Implementation of MetaLib and SFX," *Reference Services Review* 30, no. 3 (2002): 229–41; Barbara J. Cockrell and Elaine Anderson Jayne, "How Do I Find an Article? Insights from a Web Usability Study," *Journal of Academic Librarianship* 28, no. 3 (May 2002): 122–32.

4. Jesse James Garrett, *Elements of User Experience* (Indianapolis: New Riders, 2002); Rosenfeld and Morville, *Information Architecture.*

5. James S. Ghapery and Dan Ream, "VCU's My Library: Librarians Love It . . . Users? Well Maybe," *Information Technology and Libraries* 19, no. 4 (Dec. 2000): 186–90; James S. Ghapery, "My Library at Virginia Commonwealth University: Third Year Evaluation," *D-Lib Magazine* 8, no. 7/8 (July/Aug. 2002). Accessed July 16, 2003, www.dlib.org/dlib/july02/ghaphery/07ghaphery.html.

6. University of Louisville Libraries Web site (2003). Accessed July 16, 2003, http://library.louisville.edu.

7. Craig Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design* (New Jersey: Prentice Hall PTR, 1998); Martin Fowler, *Analysis Patterns: Reusable Object Models* (Boston: Addison-Wesley, 1997); James Rumbaugh, Ivar Jacobson, and Grady Booch, *The Unified Modeling Language Reference Manual* (Boston: Addison-Wesley, 1999); Robert C. Seacord, Daniel Plakosh, and Grace A. Lewis, *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices* (Boston: Addison-Wesley, 2003).

8. Seacord, Plakosh, and Lewis, *Modernizing Legacy Systems*, 9.