Enhancing Usability in CITIDEL: Multimodal, Multilingual, and Interactive Visualization Interfaces

Saverio Perugini, Kathleen McDevitt, Ryan Richardson, Manuel Perez-Quiñones, Rao Shen, Naren Ramakrishnan, Chris Williams, Edward A. Fox

Virginia Tech

Department of Computer Science

Blacksburg, Virginia USA 24061

1-540-231-6931

{sperugin, kmcdevit, ryanr, perez, rshen, naren, chwilli4, fox} @vt.edu

ABSTRACT

We describe four usability-enhancing interfaces to CITIDEL aimed at improving the user experience and supporting personalized information access by targeted communities. These comprise: a multimodal interaction facility with capability for out-of-turn input, interactive visualizations for exploratory analysis, a translation center exposing multilingual interfaces, as well as traditional usability enhancements. Pilot studies demonstrate the resulting improvements in quality, as measured across a number of metrics.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Graphical user interface, User centered design, Natural language;* H.5.4 [**Hypertext/Hypermedia**]: *Navigation*.

General Terms

Design, Human Factors.

Keywords

Interfaces, interactive visualization, community translation, out-of-turn interaction.

1. INTRODUCTION

The Computing and Information Technology Interactive Digital Educational Library (CITIDEL) was developed as part of the collection-building effort of the National Science Digital Library (www.nsdl.org). Essentially, it supports a collection of metadata about resources stored at sites such as ACM, DBLP, and NDLTD, in addition to having a small collection in its own right. As of January 1, 2004, CITIDEL had 12 source collections, and contained metadata on over 440,000 resources. During the year 2003, it received an average of over 33,000 hits per month, by users from at least 22 countries. With suitable enhancements, we believe CITIDEL can support a much wider base of users, and have significant impact on the teaching and learning of those interested in computing and information technology.

In this paper, we describe four recent interface enhancements to CITIDEL, all of which are aimed at improving usability and personalized information access to users. As the title of our paper suggests, these comprise a multimodal interaction facility with capability for out-of-turn input, a translation center exposing multilingual interfaces, interactive visualizations for exploratory analysis, as well as traditional usability enhancements. All of these enhancements are timely in that they address a growing community requirement and are concerted in that they are expected to become a permanent fixture in this and other digital library efforts. The enhancements described here are driven and evaluated by pilot studies with targeted users.

The software architecture of the present system relies heavily on the Open Archives Initiative, specifically its Protocol for Metadata-Harvesting (OAI-PMH) [8][21]. OAI-PMH allows willing archives to share metadata with a straightforward and easily implementable protocol. Within the framework of OAI, CITIDEL serves as a "service provider"; not focused on providing metadata (though it does so to NSDL, to support broader searches), but rather providing value-added services atop others' metadata. Thus, OAI provides a framework for modularity and enhancement, a central feature helping with integrating the usability enhancements described herein. Each of the following four sections presents one enhancement, and surveys pertinent work as appropriate.

2. Multimodal Interaction in CITIDEL

Our first usability enhancement to CITIDEL extends the interaction paradigm to support multimodal access, e.g., via voice. Besides improving accessibility, this enhancement enables *mixed-initiative interaction* between the user and the digital library.

Mixed-initiative interaction is a flexible dialog management strategy whereby the user and the DL can each change the flow of interaction. Browsing is typically a DL-initiated interaction, since the user merely clicks on presented hyperlinks. Voice interaction, however, can be both DL-initiated as well as user-initiated, since the user can either provide responsive or unsolicited input. It is this expressiveness afforded by voice that helps us achieve personalized interaction with the DL. When both the DL and user exchange initiative, we say that the interaction is a mixed-initiative interaction.

Mixed-initiative interaction is pertinent for DL interaction because it permits the user to supply *out-of-turn* inputs, namely values for facets that are not responsive to the DL's solicitation but nevertheless are pertinent to the information-seeking goals of the user. Intuitively, this form of access subsumes traditional browsing and provides the same form of customized interaction that is typically realized using cumbersome faceted browsing classifications [6].

Conference'04, Month 1–2, 2004, City, State, Country.

For example, as can be seen in Figure 1, CITIDEL “expects” the user to continue moving downward through ACM’s 1998 Computing Classification System. The user instead says “digital libraries” out-of-turn (through an interface mechanism described later), causing the interaction to “jump” directly to that part of the hierarchy. Without such capability, the user will have to manually traverse the presented hyperlinks to determine which of them index documents relevant to “digital libraries.” Keep in mind that, since out-of-turn interaction can be interleaved with browsing at the user’s discretion, the resulting interaction paradigm cannot be supported through traditional search interfaces.

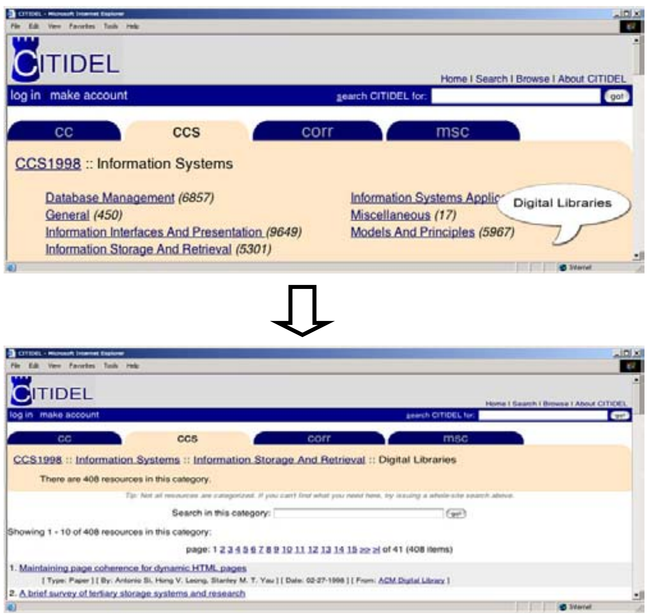

Figure 1. A mixed-initiative dialog via voice in CITIDEL to reach the webpage containing links to papers classified under "Information Systems :: Information Storage And Retrieval :: Digital Libraries”

A salient feature of our approach is the use of program transformations (e.g., partial evaluation) to *stage* the interaction [2]; this provides a functional approach to supporting dialogs, without explicitly manipulating state information.

To realize multimodal interaction, the DL needs interaction interfaces to capture input, a robust transformation engine, and a manager to mediate interaction and coordinate activities among the constituent sub-systems. The engine, which is deployed as a web service, is general enough to enable out-of-turn interaction with a variety of DLs without requiring the designer to anticipate the points at which out-of-turn interaction can occur. It handles in-turn (responsive) and out-of-turn (unsolicited) inputs (passed from interaction interfaces) in a uniform manner. The engine was designed using robust program transformation technology and implements a form of forward, followed by backward, slicing.

The transformation engine is a core sub-system, and besides applying the appropriate program transformations, conducts a whole host of other activities, including automatic query expansion. The architecture (see Figure 2) affords a customizable software framework for augmenting DLs with aspects of personalization, besides the support for mixed-initiative interaction [12]. Due to space considerations we only describe the interaction interfaces and manager below as they are the components primarily concerned with usability and interaction.

2.1 Interaction Interfaces

2.1.1 Extempore

The primary interaction interface for system is the Extempore toolbar [16], developed using the XML User interface Language (XUL) and JavaScript. This interface is designed to intelligently detect if a DL is capable of out of turn interaction, in which case it will appear within the browser, otherwise it remains hidden from the user, thus providing a non-invasive interface design. Through this lightweight, text-based interface, Extempore leverages users’ prior knowledge to provide a familiar and easy method of interaction. Another key feature about Extempore is that it exists not within the DL web pages, but in the web browser itself, making it a DL-agnostic interface.

2.1.2 SALTII

The SALTII interface, built using the SALT XML-based markup language, automatically embeds speech tags into HTML facilitating speech input and output to the DL (ref. Fig. 1). SALTII requires the SALT voice recognition plug-in for Microsoft Windows Internet Explorer 6.0. In using this interface, users could potentially carry out an entire dialog between themselves and the DL through speech alone, using speech not only for out-of-turn interaction, but in-turn as well.

2.1.3 Interaction Manager

The interaction manager is primarily responsible for coordinating communication between the transformation engine and the interaction interfaces, which, although not yet mentioned, also includes any web browser. Since there is no distinction between in-turn and out-of-turn inputs within the transformation engine it is desirable that the interaction manager also preserves this indiscrimination ability. We first outline the overall process by which interaction is established and managed (see Section 2.2.1), followed by descriptions of the three embodied subsystems (see Figure 2).

2.1.4 Preparing for Out-of-Turn Interaction

To situate the interaction manager as a dialog facilitator of both in-turn and out-of-turn inputs, we have investigated a variety of mechanisms, ranging from those that involve the full participation of the DL, to proxy-based bypass schemes. The former requires a DNS re-direct so that HTTP GET requests are forwarded to the interaction manager (notice that out-of-turn inputs are received directly from the interaction interfaces). This solution also has the attractive property that mixed-initiative interaction can be enabled at as fine or coarse a level of granularity as desired (e.g., it can be enabled for only certain subtrees). The proxy-based approach is a less configurable solution and must be targeted carefully, to avoid loss of functionality. We adopt the former approach. Once

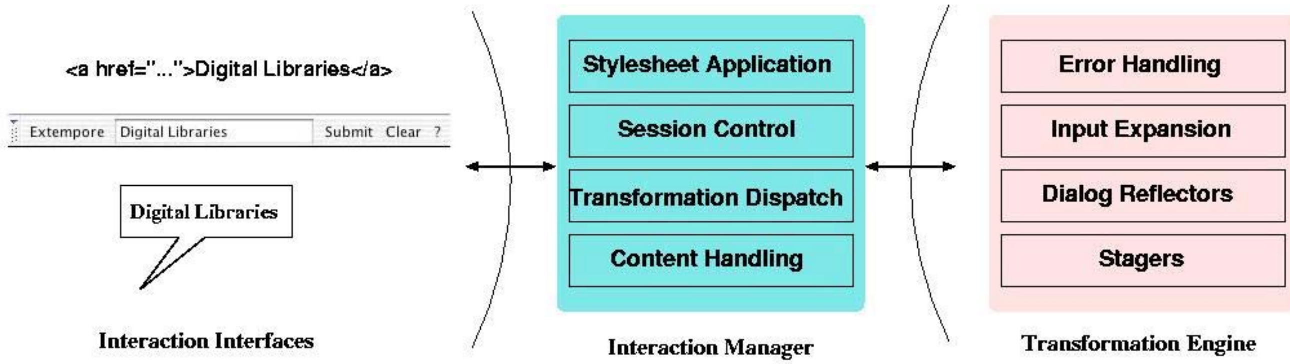

Figure 2. Multimodal DL interaction framework architecture.

such an initial handshake is established, the interaction manager is responsible for providing concurrent access to the transformation engine, from multiple interaction interfaces.

The interaction manager, now placed in the loop, evaluates if out-of-turn interaction is possible, activates the interaction interfaces as appropriate, and mediates all interactions from this point. Notice that intermediate dialog states might not correspond to any of the DL's existing pages (especially after some out-of-turn interaction), so the interaction manager must mediate the dialog to the fullest.

2.1.5 Content Handling

Content handling determines the feasibility of out-of-turn interaction, caches dialog states, and ensures currency of the DL browsing representations. It also is responsible for retrieving, caching, and updating content from websites. To determine the feasibility of out-of-turn interaction, the content handler uses a simple Representation State Transfer (REST) request for a document named 'content.xml' located at the DL's root web directory. This document is meant to supply the representation of the DL's browsing sequences, and when annotated with stager tags, helps initialize the dialog representation. If the content file is not present, it indicates that the DL is not capable of out-of-turn interaction through this system. The content file is then cached in a local database for fast transformation computations. It also will send a trigger event that will initiate the activation of the Extempore toolbar or the SALT tags, as appropriate. From this point, the content handler is responsible for ensuring the currency of the representation and updating the file as appropriate.

Interaction state caching, also managed by the content handler, is trivially implemented by associating intermediate dialog states with content files generated over the course of an interaction. A more sophisticated solution is to develop a caching policy that exploits the structure of program transformations. For instance, if a user is requesting a partial evaluation with regard to 'digital library multilingual,' but the cache only contains a document that has been evaluated with regard to 'digital library', we can partially evaluate this document internally with regard to the remaining input (namely, 'multilingual'), thus removing the need to partially evaluate from the root document. While reducing storage complexity, this approach also creates interesting design tradeoffs (including concerns about session and user security).

2.1.6 Transformation Dispatch

The transformation dispatch is responsible for handling communication with the transformation engine. It handles connecting to the transformation engine as well as notifying the interaction interfaces if such a connection cannot be made. This notification is done to prevent users' expectations for interaction being invalidated due to system error. (When the transformation engine receives partial input, recall that it does not know, or need to know, whether the partial input is a result of browsing or of supplying some information out-of-turn.) Finally, transformation dispatch supports the marshalling and un-marshalling of transformation requests into Simple Object Access Protocol (SOAP) messages, as well as the transmission and reception of those messages across HTTP in accordance with the SOAP specification.

2.1.7 Session Control

Session control has a responsibility that differs from most other web systems' concept of session management. The notion of 'state' within a dialog is just its representation, since it succinctly summarizes all remaining dialog options. Furthermore, the transformation engine does not explicitly manipulate state and is therefore a purely functional entity. Thus the goal of this system's session control is merely to enumerate one user's interaction session from another. Due to our uniform handling of in-turn and out-of-turn inputs and the variety of interaction interfaces available, session tokens, in this case a ten decimal digit session identifier, are stored in multiple different places, one in each interaction interface and a final in the browser itself. This multi-headed session format negates the application of most modern session management packages, which are primarily concerned with tracking browsing interactions. Session control was specifically designed to handle this issue as well as to handle the normal session management issues (e.g., back button browsing and threaded browsing).

2.1.8 Stylesheet Application

The stylesheet application is responsible for transforming the information returned from the transformation engine into the visitor's native presentation format. In order to do this, the stylesheet application first must identify the visitor's browser capabilities and determine how best to style the content. Notice that the stylesheet application evaluates and potentially utilizes all of the presentation styles, instead of just picking a "safe" presentation style, like HTML. In addition, the stylesheet application might have to define and introduce suitable grammar tags into the HTML page (for the voice interface), accomplished by analyzing the remaining dialog options. Currently we support [x]HTML, WML, SALT, SVG, or any XML-based presentation format, but the framework for the stylesheet application is flexible to support almost any presentation format.

2.2 User Studies

User experiences with out-of-turn interaction are described in [14]; 25 users were given information-finding tasks and were free to use either in-turn or mixed-initiative interactions to complete these tasks. Some of these tasks were *non-oriented* (meaning they could be performed with browsing alone, if desired) and some were *out-of-turn-oriented* (meaning they would be cumbersome to perform via plain browsing). We found that 100% of the users utilized the out-of-turn interfaces when presented with an out-of-turn-oriented task. Since the task type was not disclosed *a priori*, this result demonstrates that users are adept at discerning when out-of-turn interaction is desirable. Extempore and SALTII interfaces were utilized equally effectively.

3. VISUALIZATION SYSTEM

Our next usability enhancement involves building visual interfaces to CITIDEL, an idea promoted in many other DL projects [20][23]. Visualization techniques of one important class are those visualizing document attributes that are predefined (author or date) or computed (e.g., query relevance). One example is the Envision system [13] and the enhanced version of Envision [24]. Predefined categories to which documents belong also can be considered as predefined document attributes or as semantic information. Cat-a-Cone [5] assists users by displaying semantic information (categories assigned to each document). Another class of visualization techniques do not make assumptions regarding document attributes but automatically derive a collection overview via the use of unsupervised learning, such as variants of document clustering. Examples are Scatter/Gather [3], Grouper [18], and ThemeScapes [26].

Like the above projects, the framework of our visualization system is component-based (in XML) and involves a hyperbolic tree of a hierarchical concept map and a 2D scatter-plot graph. The system architecture is shown in Figure 3, where the data source and clustering components are implemented and wrapped into Java servlets to enable web access. The visualizing components communicate with these servlets via XML messages. Data source components submit a user query to CITIDEL and other DLs, and parse the retrieved HTML pages into XML files conforming to XML schemas we developed. These XML files are then input to clustering components to be processed, which are basically implementations of different document clustering algorithms. We incorporated the clustering

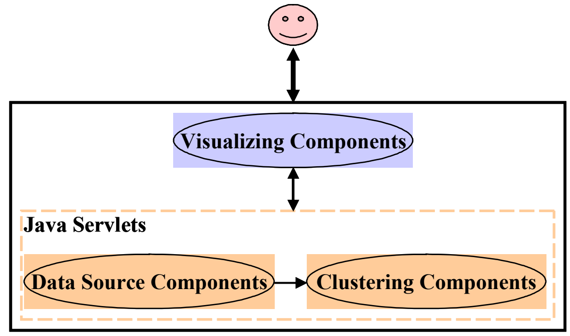

Figure 3. Visualization system architecture

components of the Carrot2 system [25] for the work presented here.

3.1 Visualizing component

The initial interface is depicted in Figure 5. On the top left of the screen is a hyperbolic tree view of the ACM Computing Classification System [1998 Version] (CCS1998, http://www.acm.org/class/1998/). On the top right is a query box. By default, a user will retrieve results from a member (source) DL (here, ACM DL) of CITIDEL. A user also has an option to retrieve results from all CITIDEL member DLs. The middle of the screen depicts a 2D scatter-plot graph. At the bottom, there are fields for the details, represented by way of values of various attributes of a selected document. The visualizing components support exploration, browsing, and analysis. The following examples illustrate the types of insights afforded, for the following three example scenarios.

Figure 4. Clustering interface

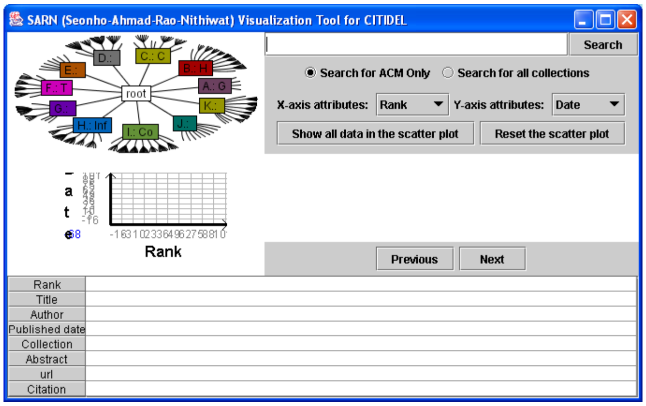

Figure 5. Initial Interface

Example of Insights:

1. How are the retrieved documents clustered according to the ACM Computing Classification System?
2. How are the retrieved documents clustered according to inter-document similarity?
3. Which cluster has the largest document collection?
4. To what category does the 1st ranked document belong?
5. Which document is cited most among the selected clusters of documents?
6. Which documents cite a selected document?
7. What's the most recently published paper by an author?

Scenario 1: Exploring retrieved results from ACM DL

A user enters the query: 'Information Visualization'. By default, she gets results from the CITIDEL member DL named 'ACM DL'. A hierarchical concept map organized by the ACM Computing Classification System is then displayed as a hyperbolic tree on the top left of the screen. Node name represents a category and a bubble attached to a node represents a document collection belonging to that category. Bubble size attached to a node indicates the size of the document collection clustered to that category. The hyperbolic tree supports 'focus+context' navigation.

After the user clicks the 'Show all data in the scatter plot' button, all the retrieved documents from the ACM DL are scatter plotted in the 2D space as shown in Figure 6. Each document is visually mapped to a dot. On the right of the screen, there is a list of colorful bars representing the categories to which those retrieved documents belong. Moving the mouse over a bar invokes animation of blinking dots in the 2D scatter plot space. These blinking dots represent documents belonging to the category visually mapped to a colorful bar pointed to by the mouse. The dots are in the same color as the bar and the corresponding bubble in the hyperbolic tree. A user can change the color of a bar to distinguish different categories.

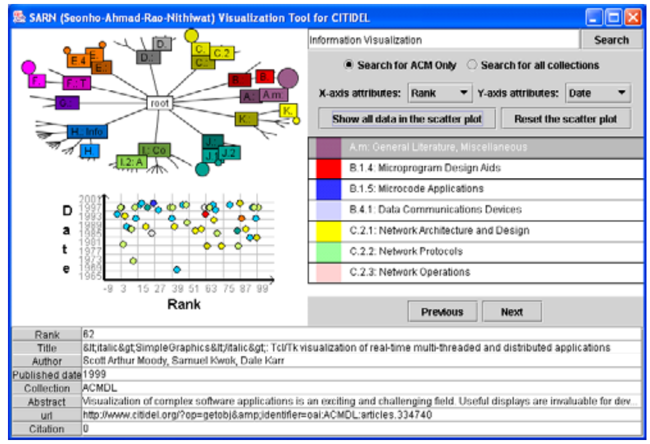

Figure 6. Visual Results of Scenario 1.

This will change the color of the corresponding dots in the 2D space and the corresponding bubble in the hyperbolic tree. Dots in the 2D space can be arranged according to attributes of rank, date, and citations. Clicking a dot causes animation of the shining dot. The details of the selected document are then displayed at the bottom of the screen.

Scenario 2: Show me papers by 'Chris North' from CITIDEL

A user inputs the query 'Chris North' and selects the option 'Search for all collections'. The retrieved results from CITIDEL are then clustered via cluster components and displayed as a hyperbolic tree. The user navigates the hyperbolic tree and finds a category named 'Visualization Displays' as her interest. She then clicks the green bubble attached to that interesting category. This causes all the eight documents belonging to this cluster to be plotted as eight green dots in the 2D scatter plot space shown in Figure 7. She continues to browse the hyperbolic tree and finds another interesting category named 'Case Study of the Visible'. She clicks the yellow bubble attached to the category named 'Case Study of the Visible', which causes three yellow dots to be plotted in the 2D space as shown in Figure 8.

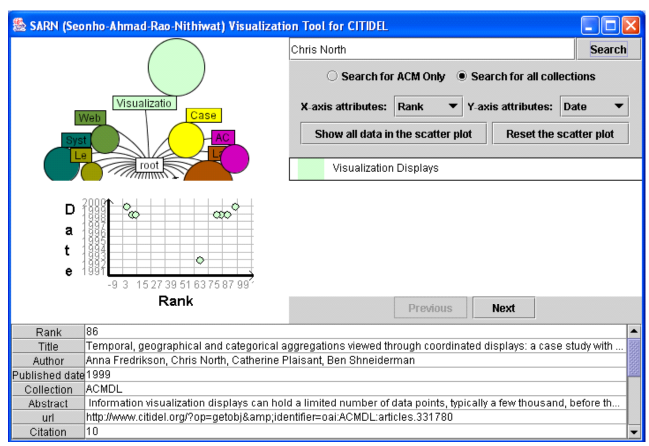

Figure 7. Scatter Plot Documents in One Cluster.

Scenario 3: Show me papers written and cited by 'Edward Fox' from ACM DL

A user inputs the query 'Edward Fox'. By default, she gets retrieved results from the CITIDEL member DL named 'ACM

DL'. After she clicks the "Show all data in the scatter plot" button, all the retrieved documents from ACM DL are scatter plotted in the 2D space as shown in Figure 9. When she clicks a dot representing the document with the title 'toward a worldwide digital library', a link pointing to a dot enclosed in a square is dynamically created on demand. She then follows this link to get details of the cited paper authored by Edward Fox.

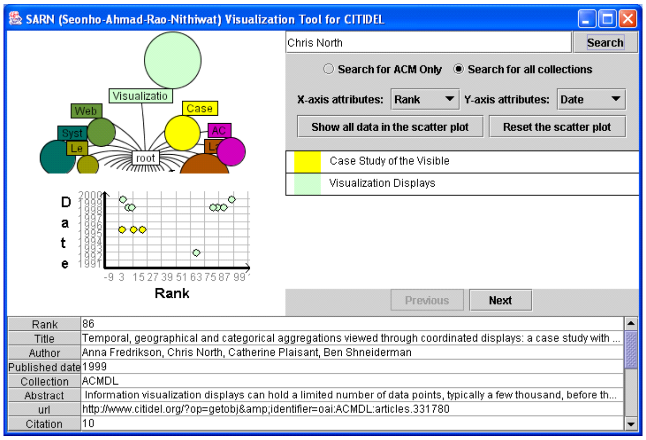

Figure 8. Scatter Plot Documents in Two Clusters.

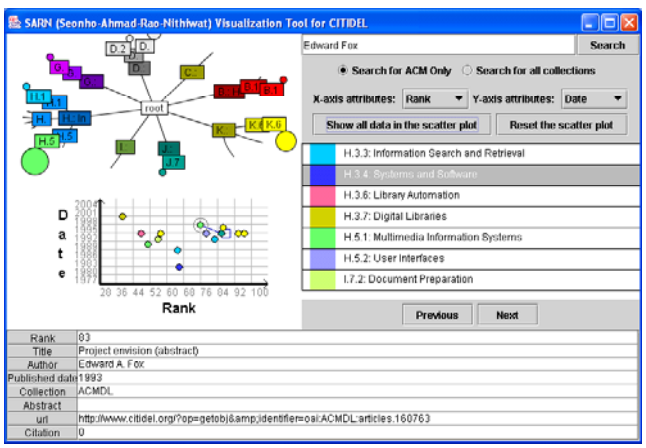

Figure 9. Show Citation in Scatter Plot Graph.

3.2 Pilot User Study

We conducted pilot user studies for both the clustering interface shown in Figure 4 and our interactive visualization system. We found that less time was spent locating a given document via the clustering interface than via traditional flat lists. We asked users the last three questions of the example insights mentioned in Section 3.1, and found that it was relatively easier for them to answer the questions using our visualization system versus the original CITIDEL interface [1].

4. GENERAL USABILITY ENHANCEMENTS

4.1 Initial User Study

To improve the usability of CITIDEL, a focused user study was conducted in late 2002 through early 2003. Data was gathered from 50 respondents, which consisted of 41 students, 7 professors, and 2 who did not identify themselves. Users were asked to perform certain tasks such as searching, browsing, and creating an account. They were asked to rate their effectiveness at finding documents and other related tasks. Users also were requested to provide feedback on what they found helpful or confusing about the system. When asked to provide an overall rating of the CITIDEL system, 74% rated it as 'above average'. However, many issues were discovered. Table 1 gives a summary of some of the most interesting points and how we addressed them.

4.2 Discussion

One interesting point raised by users was their unfamiliarity with the classification schemes supported by CITIDEL (ACM/IEEE Computing Curricula 2001, ACM Computing Classification System [1998 version], Computing Research Repository (CoRR) Subject Areas, and AMS Mathematics Subject Classification [2000 version]). Given that few educators would be familiar with more than one of these, it is not surprising that students would find them daunting. In fact, based on browse logs, 64% of all browse events were for MSC2000, 26% were for CCS1998, with the other two schemes receiving very few browse events. The finding that MSC2000 was the most popular is surprising given that CCS1998 is CITIDEL's default classification scheme. Users can personalize their browsing experience, however, and set the default to any one of the schemes. To educate users on the less commonly used schemes, we added links from the main browsing page to an explanation of these classification schemes, summaries of each scheme, and to the hierarchies themselves.

Also, users seemed unaware of the breadth and depth of resources CITIDEL contains. To improve this, we added a 'Pseudo-Random Resource' to the main page, which randomly shows an image from an interesting resource in CITIDEL (see Figure 10).

Table 1. Summary of significant usability issues that we addressed:

Section	Issue	How it was resolved
Main	Users want a sample of what is in the digital library.	Added an image of a pseudo-random resource to the front page, that links to an interesting resource.
Browse-by-category	Users are not familiar with ACM, IEEE-CS, and the various chosen classification systems.	Included more information about the classification schemes, including a link to details of each scheme from the main browsing page.
Search	Users did not understand that they could add items they had found to their binder.	Reworded and reordered the page to make this more clear.
Search	Users want to know how much of a resource is	Not possible to determine this at present. Since metadata points to other

	available (e.g., abstract, full-text, etc.).	sites, we would have to poll these sites periodically. An extension to the OAI protocol might handle this. Left as future work.
Search	Users were unclear about the different uses of various filters, and when they were being applied.	Changed wording so that users would understand when documents were filtered due to not matching the category and when they were filtered due to the user's preferences.

A more fundamental problem is that it appeared that many users did not understand the distinction between metadata and data, and that they were unclear when they were actually in CITIDEL (working with metadata), and when they had followed a link that took them to another site (working with target data, such as resources). To address this issue, we re-designed all CITIDEL pages to have a consistent look, and supplied visual clues to users to distinguish the case when they were in the CITIDEL domain from others.

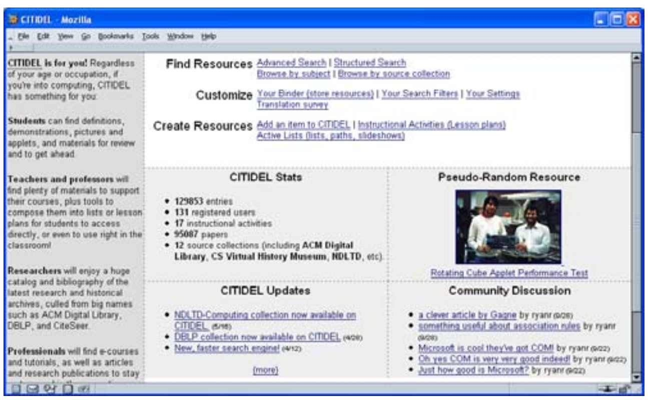

Figure 10. Screenshot of part of the main page of CITIDEL, showing 'Pseudo-Random Resource'.

The above issue led to another request by users. Many users wanted CITIDEL to show how much of a resource is available, such as abstract-only, an extended summary, or full-text. For the vast majority of the resources, CITIDEL only has metadata about resources and not the resources themselves. CITIDEL provides links to the resources that are found on many other sites. There is currently no practical way for CITIDEL to keep track of how much of a resource may exist on another site, since these sites will change over time. Perhaps such information could be gleaned as part of an archiving effort, like that undertaken for NSDL by SDSC. Alternatively, an extension to the OAI Protocol for Metadata Harvesting, like the one described in [19], could be employed to periodically poll the external collections; this approach is outside the scope of our project.

4.3 Future Work

In addition to the possible future work described above, some users indicated that they would prefer to browse the hierarchies via an interface similar to Microsoft Windows ® Explorer. This is a style many are familiar with and would allow them to more quickly understand the hierarchies, and to see the depth of the hierarchy with just a few clicks. We will evaluate if such modifications to the browsing style will be worthwhile.

5. COMMUNITY MULTILINGUAL INTERFACE

Providing a multilingual interface is one way to expand the user population of a digital library. In our case, we wanted to accommodate users who are more comfortable using languages other than English to access CITIDEL. Because most of the content in CITIDEL is harvested from other digital libraries, we have little control over the language used in many resources. However, CITIDEL also allows users to add their own resources that may be in other languages and create content in the form of lesson plans, and online discussion forums. We also are planning on harvesting collections from other countries where English is not the first language (e.g., resources in Portuguese from Brazil). Thus, it is expected that CITIDEL will have many different users who might be more comfortable using other languages to not only access the resources in this digital library but possibly also contribute their own resources to be used by others.

To support multilingual access to CITIDEL, we decided to provide the CITIDEL interface itself in multiple languages. The initial implementation focused on providing a Spanish interface, mostly to support Hispanics in the USA. But the system here described has no restrictions in implementation that would exclude other languages and has since been expanded to support six different languages.

5.1 Approaches to Multi-lingual Interfaces

There have been many efforts used to translate interfaces to technical sites. Several different approaches are popular, and quite a number of tools to make translating easier have been developed. To successfully cross the language translation divide, Dilts [4] presents many suggestions for both writing clear, translatable content as well as choosing a suitable translation service. Among these are leaving proper space for translations in languages which require more space, and providing translators with definitions of computer industry jargon.

We explored three approaches to interface translation. These are: machine translation, human translation, and community-based translation. The following three sections briefly describe what these are and provide our evaluation of the quality of the translation by each of these methods.

5.1.1 Machine Translation

One approach is via an automated translation program, though the quality of these results can vary greatly. This is not to say such systems are without some benefit. There is, however, still a long way to go before the best quality machine translation systems are readily available, and, moreover, are readily usable for specialized purposes such as web sites, which often contain computer jargon. Tucker classified machine translation systems into three categories: first-generation, second-generation, and third generation [22]. First-generation systems give rough translations, mostly based on word-to-word mappings, independent of the context of use. Second generation systems have an interlingua (language-free) approach and have transformational grammars and/or ATN-type recognizers that aid in the process. Third-generation systems have much more understanding built in, and thus produce higher quality results.

A popular example of a machine translation system is AltaVista's Babel Fish Translation [1] in which users may receive translations for words, sentences, or paragraphs by entering these directly, or translating a full web page by providing a URL. Babel Fish is based on a product by Systran[2]. It is a first-generation translation system, and is therefore rough grammatically, with no semantic understanding capabilities of structure and connections of the text being translated. However, it does afford quick and free translation.

5.1.2 Human Translation

Another approach for translation is for the designer or administrator of a site to translate the interface. The problems that can arise in this approach are discussed well in "Internationalizing Online Information" by Merrill and Shanoski [11]. It is common for developers to either translate their own sites or hire special translation teams. Sites like CNN and Yahoo take this approach. Often this results in translations in good time and of good quality. However, it is a costly approach especially when site content changes quickly.

5.1.3 Community-Based Translation

A third option, and the one we followed for CITIDEL, is to appeal to a site's user-base and allow this community of volunteers to translate the interface. This approach has been successful in translating online newsletters (e.g., TidBITS[3]), desktop applications (e.g., OmniWeb[4], Translation Project[5]) and even entire websites (e.g., 'Google in your language' beta translation initiative[6]). The key to this approach lies in the willingness of the online community of users to volunteer their time and expertise to create quality translations.

5.1.4 Evaluation of Approaches

We conducted a brief evaluation of the three approaches above, to assess the quality of the translation. The details of the evaluation [10] are beyond the scope of this paper. We present a brief summary of the findings here. The benefit of this particular study was to have a principled way to evaluate the quality of the resulting language from the three approaches. The pages that were evaluated were all of a technical nature. They were translated by automated programs, by developers, and by community. We developed a rubric and used it to evaluate a group of pages. A maximum score of 13 was possible; the higher the score the better the translation. The quality of the translation took into consideration: language structure, vocabulary (cognates, meanings, and spellings), style, and message. The following pages were evaluated:

- Commercial website (Apple's web page about their Mail program that is included in OS X) in English from their US web page and in Spanish from their Latin American page. These Spanish pages were translated by Apple's internal personal.
- Technical online newsletter page (TidBITS) in its original English form and the same page translated by a community of volunteers)
- Two English CITIDEL pages.

We also translated all pages to Spanish or English (depending on the original form) using AltaVista's Babel Fish. The results of this evaluation were as expected: human translation and community translation produced higher quality translations than machine-translations (for the particular software used).

None of the pages translated by Babel Fish received higher than a 2. The pages translated by the community (TidBits) or by a development team (Apple's Mail page) received perfect scores. In the case of Apple's page, the translation is even better than is possible with the machine-translation because they made use of images that contain text in them and these cannot be translated by a system like Babel Fish. Furthermore, this site used some marketing slogans regarding the Mail's Junk Mail filter that could only be translated properly with human intervention. For example, the Spanish ("Correo basura a la basura") and English ("The end of junk mail") versions of the slogans were very different, though conveying the correct message in both languages.

From this simple evaluation, it is clear that the automated translation services provided by one of the common services available on the web are not sufficiently ready for deployment in CITIDEL.

5.2 Design of the Translation Center

The main goal of the translation center (see Figure 11) was to create a center that could foster the creation of an online community helping with translation. An online community has many definitions; one that we will use for the following discussion is from Preece [17]: In general, an online community is a computer-supported place where people interact socially with a common/shared purpose and voluntarily following a series of norms, protocols, and practices.

The design of a user interface that would foster an online community should go beyond providing user-tasks for accomplishing the basic functionality the computer application is intended to accomplish. It should provide functionality that increases the sense of membership in the community, functionality that keeps members of the community informed about what others are doing, and enforces the will of the community over the individual needs/desires of particular users without imposing too many norms/protocols.

To that end, we studied online communities, and conducted several online surveys to identify the functionality that was needed in our translation center. We identified and implemented 23 different tasks, covering five different categories: informative, assistance, community, encouragement, and task-oriented. We put as much effort into building functionality that supports the online community as we did into building functionality that supports the translation task.

5.2.1 Architecture

The Translation Center was built using Java Server Pages. The focus of the software architecture was to provide a way to translate CITIDEL pages without changing the layout of the

[1] http://babelfish.altavista.com

[2] http://www.systransoft.com/

[3] http://www.tidbits.com/about/translations.html

[4] http://www.omnigroup.com/applications/omniweb/

[5] http://www.iro.umontreal.ca/contrib/po/HTML/index.html

[6] http://services.google.com/tc/Welcome.html

interface. We believe that the architecture could be easily extended to support language specific layouts (outside the scope of the present effort) .

Each page is formed from an HTML layout and a series of database retrievals for "items", which are words or phrases used in the interface. The key to retrieving each translation is a unique number associated with each English "item" and a language tag representing the language being viewed. The whole CITIDEL site then becomes a series of HTML pages with embedded JSP code to retrieve the particular words to be used. The length or complexity of the "items" depends on the content of the page, and not on the translation center.

The Translation Center works independently of CITIDEL, with a common backend of "items" and translations. The Translation Center provides the functionality that allows volunteers to retrieve "items" to be translated and provide translations, as well as rating different translations for quality and correctness.

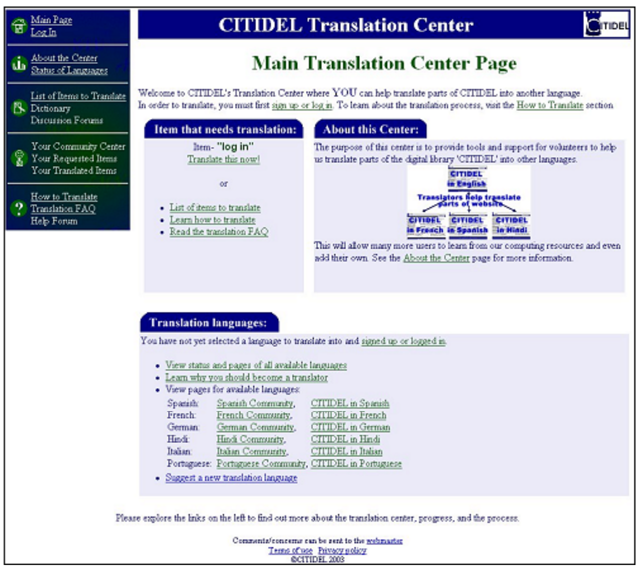

Figure 11. Screenshot of main page of the Translation Center.

5.2.2 *Interface Design*

Figure 11 shows the main page design of the Translation Center. Volunteers must register with the center before they are allowed to translate items from the CITIDEL interface. The main tasks that volunteers do are to translate items and to rate other item translations.

A priority of translation is calculated for each item. Those items that have no translation are given high priority. The rating of an item identifies good translations and poor translations. As items can be re-translated many times, each translation has a separate rating and the one with the highest rating is displayed on the CITIDEL page. Additionally, shared vocabulary tools and conflict resolution tools are provided to assist in deciding which version appears. Also each CITIDEL page has a link at the bottom that allows CITIDEL users to rate the quality of the overall page translation. Overall, the goal is to make it very easy for volunteers to select an item to translate which will have maximum impact on the site.

We conducted a multi-step evaluation of the Translation Center. The details of the evaluation are beyond the scope of this paper. Briefly, we evaluated the interface of the translation center itself, the sense of community that volunteers perceived, and the quality of the translated page. Overall, our system was rated very successful in its design and implementation. In a limited testing environment without the needed community to provide checks and balances, one half-translated page was rated a 5 on the 13 point scale. This far surpasses the rating of translations done by machine translation, and is well on its way even at an early stage to the perfect score achieved by the other community translation projects. We are now ready to deploy it together with CITIDEL and assess how well it helps a community of volunteer translators develop.

5.3 Future Work

To truly internationalize a website, it not only needs to be *readable* to users of another language but *understood*. In many cases, it is not simply enough to translate words into another language. Languages with scripting systems that read from right to left, for example, should have layouts and menus adjusted accordingly when translated from English. Furthermore, as Marcus and Gould suggest, interfaces for countries of different cultures should look and feel very different [9]. Although our implementation supports having multiple layouts, our Translation Center currently only supports translation of content keeping a single layout. As the international community of our system grows, we will be adding functionality to the Translation Center to support layout changes on a language-by-language basis, based on community desires and needs.

6. DISCUSSION

This paper has described four kinds of enhancements to the usability and functionality of CITIDEL, a large DL effort to promote teaching and learning in computing and related areas. Each type of enhancement has been designed, implemented, and improved through user studies. Our modular architecture has made that possible, and should allow similar enhancements to be applied to other DLs with minimal effort.

We are continuing our research and development efforts with CITIDEL on many fronts, including more complex visualizations, inclusion of concept maps in multiple languages, and more powerful interaction techniques.

7. ACKNOWLEGEMENTS

This research work was funded in part by the NSF through grants DUE-0136690, DUE-0121679 IIS-0086227, and IIS-0136182. We would like to thank ACM, DBLP, and others for sharing their content and classification schemes with us. We particularly want to thank Aaron Krowne, now at Emory University, for his support of prior work on CITIDEL. We also thank colleagues in the Digital Library Research Lab at Virginia Tech, as well as colleagues at Villanova University, Hofstra University, Penn State University, and the College of New Jersey for their useful suggestions.

8. REFERENCES

[1] Abdelmoumin, G., Alafaliq, A., Kampanya, N., and Kim, S., project report for CS5764 at Virginia Tech, avail. at http://rocky.dlib.vt.edu/%7Ershen/Research/Viz/CS5764_Project.html, Dec. 2003

[2] Capra, R. Narayan, M., Perugini, S., Ramakrishnan, N., and Pérez-Quiñones, M.A. The Staging Transformation Approach to Mixing Initiative. In *Working Notes of the IJCAI 2003 Workshop on Mixed-Initiative Intelligent Systems* (Acapulco, Mexico, August 2003, AAAI/MIT Press, 2003, 23-29.

[3] Cutting, D., Karger, D., Pedersen, J. and Tukey, J., Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, 1992, 318-329.

[4] Dilts, D. Successfully Crossing the Language Translation Divide, (ACM) *Proceedings of the 19th annual international conference on computer documentation*, 2001, 73-77. Available at: http://doi.acm.org/10.1145/501516.501531

[5] Hearst, M.A. and C., K., Cat-a-Cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, 1997, 246-255.

[6] Hearst, M. A., Elliott, A., English, J., Sinha, R., Swearingen, K., and Yee, K-P. Finding the Flow in Web Site Search, *Communications of the ACM*, 45, 9 (Sep 2002), 42-49.

[7] Krowne, A. and Fox, E.A. An Architecture for Multischeming in Digital Libraries. In *Proceedings of International Conference of Asian Digital Libraries 2003*, 2003, 563-577.

[8] Lagoze, C. and Van de Sompel, H.. 2001. The Open Archives Initiative: Building a low-barrier interoperability framework. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, Roanoke VA, June 24-28, 2001, 54-62.

[9] Marcus, A. and Gould, E.W. Crosscurrents: cultural dimensions and global Web user-interface design. *ACM Interactions*, Volume 7, Issue 4, July 2000.

[10] McDevitt, K., Pérez-Quiñones, M. A., and Padilla-Falto, O. I. "Design of a Community-based Translation Center", Technical Report cs.HC/0401007, Computing Research Repository (CoRR), Jan 2004, available at http://arxiv.org/abs/cs/0401007/

[11] Merrill, C.K. and Shanoski, M, Internationalizing Online Information. SIGDOC, ACM Press, 19-25, 1992.

[12] Narayan, M., Williams, C., Perugini, S., and Ramakrishnan, N. Staging Transformations for Multimodal Web Interaction Management, in *Proceedings of the ACM International World Wide Web Conference (WWW'2004),* New York, NY, May 2004, to appear.

[13] Nowell, L.T. Graphical Encoding for Information Visualization: Using Icon Color, Shape, and Size to Convey Nominal and Quantitative Data. Ph.D. dissertation, Virginia Tech, Blacksburg, 1997, available at http://scholar.lib.vt.edu/theses/delayed/etd-111897-163723/etd-title.html.

[14] Perugini, S., Pinney, M. E., Ramakrishnan, N., Pérez-Quiñones, M. A., and Rosson, M. B. Taking the Initiative with Extempore: Exploring Out-of-Turn Interactions with Websites. Technical Report cs.HC/0312016, Computing Research Repository (CoRR), 2003.

[15] Perugini, S. and Ramakrishnan, N. Personalizing Interactions with Information Systems, in Advances in Computers, edited by M. V. Zelkowitz, Academic Press, 9,57, 323-382.

[16] Perugini, S. and Ramakrishnan, N. Personalizing Web Sites with Mixed-Initiative Interaction. *IEEE IT Professional*, 5, 2, (Mar 2003), 9-15.

[17] Preece, J. *Online Communities*, Designing usability and supporting sociability, New York: John Wiley & Sons, 2000.

[18] Rennison, E., Galaxy of news: An approach to visualizing and understanding expansive news landscapes. In *Proceedings of UIST'94, ACM Symposium on User Interface Software and Technology*, (New York, 1994), 3-12.

[19] Sanchez, J.A., Nava, S, Fernandez, L., and Chevalier, F. Distributed Information Retrieval from Web-Accessible Digital Libraries using Mobile Agents. *UPGRADE*, Vol.III, No. 3, June 2002, 37-43.

[20] Shen, R., Wang, J., Fox, E.A. A Lightweight Protocol between Digital Libraries and Visualization Systems. *Proceedings of Visual Interfaces to Digital Libraries 2002*: 217-225

[21] Suleman, H. A Framework for Building Open Digital Libraries. *D-Lib Magazine* 7(12), December 2001.

[22] Tucker, A.B. A Perspective on Machine Translation: Theory and Practice. *Communications of the ACM*, Volume 27, Number 4, 1984, 322-329.

[23] Wang, J. VIDI: A Lightweight Protocol between Digital Libraries and Visualization Systems. Master's Thesis, Dept. of Computer Science, Virginia Tech, 2002.

[24] Wang, J., Agrawal, A., Bazaza, A., Angle, S., Fox, E.A. and North, C., Enhancing the ENVISION interface for digital libraries. In *Second ACM/IEEE-CS joint conference on digital libraries*, (Portland, Oregon, USA, 2002), ACM Press, 275-276.

[25] Weiss, D. *A Clustering Interface for Web Search Results in Polish and English*, Master's Thesis, Poznan University of Technology, Poznan, Poland 2001.

[26] Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M. and Schur, A., Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proceedings of the Information Visualization Symposium 95*, (1995), IEEE Computer Society Press, 51-58.

[27] Zamir, O. and Etzioni, O., Grouper: A Dynamic Clustering Interface to Web Search Results. In WWW8 / Computer Networks, 1999, 1361-1374.