Creating an Instant Messaging Reference System

Jody C Fagan, James Madison University
Michele Calloway
Creating an Instant Messaging Reference System

Jody C. Fagan  
*James Madison University, faganjc@jmu.edu*

Michele Calloway

Follow this and additional works at: [http://commons.lib.jmu.edu/letfspubs](http://commons.lib.jmu.edu/letfspubs)

Part of the [Library and Information Science Commons](http://commons.lib.jmu.edu/letfspubs)

**Recommended Citation**

Creating an Instant Messaging Reference System

Jody Condit Fagan and Michele Calloway

Libraries are expanding digital reference services to include chat and instant messaging. Various companies have developed systems for purchase that meet this need without requiring users to download new software or learn a new skill. These systems share many features in common, including transcript storage, sending URLs, and pushing Web pages. Several librarians at an Association of Research Libraries academic library wished to test instant messaging reference with the ability to control and experiment with different implementations of these features. This article describes the features of instant messaging reference systems and the in-house development of a system that incorporates them.

Online reference, specifically the flavor known as chat or instant messaging reference, is definitely a hot topic. At the American Library Association (ALA) 2001 Annual Conference, the topic cropped up in poster sessions, in programs, at the exhibits, and in planning committees. There are many resources available for libraries interested in starting online reference services, including bibliographies, registries of services, and electronic discussion lists. Morris Library at Southern Illinois University–Carbondale has offered digital reference through e-mail and online Web form for about three years. Use has increased dramatically each year; in FY00, it received only 348 questions, while during FY01 it received 629 questions. Some of the increase in digital reference traffic is credited to the redesign of the library's home pages, in which the e-mail service was made more prominent. Also during the redesign, the library implemented a Web site search that categorizes results by resource type and ranks items by librarian-assigned metadata. While the site search was a popular tool, library patrons used it ineffectively, with over half the searches getting no matches. Many were typing in specific topics as if they expected books and articles to magically appear. Their desire for reference assistance on the Web was an inspiration to expand services.

During a study of the site search in which transactions were logged, the number of searches approached the number of desk reference questions library wide. This made it clear that some patrons wanted to get help from their workstations, and they wanted to use a tool that provided the advantages of site search. We hypothesized that the simple input box was the attraction of site search. It delivers instant results and is (seemingly) anonymous. Instant messaging services have these same advantages.

The usage statistics from libraries currently using instant messaging reference show patrons are beginning to show interest, but not at an overwhelming rate. Table 1 shows statistics for several libraries that were reported at the ALA 2001 Annual Conference poster sessions.

We wanted a system that would take advantage of the success of site search, which we felt was due to its up-front implementation at the top of our Web pages, its speed, and the implicit promise of instant results. In a digital reference survey by Joseph Janes, only half of the one hundred fifty libraries surveyed linked to their digital reference (e-mail or Web form) from the library home page. Ease of use and access were of top importance; this meant simplicity, high speed, and no technical difficulties.

What's Out There: Library Instant Messaging Reference Systems

After deciding to try out instant messaging reference, we needed to find out what software was available, what features were offered, and how well different systems met our requirements. Steve Coffman from Library Systems and Services (LSSI) says there are “fifty or more vendors developing software with varying degrees of sophistication at present.” Products to be evaluated for this study were selected by looking at the archives of the Dig-ref electronic discussion list and LiveRef: A Registry of Real-Time Digital Reference Services by Gerry McKiernan. From these sources three products were selected for review: Live Assistance, LivePerson, and Virtual Reference Desk (VRD) (LSSI). Each of these companies' Web sites was visited and their chat demonstrations were evaluated. Transcripts of the demonstrations, documentation available on the Web site, and e-mail from company representatives were used to gather information about these products. Although all three companies provided a live demonstration of what patrons experience, none provided a demonstration of the library side of the product. This made it difficult to determine if the various products would meet staff expectations.

Paul Constantine's presentation on Cornell University’s LiveHelp Service at the 2000 Virtual Reference Desk Conference outlines several key points that libraries should look for when selecting a live reference service. The following list was modified to use as a checklist for comparing the three software products:

1. Does not require special software or hardware for the user.
2. Offers ability to push patrons' browser to a desired page.
3. Offers ability to send active embedded URLs.
Table 1. Use of Real-Time Reference at Selected Institutions

<table>
<thead>
<tr>
<th>Institution</th>
<th>Software</th>
<th>Hours</th>
<th>Usage</th>
<th>Usage per Hour of Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Florida</td>
<td>eShare’s Net Agent</td>
<td>M-F 10 A.M.–5 P.M.</td>
<td>375 questions in 3 months</td>
<td>5.3</td>
</tr>
<tr>
<td>Golden Gateway Library Network*</td>
<td>LSSI</td>
<td>unknown; 6 hours</td>
<td>5-6 questions in 6 hour shift</td>
<td>1</td>
</tr>
<tr>
<td>Carnegie Mellon University</td>
<td>live person</td>
<td>M-F 1-5 P.M.</td>
<td>60-80 questions per month</td>
<td>3-4</td>
</tr>
<tr>
<td>North Carolina State University</td>
<td>LSSI</td>
<td>M-F 9 A.M.–5 P.M.</td>
<td>100 questions per month</td>
<td>less than 1</td>
</tr>
<tr>
<td>Alliance Library System**</td>
<td>LSSI</td>
<td>24x7</td>
<td>474 transactions in approx. 3 months</td>
<td>less than 1 every four hours</td>
</tr>
</tbody>
</table>

* Includes 22 libraries and 120 staff members participating in online reference.
** Includes eight institutions.

4. Offers ability to survey user.
5. Supports multiple browsers.
7. Provides searchable transcripts.
8. Offers ability to create preformatted answers for frequently asked questions.
9. Identifies patrons by IP address.
11. Minimizes technical difficulties.
12. Is reasonably priced.

Summary and Evaluation of Library Chat System Features

Installation of Software

None of the products required installation of software for the user or for the library. Although many free instant messaging and chat programs exist, most require installation, which is why systems are being developed specifically for Web customer support and chat reference.

Push the Patrons’ Browser

All three provided the ability to push patrons’ browsers to Web sites but implemented this feature in different ways. Live Assistance opened a new window on top of the chat window, LivePerson opened a new window behind the chat window with both pages visible, and VRD opened the new window in a frame next to the chat window. LivePerson’s chat session crashed twice when a Web site was pushed.

Active Embedded URLs

VRD did not support sending active embedded URLs in the chat box but when a patron exits the system a summary list is provided of hyperlinks to sites visited in the session. The links are also active in the transcripts that are sent via e-mail to patrons. Both LivePerson and Live Assistance make URLs active hyperlinks in the chat windows. Live Assistance also allows patrons to send URLs as active hyperlinks.

Surveying the Patron

Live Assistance provided a password-protected administrator account that allows the library to create and edit surveys. LivePerson offered surveys in their Corporate edition. VRD did not have a survey feature at the time of writing; however, company representatives say it is in development.

Supports Multiple Browsers

Live Assistance documentation claims that their product works with all standard browsers version 3.x or better. VRD’s online documentation could not be viewed with Netscape 4.7. When viewed through Internet Explorer, the VRD documentation recommended Internet Explorer or Netscape 4.0 or greater. According to customer service representatives, LivePerson Pro supports all browsers.

Logs Transcripts for Future Reference and Searchable Transcripts

Live Assistance creates a database of transcripts that can be searched and allows librarians to assign categories to each transaction for report purposes (such as circulation and reference). Librarians can also type additional notes for follow-up using a post chat wrap-up tool. Live Assistance does not automatically make transcripts available to patrons, but the librarian can email a transcript after the chat session is over.

LivePerson logs transcripts and creates patron histories. These patron histories identify repeat visitors with cookies. Since LivePerson Pro does not automatically make transcripts available and they do not require patrons to
log in with an e-mail address, the librarian would have to ask the patron for their e-mail during the chat session and cut and paste the transcript into an e-mail message. It is possible to search transcripts by date, librarian, for the past week, or for a given month.

VRD logs transcripts but they are not searchable. Patrons are automatically sent a summary of their chat session as they close the chat and also via e-mail. In one demonstration, the e-mail of the transcript failed to arrive, and the customer service associate noted she had not received a transcript of the conversation in question either. When the patron closes the chat, a window opens, listing all of the Web sites or documents that were sent by the librarian.

**Fast Response Time**

All three systems were reasonably fast once the chat session began. Some of VRD's more advanced delivery features such as slide show and escort took some time, and it seemed to take a long time before an operator replied to the initial inquiry during a demonstration. They do send several messages indicating that someone knew a user was waiting and that "a senior librarian would answer shortly." These response times may not replicate the real world where patrons or librarians may be using various connection speeds.

**No Technical Difficulties**

All three systems were easy to use and had a simple patron interface. Live Assistance and VRD required the patron to log on with a name and e-mail address while LivePerson provided just an input box. One of the chat sessions with VRD had some system slowdown problems, and the e-mail transcript of one of the conversations never arrived. In two separate demonstrations with LivePerson, the chat session crashed when the company representative pushed a page.

**Price**

LivePerson Pro Edition was the least expensive of the three tested. The Pro edition offered the basic chat features for $89 per month. LivePerson Corporate Edition allows surveying and more reporting features for $350 per month. LivePerson does not provide on-site training, free or otherwise, but provides live help through their home page, and one-on-one usage instruction from account managers. Live Assistance was the next most expensive at $150 per month. They also charge a one-time fee of $500 for installation and over-the-phone training. The most expensive by far was VRD at $6,000 per year ($500 per month). VRD also charges a one-time fee for setup and training of $8,000. Two-day on-site training sessions are conducted by experienced online reference librarians. This initial training is supplemented with regularly scheduled online updates.

**Summary of Features and Product Comparisons**

A summary of the above features and products is provided in table 2. All three products provided the same core capabilities. VRD provides more features for sending information to patrons but at $14,000 was deemed too expensive, particularly because potential usage is unknown. Also, VRD did not provide some of the features we were looking for. The VRD system does offer nice features and has the advantage of specializing in library services.

Live Assistance had a more reasonable price and offered most of the features that we thought were important. They also had ample documentation on their Web site and very helpful customer service representatives who were knowledgeable about the needs of library customers.

LivePerson Pro was the least expensive product and offered the basic services we wanted but unlike the other two providers, LivePerson did not seem interested in the special needs of the library customer.

**Our Instant Messaging Reference System**

Although Live Assistance would have been our choice of the three systems we evaluated, we decided to create our own instant messaging system incorporating the features on the
checklist. This would allow us to customize the way it would be presented how quickly it responded, how it operated, and what kind of data it stored. We wanted to provide patrons with a service similar in appearance to our site search—but with librarians answering the queries rather than a search engine. Creating our own system would also allow us to protect the privacy of transcripts and patron data and develop customized reports. Also, our library was not in a position to purchase a system at this time. The administration was supportive of our ideas, but not ready to commit to a new service. External funding would have taken time to acquire, and we wanted to get started right away.

Morris Library is not the first to use a homegrown system. Marc Meola and Sam Stormont created a chat program called TalkBack, described by Meola as more of an “Internet paging” system using pop-up windows than a true chat. In their test from fall of 1998, librarians at the physical reference desk additionally answered questions through TalkBack. In six weeks (seventy-six hours per week), they received eighty-six questions through TalkBack. Meola and Stormont further refined their program and have an updated program called TalkNow, which uses PHP and MySQL. Comparing the features of their system and the three systems above helped us in the development process of our own system.

Our system was designed to meet the requirements of the preceeding checklist; figure 1 provides a brief summary, while a more detailed description of how the system works follows.

The Morris Library Online Reference system uses a Web browser for both staff and patron clients, and was tested with success in both Internet Explorer (5+) and Netscape Navigator (4.0+ but not Netscape 6). Both staff and patron clients are extremely responsive, with nearly instant response when both parties are directly connected to the Internet, and only a second or so delay when one party is on a 28K dial-up connection. Since the only requirement for the system is a Web browser, library staff and patrons can be in virtually any location. A test between the library, located in southern Illinois, and Cleveland, Ohio, resulted in no significant time lag.

When a librarian is online, the library’s home page (see figure 2) and the footer on every subsequent page have an input box and an invitation for patrons to enter a question if they need help, thus providing access to our instant messaging reference service in a similar manner to the site search. After clicking the submit button, a patron is connected with the service.

Patron Interface

The patron window is split into two frames (see figure 3). The top frame contains brief instructions, an input box with a submit button, and a disconnect button. The bottom frame contains either information about the connection or a conversation.

After a staff member responds, the bottom frame patron window changes from “attempting to connect” to a display of the conversation so far: the patron’s first submission and the staff member’s first response. The top frame changes to include a graphic showing that the patron is online, and also displays the screen name of the staff member at the other end of the connection.

Once the patron is connected, the conversation continues as long as both parties are interested. The patron can disconnect at any time with the Disconnect button, and is then taken to a screen that offers the option to reconnect, close the window, or display a transcript of the conversation. This screen is also where a brief Web

<table>
<thead>
<tr>
<th>Features</th>
<th>Live Assistance</th>
<th>Live Person</th>
<th>Virtual Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>No special software or hardware</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Push the patrons browser</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes*</td>
</tr>
<tr>
<td>Send active embedded URLs</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Ability to survey user</td>
<td>Yes</td>
<td>No**</td>
<td>No</td>
</tr>
<tr>
<td>Supports multiple browsers</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Creates transcripts</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Searchable transcripts</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Preformatted answers for FAQs</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Identify patron by IP address</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Fast response time</td>
<td>Yes</td>
<td>Yes'</td>
<td>Yes</td>
</tr>
<tr>
<td>Technical difficulties</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Training/setup fee</td>
<td>$500</td>
<td>$0</td>
<td>$8,000''</td>
</tr>
<tr>
<td>Subscription price</td>
<td>$150/month</td>
<td>$89/month</td>
<td>$6,000/yr</td>
</tr>
</tbody>
</table>

* VRD also offers delivery options of escort, slideshow, and capture.
** LivePerson Corporate edition provides surveys for $350 per month.
' LivePerson was fast, but experienced technical difficulties when pages were pushed.
'' Two-day, on-site training.
evaluation of the service is posted. In case patrons close the browser window or walk away from the workstation without disconnecting, the program periodically tests the patron’s window to see if it is still active. If it is not, the staff member is notified that the patron’s window probably has closed.

The patron transcript includes a full record of the conversation, including the time of each line of the conversation, the screen name of the staff member, and any URLs sent during the conversation. The patron may use the browser’s print, e-mail, or save functions to preserve the transcript for future reference.

**Staff Interface**

The staff interface begins with an offline screen. The top frame of the screen offers three options to connect: “Willing to backup others,” “Ready to answer questions,” and “Able to take redirects only.” The first option is equivalent to being on the front line of a reference desk, while the second is similar to being in a back office but monitoring desk traffic to help if things get busy. The third option was designed for nonreference staff to be available for specific types of questions—for example, a member of the circulation staff might be available for redirected conversations concerning renewals or overdue fine complaints.

The three connection options are followed by a password input and a Connect button. There is also a link to the maintenance pages. The bottom frame shows connection status (such as “You are offline”) and conversations when they occur. When staff members choose a connection option and enter their passwords successfully, the bottom frame changes to display the conversation with the patron.
"Waiting . . . ." When a patron connects to the service, the staff screen top frame changes to an input workspace. The bottom frame shows the conversation and information about the patron: IP address, Web browser, and the time the conversation began (see figure 4). The lines of the conversation are listed in reverse chronological order, so that the most recent remark in the conversations is listed first. This line is also in bold text. Each line is labeled "Patron" or by the staff members’ nicknames.

The top frame of the connected staff window identifies the name under which the staff member has logged in and has various input boxes for sending messages in different ways.

The largest text area is for free-text input, and is immediately followed by the Send button, making it easy to select after entering a comment. Other input boxes allow staff to select a preformatted answer, send a URL as a hyperlink the patron can click on, or enter a URL to which the user’s browser is pushed. If a preformatted answer is selected, the text appears in the main input box, giving the staff person time to edit any details before sending the message. The row of buttons following the input boxes are staff-specific and are similar to the preformatted answers except they send their associated messages instantly. Their intention was for the most common of comments, for example, “Welcome to online reference,” or “Hang on, I’m looking for the answer.” The Warn buttons send warning messages of increasing severity to ask if the patron is still there and interested in continuing the conversation, while the Kill button disconnects the patron.

If other staff members are logged in, their names appear at the top as options (radio buttons) to send messages to. The system defaults to "patron," however, as that is who the staff member is supposed to be communicating with when connected. If staff send each other messages, these lines only appear in the staff interface and are not displayed in any transcripts. A redirect feature is also available when multiple staff are logged in, allowing staff to pass entire conversations between each other. This was designed to facilitate referrals.

**Staff Maintenance**

The staff maintenance pages have five screens: View Summary, which shows current settings (see figure 5); Edit Buttons and Edit Answers, which allow staff members to add or change their preformatted buttons and answers on an individual basis (see figure 6); Edit Nicknames, which allows users to add or edit nicknames and passwords; and View Conversations, which allows for viewing or simple searching of past conversations (see figure 7). Staff can see others’ settings and conversations by selecting a name from a pull-down menu before they enter the maintenance section. Changes, updates, and deletions may be made while the staff person is logged into the service.

**How Does It Work?**

Online reference is mounted on a UNIX machine (Sun/Solaris) using the Apache Web server and uses Perl, MySQL, JavaScript, and HTML. All four languages are free and ample documentation is available on the Web. Perl has excellent process, file, and text manipulation facilities, particularly with Web applications. This made it a natural choice for the backbone, as described below. MySQL, a database management system, is easy to install and can store and retrieve large amounts of data very quickly. Anyone who can write SQL queries can query it. JavaScript and HTML, are necessary for the manipulation of Web browsers, which serve as the client.

In summary, the operation of this instant messaging system can be thought of as a database that contains the full text of a conversation. By submitting Web forms, the staff member and the patron update the same database table. Updated results are displayed in the bottom frame of each interface, so that each sees the current
conversation. The mechanisms that coordinate these actions are written in Perl, the information is stored in MySQL, and the display is presented with both JavaScript and HTML.

Perl

Perl is the actual program which sends the MySQL queries, JavaScript scripting, and HTML tags to the correct place at the correct time. The MySQL queries, the JavaScript scripts, and the HTML tags all reside in the Perl code.

The program itself resides in three files written in Perl: chat_patron, chat_staff, and chat_main. The first two are the actual instant messaging programs, while chat_main provides an interface for database management. All three Perl scripts use the Perl DBI module, which contains the necessary functions to interact with the MySQL database.

Chat_patron and chat_staff directly communicate with each other very little. Instead, each program takes the input provided by the staff or patron and saves it into a database. The other program then reads and displays the conversation from the database. Because of the speed of MySQL, this querying process can take less than a second. The only variables directly passed between the two programs are the session ID and the recipient, which link the clients to the correct database records. When a conversation is redirected from one staff member to another, only the recipient variable is changed. In this way, a new staff member receiving a redirected conversation gets all the information the previous staff member had—they literally pick up the conversation where the last person left off.

Although the patron interface looks quite simple, the underlying procedures are quite complex. Chat_patron generally alternates between two procedures. The first, bottom_connect, contains the intricacies of creating a file for the two programs to share, properly identifying which one it is and who the recipients are, and updating the status of staff
and patrons connecting and disconnecting. The second, do_work, checks to see if anyone is trying to talk to someone else and lets the involved parties know they are connected.

Chat_staff is very similar to chat_patron, in that it is reading from the MySQL database and posting the results of a query to the bottom frame of the staff window. However, the staff client also checks to see if the patron process is still alive—(about every thirty seconds) in case the patron has closed their window. The staff side is also watching for a “warn” file, which would indicate that a message from another staff member has arrived, or a warning that a patron is waiting to get on the system. The warn file reloads the staff interface to include the new information.

The two programs share many features involved in providing clear conversations; one example is that actions such as pushing the patron’s browser or sending a URL as a hyperlink are sent as lines of the conversation—but the actual characters are replaced with a message that makes sense to a librarian: instead of

\[
\text{<script>open("http://url2goto. com","onrefpush")</script>,}
\]

the conversation would display

[opened page: http://url2goto.com].

Chat_maint contains the insert, append, and query statements necessary to update, edit, and display the database of conversations, staff members, and preformatted answers. The Perl script allows users to interact with the MySQL database through Web forms.

**MySQL**

MySQL is an open source relational database management system. It is freely available for download on the Internet, is easy to install on several platforms and operating systems, and supports several client programs and programming interfaces, including the Perl DBI module.

The database is structured into seven tables, which are listed in figure 8.

SQL queries within the Perl scripts query the database for several functions:

1. Create a staff member’s customized workspace, including preformatted answers and buttons.
2. Update and display the current conversations to both staff members and patrons.
3. Show transcripts of conversations to both staff members and patrons.
4. Display the current settings of staff members’ profiles and allow additions, editions, and deletions through a Web form.

**JavaScript**

Functionality provided by JavaScript includes most windows-based management tasks in both staff and patron Web browsers. One of the most important things JavaScript takes care of is demanding focus from the correct window, frame, and input box. Focusing simply means keeping the most important window at the front of the screen, preventing the online reference window from getting buried under other browser windows that might be open. The Perl scripting initiates the focusing action at the correct time, but a JavaScript method
accomplishes it. Under direction from Perl, JavaScript methods also open and close windows, creating them to be the correct size and with the appropriate scrollbars and names.

JavaScript also plays a role in managing the various HTML forms. In the staff interface, for example, JavaScript transfers the text of the preformatted answers to an input box where they can be edited. JavaScript also confirms actions with dialog boxes, for example, deleting records from database tables.

**HTML**

HTML is used to provide a graphical interface for the online reference system. In addition to fonts, colors, and page layout, HTML performs several other functions, as directed by the Perl scripting.

The bottom frame of the staff and patron clients is directed by the Perl script to check the database table for new lines of conversation. When there is a new line added, the HTML `<META REFRESH>` tag updates the display in the correct frame. Also, the use of the HTML `<FRAME>` tag allows separate processes to display results in the same window, which is split into two frames.

### Advantages and Disadvantages

There are many advantages to using these languages in addition to the simple functionality described above. All are free and can be edited with a simple text editor and require no compiling. Programs written well in Perl are exceptionally efficient; the same is true of MySQL.

The only disadvantage of using Perl is that programming knowledge is required to install and configure the system. As a database system goes, MySQL is standard and therefore would present no problem to anyone familiar with writing SQL queries. However, anyone familiar with the other three can make changes within the Perl script. For example, someone who knew HTML could enter the Perl code and change such things as colors, fonts, and words.

Almost all of the disadvantages in using these languages to create an instant messaging system resulted from the system’s reliance on Web browsers as clients. For example, it is challenging to write JavaScript that is acceptable to both Netscape Navigator and Internet Explorer. Both browsers have internal rules for handling JavaScript commands, and use slightly different flavors of HTML. This made it difficult to design a system that worked identically in both languages.
Testing the New System

The instant messaging system described above took the library’s Web programmer the equivalent of three work weeks, or about $2,000, to complete. Morris Library will be testing this instant messaging reference system in the summer semester of 2001. We plan on limiting our audience to the SIU domain by only showing the service to SIU IP addresses. Creating our own system and hosting it on our own server allows us to authenticate by any desired IP range and limit our audience as necessary. We will be providing a URL to distance learners so that they may access the service from any location.

The service will be staffed Monday through Thursday, 1-4 P.M., which are our peak hours of use. Staff working digital reference shifts are doing so on a semi-volunteer basis and do not necessarily receive relief from other duties. In general, staff are taking one-three hour shift per week. Staff participants meet regularly to evaluate experiences and suggest improvements. As a result, we may change the way we offer access to the service, the wording of links to the service, staffing, or other options as we respond to patron input.

If our system offers good service to patrons and it is something the library is interested in pursuing, policy issues of greater depth will be examined. These would include how we will offer service to, how we will staff the service, and issues of reference quality and evaluation. We are continually reexamining our system requirements to meet our needs.

Future Exploration

By creating our own system, we will be able to tweak it to accommodate our use. We are able to quickly insert and edit survey questions, search our transcripts, and manage the data with great flexibility.

In addition to studying the actual system, we will be trying to evaluate how reference practices may be different through instant messaging. During initial use, we have already noticed a difference in the type of questions we receive. In a study of human-computer-librarian interaction, SooYoung Rieh found that librarians who formerly had the role of intermediaries now seem to have a role of technical assistants based on the types of questions they receive at the reference desk. Based on initial use, we have received no technical assistant-type questions through online reference, although they are a frequent occurrence at our physical reference desks.1 There are numerous avenues for investigation concerning this new technology and its use that could be investigated in future research.

The source code for online reference will be posted on the OSS4lib Web site (www.oss4lib.org) for download, however no technical support is available due to lack of resources. Librarians are encouraged to become involved in the development of reference software, whether it is through suggestions to companies who make such software or by creating in-house systems.

Acknowledgement

The authors thank Keith VanCleave, one of the “fretful minority” and the creator of our online reference system.

References and Notes


4. The total number of all types of questions for one week at all reference desks was 2,644; the total number of site searches in a week was 2,053.


11. Steve Coffman, E-mail to LIBREF-L on June 12, 2001. LSSI introduces New Privacy Protections for Virtual Reference, Libref-l@listserv.kent.edu.

12. Ibid.