Abstract
During the last decade, software designers have made progress in developing usable systems for products such as word processors. Less attention has been given to usability in digital library (DL) design. In this paper, we discuss two forms of DL usability - interface and organizational. While the Human-Computer-Interaction research community has helped pioneer design principles to improve interface usability, organizational usability is less well understood. "Design for usability" is a new term that refers to the design of computer systems so that the organizational usability is addressed. DL developers need to consider "design for usability" issues during DL system design.

Anthropologists have successfully used cultural models to understand the cultural constructs by which people view the world. Computer systems development communities, including the DL design community, usually have some consensus (cultural models) about the role of usability in their development process. We discuss five typical models of computer systems design. These models become cultural models when they are taken for granted within a professional community as THE way to design all systems. A new model that incorporates "design for usability" principles into system design is proposed. We believe that this model has the strongest chance of producing usable DL systems.

Keywords: Cultural models, usability, usability engineering, user interface.

1. Introduction
During the last decade, software designers have learned to craft systems that are easier for many people to learn and to use so as to improve their performance at work. Although some progress has been made in developing usable systems for such software as spreadsheets and word processors, less progress has been made in the design of more complex information systems such as a digital library (DL). Even talented computer scientists find some DL systems that they should find useful to be too clumsy to bother with, and so they avoid using them. Systems need to be created which are easy to learn and to use in order to prevent the insidious problem of "underused" systems.

"Systems usability" refers to how well people can exploit a computer system's intended functionality. Usability can characterize any aspect of the ways that people interact with a system, even its installation and maintenance. Usability issues should be considered during the design of digital library (DL) services in order to build systems which people with limited technological skills can readily use. For the NII to be successful in its attempt to "wire" the United States to a superhighway (NREN) of information within the near future, DL developers need to consider how to accommodate users' mix of skills, work practices, and resources to maximize DL usability. Otherwise, the realization of NII will be hampered by "underused" DL systems.

In this paper, we discuss two key forms of DL usability - interface usability and organizational usability. The Human-Computer-Interaction (HCI) research community has helped pioneer design principles to improve interface usability. And software product firms frequently advertise their products in terms of more usable interfaces. In contrast, organizational usability is less well understood. Since it depends upon the "fit" of systems to specific organizations, it cannot be readily packaged, advertised, and sold. "Design for usability" is a new term that refers to the design of computer systems so that the organizational usability is addressed. DL developers do not consider "design for usability" issues during DL system design. We raise this issue to stimulate DL designers to think in these terms. This is pertinent now since DL design is in the early stages of conception.

The concept of "cultural models of computer systems design" helps us explain how different system developers interpret the meaning of the usability of their products. Anthropologists have been successful in using the concept of cultural models to understand the cultural constructs by which people view the world and how they internalize these constructs [3, 14]. For example, anthropologists have used cultural models to study romance, marriage, parenthood and success of women and men from different class and ethnic backgrounds of people [3]. Computer systems development organizations usually have some consensus (cultural models) about the role of usability issues in their development process and resulting products. Similarly, DL development communities also have a cultural model of the role of usability in their design process.

In this paper, we discuss five models of computer systems design which are typically found in the computer industry and research community. These models become cultural models when they are taken for granted within an organization or professional community as THE way that all systems should be designed. In addition, we propose a new model that incorporates "design for usability" principles into system design. Our belief is that this model has the strongest chance of producing DL systems which many people find to be routinely usable in their workplaces. It is our intent to provoke thought and research on this important topic in DL development.

Section 2 describes the dimensions of usability and how they apply to DLs, section 3 conveys how the content of DLs affect their usability, and section 4 outlines the "design for usability" principles applied to DLs. Section 5 describes the use of cultural models as frames of reference for design approaches, section 6 presents six models of computer systems design, and section 7 contains the conclusion.
2. Dimensions of Usability Applied to Digital Libraries

In this section, we outline two key forms of DL usability: interface and organizational. The interface dimensions are centered around an individual's effective acclimation to a user interface, while the organizational dimensions are concerned with how computer systems can be effectively integrated into work practices of specific organizations. Interface usability can be tested in a usability laboratory setting using a sampling from a relevant population, but organizational usability must be measured by its "fit" within specific organizations. Examples of these dimensions applied to DLs are given. We do not contend that one DL is better than another through these examples, but wish to show the differing dimensions of usability in DLs.

2.1. Interface Usability

Usability can characterize any aspect of the ways that people interact with a system, even its installation and maintenance. Nielsen [15] discusses five attributes of a system's interface usability. He suggests that these attributes can be evaluated through usability testing relative to certain users and certain tasks. Four of these are listed and discussed below as interface usability dimensions:

1. Learnability - Ease of learning such that the user can quickly begin using it.
2. Efficiency - Ability of user to use the system with high level of productivity.
3. Memorability - Capability of user to easily remember how to use the system after not using it for some period.
4. Errors - System should have low error rate with few user errors and easy recovery from them. Also no catastrophic errors.

As an illustration of how these four attributes of interface usability contribute to effective usage of DLs, a comparison is made between two DL services - Gopher and Mosaic. Gopher is a DL indexing service which enables researchers to search for electronic files containing information on a particular topic over a network of computers - both nationally and internationally. Its user interface is menu-driven with deep levels of submenus. Mosaic is a networked information discovery, retrieval, and collaboration tool and World-Wide Web (WWW) browser. Gopher is also accessible from Mosaic. WWW merges information discovery and hypertext techniques for linkage to text, sound and animation files.

The learnability of Gopher requires technical understanding of the platform's operating system (Gopher can be used on many platforms: Unix, Dos, Microsoft Windows, Macintosh, Next, and VM). Irrespective of the platform, people must learn how to navigate Gopher submenus and how to research, retrieve, and save information. Similarly, learning to use Mosaic requires ready knowledge of the windowing system upon which it resides (Mosaic is available for X-windows, Windows for DOS, and Macintosh) and familiarity with the hypertext concept. Mosaic includes online demonstrations and detailed "help" menus, both of which are helpful to novice users. While Gopher lacks the online demonstrations, it does include a shorter version of a "help" system. Some people might consider Gopher's interface easier to learn than Mosaic's because Gopher's text-based menu-driven system may be more familiar to them than Mosaic's graphical mouse-driven windowing system requiring knowledge of hypertext links.

Efficiency of use refers to an expert person's steady-state level of performance when no more learning takes place. This can be measured by defining a level of expertise and testing a representative sample of people who have that expertise by timing some typical tasks. Gopher's efficiency of use could be measured by giving experienced people a topic to research, or a question to answer. Similarly, Mosaic's WWW efficiency of use could be tested using the same technique. In both cases, the longer people use these DL interfaces, the more adept they become at seeking information. Since these interfaces are amorphous by their definition, efficiency of use over an extended period of time might vacillate and be difficult to measure.

The memorability of Gopher's interface appears to be higher than Mosaic's. It is simpler and one can save bookmarks to a particular search retrieval location to be used in a future session. In contrast, Mosaic's interface requires knowledge of X-windows and the proper usage of a mouse to navigate the windows - a skill which may need to be relearned after a period of abstinence. Saving Mosaic's search paths consists of collecting a Uniform Resource Locator (URL), a label associated with a retrieved location, into a "hotlist". The items in the "hotlist" menu - analogous to Gopher's bookmark list - may be selected again. However, Gopher's bookmarks, labeled with the same nomenclature of the original menu selection, are easily identifiable. In contrast, Mosaic's "hotlist" items are named with a portion of the URL which may not be unique, requiring editing of the "hotlist" items to translate them into easily remembered selections.

One of the errors which could occur during usage of either system is the improper use of retrieval keywords. Searches using Gopher sometimes appear to invoke directory and file selections which are more pertinent to the search topic than a WWW search through Mosaic. For example, we performed an experiment where we chose to search for a specific topic - computer usage at the government agency, Center for Exploited and Missing Children. We searched with the string "Missing Children" in both systems. Gopher returned menu items directly pertaining to missing children. Whereas, Mosaic's WWW search produced unrelated items, such as one on the advertisement for flower baskets purchased over the Internet. However, Gopher is often unreliable, failing to access specific directories and getting "stuck" in a loop during searching. A skilled and patient user can easily overlook these reliability problems. But we have found that some academics find Gopher's behavior so confusing or annoying that they abandon Gopher as a viable work tool.

2.2. Organizational Usability

Organizational usability refers to the ways that computer systems can be effectively integrated into work practices of specific organizations. As we shall show, the same DL may be much more usable in some organizations than in others. In this section, we examine four important characteristics of the fit between DLs and organizations which make them more or less usable by people in support of their work. We illustrate them with the design of Gopher and Mosaic and also the way that we use these systems at UC Irvine.
The organizational usability dimensions include:

1. **Accessibility** - Ease with which people can locate specific computer systems, gain physical access and electronic access to their electronic corpuses. This dimension refers to both physical proximity and administrative/social restrictions on using specific systems.

2. **Compatibility** - Level of compatibility of file transfers from system to system.

3. **Integrability into work practices** - How smoothly the system fits into a person or group's work practices.

4. **Social-organizational expertise** - The extent to which people can obtain training and consulting to learn to use systems and can find help with problems in usage.

The accessibility of Gopher and Mosaic varies with an individual's exposure to the necessary platforms. The availability of platforms has a strong influence on whether people can use these systems. At UCI, computer science undergraduates are permitted access to Gopher (for limited periods of time) but not to WWW, because Mosaic requires workstations which are only available in the computer science graduate student laboratories. UCI's undergraduate students are also not permitted to telnet to locations where Gopher or WWW servers exist. However, in the near future, the UCI library plans to install Mosaic on three workstations which will be for general use.

The compatibility dimension of DL usability refers to the retrieval of files in text, graphics, audio, and/or animation format. Text retrieval in Gopher and Mosaic for ASCII files is compatible for most computer systems. However, these files may be in a particular word processing format which is not easily convertible to the individual's preferred format. In the X-Windows version of Mosaic, hypertext links of sections of a paper are retrievable by selecting a hyperlink. If one then chooses to save this selection from a menu, it is saved in the format and size of the current window. This format may not be easily ported to a printer at a future time. In addition, these hypertext links may comprise a list of sections of a document. So instead of retrieving the document as one file, one may need to retrieve the document in pieces. In Gopher, documents are retrieved and saved as one file. The usage of graphics, audio, and animation files retrieved from these DL services depends on the availability of correct translators and platforms on the receiving end.

The integration of Gopher and Mosaic into work practices is a function of the accessibility and performance of the platforms. For a professor or student with time constraints on the need for a body of text, Gopher's long delays during file retrievals could impede a person's progress. Mosaic's WWW browsing may also deter someone from quickly locating a specific piece of information. However, for people with more time and patience, these DL services would fit better into work practices. Also, a style of work which involves exploratory procedures lends itself well to the use of DL services. People without a computer in the office or nearby may not be able to easily integrate DL research into their work place.

The availability of the social-organizational expertise concerning Gopher and Mosaic influences the ready access of these DL services. For example, UCI's students and faculty typically can contact local support organizations on campus when questions arise. In some cases, training sessions are provided in these settings. Conversely, a person using these systems from a home computer lacks access to experts and may need to rely on local bulletin boards for help.

3. **Content of Digital Libraries and Usability**

A great deal of people's satisfaction is influenced by the size and content of the corpus of a DL service. Some people using Gopher may become dissatisfied and/or confused with the unlimited size and content of the corpus when searches result in retrievals of directories and files located on databases worldwide. On the other hand, others may find that this vast wealth of information is motivating and very satisfying. Similarly, while some people using WWW may be unhappy with the "unknown" links reached by selecting hypertext links, others may find this very appealing. In both cases, these DL services have unforeseen boundaries around the data provided. In contrast, consider a university computerized library search system such as the University of California's Melvyl system. People using this type of system are aware of the location of the corpus being searched and might be more satisfied with this more direct search for information in "known" locations.

The incentive for using a DL service is highly dependent on the content of the corpus, which may be useful to one person and not useful to another. Usefulness is the capability of the system to be used to achieve a predetermined goal. A system's usefulness to an individual is influenced by the extent to which that person knows they will find something useful. Usefulness is different from usability in that it is germane to individual preferences towards a DL service. For example, a search for information on a specific topic using Gopher may result in dozens of files which are not useful. Since there are no strict naming conventions or regulation of content on Gopher, some retrieved files may be worthless to a researcher or even empty. The results of such searches are not useful to the person seeking information. An advantage of Mosaic is that hypertext links sometimes connect to the Wide Area Information Server (WIAS), which performs full-text search for keywords, returning a list of selections with ranks from 1 to 1000 points depending on how many search words appeared in the document and how many times. This may be useful when weeding out irrelevant material.

4. **"Design for Usability" in Digital Libraries**

"Design for usability" is a new term that refers to the design of computer systems so that they can be effectively integrated into the work practices of specific organizations. It goes beyond the focus on user interfaces. "Design for usability" includes designing the infrastructure of computing resources that are necessary for supporting and helping people learn to effectively use systems. "Design for usability" encourages system designers either to accommodate to people's mix of skills, work practices, and resources or to try to systematically alter them. "Design for usability" can be applied to the selection and integration of diverse computer systems or to the...
DL computer systems should be a prime target for "design for usability". Their effective value depends upon their practical usability by people in organizations and communities. The emerging DLs have come to include publicly available information and private information shared by collaborators: reference volumes, books, journals, newspapers, national phone directories, sound and voice recordings, images, video clips, scientific data (raw data streams), and private information services such as private newsletters and stock market reports.

We know very little about the working conditions and institutional and organizational practices that make DLs most usable by faculty, academic staff, students, business associates, and people using home computers. Developers take it for granted that men and women using DLs are self-serving. Research [2] shows that people appear to need help in effectively using DL services, but they are being pushed by DL developers to "self-service". Developers of DL systems need to address complex usability issues to ensure that many people gain value from using these services. In fact, we argue that individuals need to learn about the services' contents, learn to use them, and have access to complementary computing resources (like disk space and printers) to effectively utilize their output.

Figure 1 illustrates the set of players involved in the design and usage of a DL in a typical university. It indicates which groups should consider "design for usability" when making design decisions. First, the vendor-based design team (often influenced by researchers), create shrink-wrapped product designs which are distributed by the content providers to end-users - in this case, faculty and students. Both the design team and the content providers need to address usability issues to provide DL systems which will be effective for people's varied skills, work practices and resources. Within the organizationally-specific box, the structuring group needs to consider the organizational dimensions of usability including the accessibility and integrability of DL services into the work place. The box containing faculty, students, local consultants and gurus are the people most affected by the dimensions of usability. Our model of "design for usability" for DLs, presented in section 6, is intended for designers, content providers and organizationally-specific structuring groups in order to promote usable DL systems.

5. Cultural Models as Frames of Reference for Design Approaches

Computer systems development communities view usability differently and each has a cultural model of systems design which motivate them to use one model versus another. Further, the DL community acts as a culture and has very specific models of design to be discussed later. Anthropologists have been successful in using the concept of cultural models to understand the cultural constructs by which people view the world and how they internalize these constructs [3, 14]. For example, anthropologists have used cultural models to study romance, marriage, parenthood and success of women and men from different class and ethnic backgrounds of people [3].

Shared knowledge, including

values and assumptions, within a group is part of what anthropologists call culture. Our connection with culture influences how we make sense of different situations, and how we interpret actions as pertinent to any given situation [7].

Cultural models are "presupposed, taken-for-granted models of the world that are widely shared (although not necessarily to the exclusion of other, alternative models) by the members of a society and that play an enormous role in their understanding of that world and their behavior in it [14]." Cultural models can have motivational force [14] because these models not only define the world but also set forth goals (both conscious and unconscious) and invoke or include desires [3].

6. Cultural Models of Computer Systems Design

Computer systems development industry and research communities usually have some consensus (cultural models) about the role of usability issues in their development process and resulting products. These cultural models of usability differ in the priority that they place on producing computer systems that are easy to use relative to other functional requirements. They also include typical ways for thinking about "who is a user", "how hard should people have to work to learn a system", "whether to include the user in the design process", etc.

We present several models of computer systems design. The traditional functional life-cycle model, the user interface model, and the usability engineering model are discussed as models of computer systems design. The highly automated model seems to be a cultural model for the DL design world, and the artificial intelligence (AI) model of medical informatics design represents the cultural model of AI expert systems design. The last model - the organizationally sensitive "design for usability" model - represents a new approach to ways that software developers can design their systems to ensure usability. These models become cultural models when they are taken for granted within an organization or professional community as THE way that all systems should be designed.

6.1. Traditional Functional Life-cycle Model of Computer Systems Design

The traditional functional life-cycle approach to computer systems design includes six stages performed sequentially: Project Definition, Systems Study, Design, Programming, Installation, and Post-implementation. Each stage includes a set of activities which must be completed before advancing to the next stage. This is a fairly rigid, simple model of computer usability devoid of the influences of the surrounding computing infrastructure -- the physical, technological and social supporting resources. While usability issues are addressed, the concerns are often supplanted by cultural influences of how developers should view the needs and preferences of people who will use their systems.

Typically, communication between the developer and end-user is frequent during the requirements specification period. But once requirements are set, the level of communication diminishes and is mostly concerned with budgetary matters. For example, the Project
Definitions step might entail assessing the end-user profilein preparing a project proposal report. One possibility is for a designer to involve the end-user which includes interviewing prospective end-users. In contrast, a different designer might decide not to involve the end-user, and choose to determine the profiles of the people who will use the system by his or her individual assumptions. The instigation of one method versus another depends on individual backgrounds and organizational cultural constructs. For example, the motivations for choosing not to involve the end-user might include:

1. "Conform to standards" - It may be a standard practice in an organization to exclude the end-user in requirements planning. Workers may want to maintain their position by following orders, or they may just want to "fit in" with the norm.

2. "Expectations that end-users will view the system as the developer does" - Involving the end-user in seeking requirements is not necessary because assumptions are made that future end-users "think" like the developer.

3. "Designer is developing a new technique with the attitude that I'm the expert in this area and the future end-users might not understand what they want."

4. "Rote application of present or former training" - The developer may have had training in school or in a current or previous organization, where it was learned that requirements documents are written without consulting with the end-user.

Typically, this model does not involve the end-user in very many steps during system development. Consequently, computer systems developed using this model often have serious drawbacks when implemented and become "underused" systems by the people for whom the system was designed.

6.2. User Interface Model of Computer Systems Design

The user interface model of computer systems design is proposed by Rettig [16] for users like himself who have never systematically addressed usability issues, except to try to make software look like other tools with which users are familiar. This model is a typical contemporary Human Factors model. Rettig bases his model on personal experience with a software development project which needed a manageable method for users to conceive, view, and manipulate complex data structures.

The user interface approach to computer design includes more concern for the potential people who will use the system than the traditional approach. One way is through emphasis on ergonomics, the ease with which a person can work in the physiological surroundings of a computer system's environment. This is an improvement over the traditional model but it still has limitations. One study [1] on the use of groupware, software which supports collaborative work groups, showed that even though groupware systems provide a myriad of services - scheduling, calendaring, and electronic mail - most people opted to solely use the groupware for electronic mail and ignore the other functions. In this case, the groupware's user interface with its many options did little to encourage usage in the work environment.

Iterative design and testing is emphasized in this model with the end-user's involvement permeating every step. Motivations for including the end-user in this model might include:

1. "Conform to standards" - Organizational policy might dictate end-user involvement.

2. "Ensure usability in final product" - Developer might foresee the need for end-user involvement as a prerequisite for useful systems.

Efforts to focus on the end-users' needs and involve them in the design process with user testing early and often are integrated into this model. This model succeeds in application to shrink-wrapped products such as word processors or airline reservations systems where the user interface is the primary concern. For computer systems requiring assimilation into a person's work practice such as DLs, this model has limitations. The computing infrastructure including services for training and support are not considered.

6.3. Usability Engineering Model of Computer Systems Design

The usability engineering model is a leading-edge HCI approach, and extends the user interface model. Usability engineering [15], a systematic guide to ensuring a high degree of usability in the final user interface, is the basis of this model. In this model, a heightened awareness of user preferences and needs -- more so than in the user interface model -- is proposed. The usability engineering life cycle model includes these stages proposed as a paradigm for companies to follow:

1. Know the user - Study intended users and use of the product. At a minimum, visit customer site to study user's current and desired tasks, and to understand the evolution of the user and the job.

2. Competitive analysis - Analyze existing products according to usability guidelines and perform user tests with products.

3. Setting usability goals - Establish minimal acceptable level of usability and estimate the financial impact on cost of users' time.

4. Parallel design - Use several designers to explore different design alternatives before deciding on one final design.

5. Participatory design - Include end-users throughout design phase.

6. Coordinated design of the total interface - Maintain consistency across screen layouts, documentation, on-line help systems, and tutorials.
1. Apply guidelines and heuristic analysis - Select user interface guideline appropriate for situation.

2. Prototyping - Build prototype to pretest on end-users.

3. Empirical testing - Test end-users on specific usability attributes.

4. Iterative design - Capture design rationale through iterative testing and design.

5. Collect feedback from field use - Gather usability work from field studies for future design.

As in the user interface model, the motivation to involve the end-user is used by developers throughout the application of this model. Consideration of the preferences of potential end-users is strongly adhered to in this model through the use of prototyping and iterative usability testing in which end-users are repeatedly measured on their effective use of revisions of the system until a satisfactory product emerges.

There are several means of measuring usability. One is to select a representative group of users from the potential user population and have them use the system in a predetermined set of tasks. Another is to observe real users in the process of performing the tasks they normally do as part of their job. In both cases, the system's usability is measured against certain users and certain tasks. Those measurements are part of Step 10, Iterative testing, in the usability engineering life-cycle. Based on the values of these measurements during the first testing, the system is revised in hopes of improving the attribute values closer to the optimum.

Although this model more closely attends to the needs and preferences of potential end-users than the traditional or user interface models, it is still lacking in its assessment of the "fit" of these systems into people's social and organizational environment. Even though systems may be designed with slick user interfaces, there is no guarantee that user acceptance is automatic as we shall see in the medical informatics model of artificial intelligence (AI) design.

6.4. Medical Informatics Model of AI Design

Even more important than having a user interface that is comfortable is having a system that saves work and adds expertise which comes from people's work. In the expert system domain of medical informatics, AI systems have been developed for the medical profession but nobody seems to want to use them. One of the things that worries researchers and developers is that AI designers of medical informatics systems continue to design expert systems which people do not use.

Forsythe [4, 5] has conducted a long-term study of the culture and action in AI design. Her research focuses on the relationship among the values and assumptions that a community of scientists bring to their work, the practices that comprise this work, and the tools which they construct as a result of this work. Her study of the non-acceptance of medical expert systems illustrates some of the key assumptions that AI developers make about usability [4]. After an extensive study of computer scientists who develop medical expert systems, she concluded: "...the problem of user acceptance is to a significant extent the outcome of values and assumptions that the scientists bring to their own research and development process. While I do not suggest that this connection is conscious or intentional, the nature of this particular tradition of scientific practice makes it very difficult for its followers to see or entertain strategies from other traditions that would probably help to solve their problem [4, p. 100]."

In essence, the AI developers of medical expert systems see the problem of user acceptance as the medical community's inability to use their expert systems. They do not seriously question their own abilities to design systems that will match the work styles of doctors, nurses, or medical technicians. She outlines some dysfunctional beliefs and assumptions of these AI medical researchers:

1. Technical bias - Focus on technical factors of problem assessment, design and evaluation.

2. Decontextualized Thinking - Use of simple, quantitative models for problem-solving and evaluation. AI researchers believe that they gain design power by ignoring working conditions. But this leads them to produce systems as cultural artifacts that medical doctors typically can not use. AI professionals prefer this decontextualized thinking so that they do not involve physicians in their design. Consequently, doctors judge these systems as irrelevant to their work.

3. Quantitative, Formal Bias - Use of a formal perspective on problem-solving based on computer science format training in mathematics, logic, or a physical science.

According to Forsythe [4], these beliefs and assumptions of the cultural model of medical informatics design prevent the AI developers from constructing expert systems which the medical profession is able to routinely use.

6.5. Highly Automated Model of DL Design

We will briefly discuss a design model which we believe is a cultural model of DL design in the DL research community. This model includes many of the steps of the traditional, functional life-cycle and user interface models. However, there is no pointed effort in this design model to study the needs and preferences of people who will use the system in their workplaces, nor to investigate what computing-support infrastructure is necessary for training and continuing DL use.

For example, Lesk, Fox, and McGill [12] call for a basic science, engineering, and technology library on-line accessible over the Internet/NREN and for research to build and exploit it. The usability issues regarding how these facilities would fit into work organizations have not been addressed as a research topic. Instead, recommendations were made to incorporate sophisticated computerized techniques into DLs of the future such as "agent" programs which do the searching and then return documents:
A traditional library is passive; users come to it and search for what they want. In an electronic library it is possible to have pro-active elements, which send users messages when agents discover items that, based on earlier commands, it is known the users are looking for. In the electronic network of the future, we must compensate for the inability of the users to look around rooms of physical books. To facilitate this, we can build either asynchronously running programs which sit in the background, waiting for new information to arrive, as in SDI (selective dissemination of information) programs; or we can provide sophisticated “agent” programs, such as the “knowbots” described by Vinton Cerf, which are dispatched by users to search out and return desired documents. To truly serve a community of users these must learn about the needs, preferences, and capabilities of those users and consider that information during both search and presentation phases [12, p. 14]."

They suggest that the research should focus on the creation of on-line resources in varied scientific areas, to be used in new and exciting ways. These researchers are following a form of the method of involving the end-user by surveying communities of potential end-users for general requirements. However, they don’t identify people’s individual work practices, and institutional and organizational practices as explicit considerations for DL design refinements. One of the cognitive beliefs of the AI cultural model, "Decontextualized Thinking", is closely related to the preference of DL designers for SDI programs and "knowbots". The attention to spiffy interface assistants focuses the DL design on quantitative problem-solving rather than on organizational issues of the appropriate "fit" of future DL systems into people’s workplaces.

This DL design model shares some key features with the medical informatics design model. Forsythe’s results suggest that it would be risky for DL developers to ignore usability issues in designing computer systems. Otherwise their systems, like the AI expert systems, may end up "on the shelf" with low levels of end-user acceptance.

### 6.6. Organizationally Sensitive Model of "Design for Usability"

The organizationally sensitive model of "design for usability" is a new model. It refers to the design of computer systems so that they can be effectively integrated into the work practices of specific organizations. It goes beyond the focus on user interfaces. "Design for usability" includes the infrastructure of computing resources which are necessary for supporting and accommodating people as they learn to maintain and use systems. "Design for usability" encourages system designers either to accommodate to end-users' mix of skills, work practices, and resources or to try to alter them. Some HCI researchers recognize the need for further research to "raise the level of accountability of both management and usability people to meet human and organizational productivity goals [9]." This model addresses those issues as well as allowing for the altering of work practices to support effective use of a computer system in an organization. Further, we argue that this model is the most effective way for DL designers to ensure usability in DLs.

One means of ensuring usability in computer systems design is to apply a model of the organizational structure and examine how it fits with the intended computer system design. For example, web models of computing consider the social and technical infrastructures for supporting effective computer use in an organization [10, 11]. Web models make explicit connections between a focal technology (such as a digital library), its immediate users, and a rich ecology of social relationships with other social groups and organizations in which the technology is developed, adopted and used. Computer systems, in this conception, are treated as a form of social organization with important information processing, social and institutional properties: they are not only flexible information processing tools. Their capabilities and “fit” within organizations depend upon these social relationships as well as upon their information processing characteristics.

The evidence for the value of the organizationally sensitive "design for usability" model comes from a body of research and our own observations, which examine bottlenecks and failure points that prevent computer systems from being used. Research [11, 13] and observations show that organizations would get more from their original investment in computer systems design, if they assessed the social infrastructure for supporting computer use as well as specific tasks people need to use a particular system. Likewise, we believe that DL research and development communities could benefit from this approach.

In order to entourage the cultural constructs needed to design computer systems using the "design for usability" model, changes in behavior patterns and work practices must take place. Future research could provide guidelines to DL development communities for the adoption of the "design for usability" model of computer systems design.

### 7. Conclusion

We have introduced the term "organizational usability" to characterize the effective "fit" between computer systems (and DLs) with the social organization of computing in specific organizations. While designers may not know the specific organizations that will acquire their systems, they may learn key characteristics of typical organizations in their potential client community. For example, someone designing a DL for earth scientists is designing for an audience whose researchers are likely to have adequate access to needed platforms and available support staff with expertise. In contrast, someone designing DL services for elementary school children has client organizations where Apple IIs are commonplace, and where computing expertise is thinly spread. The same DL architecture which may add a nice database of space satellite videos to a geophysics lab may be far too costly, inaccessible and complicated for the grade school. These aspects of organizational usability go beyond the interface design.

In this paper, we have examined five models of computer-system design which are known in the information systems and computer science research and professional communities. Each of these models is a cultural model only in the specific organizations or professional communities where it is taken for granted as the proper way to design new systems. Cultural models are hard to change, because their assumptions seem so natural to members of the culture. We have characterized one design model which we believe is the dominant cultural design model in the DL research community. Each of the five design models has strengths and weaknesses. We proposed a new organizationally-sensitive model which we believe has the strongest chance of producing DL systems which many
people find to be routinely usable in their workplaces.

This is a good time for the DL research community to examine its own design models in light of systematic empirical research about the nature of usable computer systems and the particular organizational conditions under which people will often use DLs. It is possible to begin useful field studies which are based on DLs in use today. It is also possible to study the use of diverse DL resources, including OPACs, WWW, and e-journals, by faculty and students in research universities. We can start developing systematic understanding of the actual working conditions under which faculty, students, and staff find these resources most useful. We can also learn about the key bottlenecks that they face in a way that can help improve DL design methods. Empirical studies like these can also help systems designers have a better understanding of the social requirements for DL services and thus, they can better plan and develop effective systems. The time is ripe for DL designers to appreciate the importance of "design for organizational usability" to develop significantly more effective systems.

Acknowledgments

Thanks to Jonathan Allen and Lisa Covi for their suggestions on the content of this paper. We also appreciate the valuable comments from the anonymous referees.

References


The usability of a digital library primarily depends upon its user interface (UI) system. Designing effective user interface system for digital libraries is a complex task. User Interface Framework is helpful to design effective interface system, as it identifies and focuses on the various factors and limitations that influence the design of the system. Hence, this paper presents a novel user interface design framework for digital libraries based on user-centered design methodology and digital library user interface design principles, for its effective implementation. Keywords: User Interfaces