A Web Link Management Tool for Optimizing Utilization of Distributed Knowledge in Health Care Applications

David F. Lobach, M.D., Ph.D., MS; Rhazes U. Spell, M.S.; Joseph W. Hales, Ph.D.; and Jean S. Rabold, B.S.N., M.B.A.
Duke University Medical Center, Durham, North Carolina

The number of health-related Web sites on the Internet is increasing. Incorporating these sites into clinical decision support systems and other health care applications can significantly enhance the educational and instructional value of such systems. While search engines exist for finding sites and criteria are available for assessing site quality, few tools are available for managing Web-based health care information. Management of Web-based information is particularly challenging because the information is continually changing and new resources are continually being added. In this paper, we describe the development and use of a Web-link manager for health care applications. This system retains search strategies for repeated use, catalogues search results in a search results database, accommodates tracking of site review and use status, and provides periodic checking of link integrity for sites that are used in local applications. The Web-link manager is currently in use to manage the links used in a clinical decision support system that presents clinical practice guidelines interactively to clinicians at the point of care.

INTRODUCTION

Increasing amounts of medical information are becoming available on the World Wide Web. Thousands of Web sites now post health-related information. This information can significantly augment the content of Web-based medical applications by incorporating linkages to these distributed knowledge resources [1,2]. In order to take full advantage of these information resources, adequate infrastructure is needed to identify relevant Web sites, evaluate the quality of the information they contain and manage the accumulation of these resources over time. Searching for information on the Web has been simplified by the availability of powerful Web-based search engines with user-friendly interfaces and health care filters (e.g., AltaVista, Excite, Lycos and Yahoo) [3] and medical Web site indices (e.g., Medical Matrix) [4]. The quality of Web-based health care information has been addressed by several efforts [5,6]. Nationally sponsored working groups [7] and private organizations [8] have developed sets of guidelines which identify the criteria one needs to consider in evaluating a Web site. In a separate initiative, the American Medical Informatics Association Education Working Group has drafted an initial report on developing quality assessment criteria for Web-based healthcare resources (personal communication-JWH). Additionally, several sites have instituted a process of peer review for site content that is similar to the peer review process for medical journals [9,10].

Beyond the searching and quality infrastructure issues, however, optimal utilization of these Web-based knowledge resources is difficult because, by design, Web-based information dynamically changes and increases. Managing this rapidly expanding pool of knowledge requires special tools. These tools must be able to confirm that previously identified sites are still active, determine which sites have been altered since last review and find newly posted sites relevant to the search of interest. In this paper we describe the design, development and utilization of WILLIAM: Web-based Internet Link Locator and Information Acquisition Manager. This Web-based tool stores Web searches; allows selection of a pre-defined, stored search string, passes this search to a commercial search engine for processing; accumulates search results in a relational database; facilitates review and categorization of these Web search results; and assesses Web-link integrity using a commercially developed link-checking application. We illustrate the use of our tool for selecting and managing Web links in the context of a Web-based clinical decision support system known as SIEGFRIED that interactively presents clinical practice guidelines at the point of care [11].

METHODS

Development of System Specifications

The goal of this project was to design and develop a system that identifies and manages the large number of Web links that could be used in support of clinical applications. The functional specifications for WILLIAM were derived from individuals who had experience managing Web links for Web-based health care resources. The functional requirements for this system were the following:

1) Allow users to enter search strategies, which can be stored for future reuse.
2) Search the Web using the most current and powerful technology available.
3) Return a list of candidate site titles and locations...
that would meet management proceeded review; not been identified. Accommodate the differentiates new applications. We developed the following design strategies to facilitate new functional requirements we had defined for the system. First, we concluded that the system needed to be Web-based. Development on the Web facilitates on-line access to sites for searching and review; it also allows inter-operability with existing Web-based tools. Second, we realized that we could not develop Web searching capability that could facilitate the search engines available through commercially developed search tools. Therefore, we decided to integrate our system with commercially available Web searching applications. Third, we concluded that we would need to maintain a database of all sites returned from searches so that we could differentiate new sites from sites that had previously been identified. We also realized that a comprehensive database was needed to include information about the review of sites and the local use of sites. In addition to the search result database, we recognized that we would need a database to store previous search strategies so that these searches could be periodically re-run. Last, we observed that excellent applications to assess Web-link integrity were already commercially available.

Creation of the Web-Link Management Tool

We built our Web-link management system on a three-tier client-server architecture (Figure 1). The application layer contained the programs that would allow the user to interact with the system. These programs accommodated the two central functions of the system, namely, performing searches and reviewing search results. The data layer housed the databases for the search strategies, the results returned from searches, and the review process and utilization tracking information. The middleware layer contained the programs that executed the business logic for running the system by allowing communication and data exchange between the application and data layers.

The functionality of the application layer was developed as two independent applications: the Search Manager and the Result Viewer. The Search Manager was subdivided into two components, the Search Initiator and the Search Result Parser. Both components were written in the Java programming language. Java was selected because of its support for Web-based applications and because of the robustness of the java.net Application Program Interface from Sun Microsystems, Inc. (Palo Alto, CA). The Result Viewer was developed as a series of HTML forms based on HTML 3.2 specifications. HTML forms were selected for this application because of their familiarity to Web users and their ease of integration with other Web-development resources.

Figure 1. Web Link Manager System Components and Function
The middleware layer contained the Search Database Management Tools and the Results Database Management Tools. These tools were developed using Microsoft Active Server Pages (ASPs) and Visual Basic scripts. These technologies were selected because of their robustness and relative ease of maintenance compared to traditional common gateway interface applications. In addition, ASPs support Microsoft’s active data objects technology, which simplifies the programming of connections and commands to and from the system’s databases.

The data layer consists of the Search Database and the Result Database. The databases were created in a relational format using Microsoft Access '97. The Search Database stores all of the information necessary to describe an individual search strategy. This information includes the strategy name, associated search string, search engine used, date run, the search author, and associated search topic. Searches are retrievable by subject or strategy name (alphabetized). The Result Database stores the information necessary to track all sites returned from searches, and process information for managing the review and use of the sites. Information stored in this database includes the site name; URL, date of last site update; site size; site Webmaster; the name(s) and subject(s) of the search(s) that returned the site, the date returned, the date reviewed, and an indicator regarding whether or not the site was selected for use in a local system.

Selection of a Web-link Checking Application

To select a commercial tool to assess the integrity of Web links stored in the Results Database, we established three evaluation criteria. These criteria were: 1) the ability to evaluate multiple sites during a single session, 2) the generation of a list of both active and inactive links, and 3) the provision of forwarding site information, if available, when a site was relocated. We identified four commercial Web-link checkers through Web searches and on-line publications about Internet tools: InfoLink (BigByte Software) [12], (Site Technologies) [13], Web Doctor (Blue Sky Software) [14], and Site Hog (RedHog Web Software) [15]. From these link checkers, we selected for a more stringent review three applications, which appeared to meet our evaluation criteria. Results of this review are summarized in Table 1. InfoLink was selected based on the close fit between the information it returned and the information we desired for our system. For example, other link checkers had either too little information or had information that was extraneous for our purpose such as arbitrary quality measures and pie charts for access time.

RESULTS

Function of the Web Link Manager

The following section describes the function of WILLIAM for identifying and managing web links. This description is divided into the searching and reviewing functionalities. The "step" numbers throughout this description refer to the interactions between system components (Figure 1). The user initiates a search by entering a search string consisting of key words connected by "+", a search strategy name, a search topic, search author, and selecting a commercial search engine. Alternatively, the user can select a previously defined search by picking it off of a list displaying all of the subject and strategy names that are stored in the database. This list is created by a request from the Search Initiator to the Search Database Management Tools when the application is opened (Step 1) (Figure 2). The Search Database Management Tools send a structured query language (SQL) query to the Search Database (Step 2). The Search Database returns a record set to the Search Database Management Tools (Step 3), which format the query results for display by the Search Initiator (Step 4). After a search strategy is entered or selected, the Search Initiator connects to the selected commercial search engine and passes a formatted search string (Step 5). Note that the Search Initiator must generate the query string based upon the search engine’s proprietary format. The search engine

<table>
<thead>
<tr>
<th>Criteria</th>
<th>InfoLink</th>
<th>Site Sweeper</th>
<th>Web Doctor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displays Active Sites</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Displays Inactive Sites</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Provides Forwarding Site</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Provides Date Last Modified</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1. Evaluation of Web-Link Checkers

Figure 2. Screen shot from the Search Manager
processes the query string and returns site results in a series of HTML pages to the Result Parser (Step 6). The Result Parser parses the HTML pages into site names and site URLs. Note that the Result Parser has to be unique for each search engine because each search engine has its proprietary method for returning search results. The Result Parser then passes the parsed results to the Results Database Management Tools (Step 7). The Results Database Management Tools form an "Insert" SQL statement and pass the site names and URLs to the Result Database (Step 8).

The approach to accessing search results is depicted in Figure 1, steps 9-12. Upon opening the Result Viewer application, the user is prompted for an action or request (Figure 3). The user may insert information about a site, delete a site, review a site, or request information about sites returned from a particular search. The Viewer passes this request to the Results Database Management Tools (Step 9). The Results Database Management Tools reformat the request to generate an SQL statement to pass to the Result Database (Step 10). The Result Database processes the request, generates a record set and returns it to the Results Database Management Tools (Step 11). The Results Database Management Tools delimit the records in a format that the Result Viewer can process and passes the records to the Viewer, which displays the requested information for the user. The user can then select a site URL to view the page or annotate the site entry with information about the review process or the use of the site in a local system.

**DISCUSSION**

In this paper we describe the development and illustrate successful use of a tool that assists with the identification and management of Web-based knowledge resources used in a clinical decision support system. We demonstrate that this tool can effectively facilitate the maintenance of Web-based distributed knowledge resources. The type of Web-link management tool we have developed will facilitate the incorporation and use of links to Internet resources that augment Web-based decision support systems and other Web-based clinical applications by enabling application managers to find, review, update and track relevant Web-based information.

Our Web link management system offers several advantages over a non-systematized management approach by integrating diverse functionality in a system which couples review of sites with an application that tracks the review and use of Web links. The system facilitates identification of new Web sites related to a specific topic. The search manager supports periodic repetition of previous searches. The retention of all previously identified sites allows filtering of new searches to selectively identify new sites for review. The Result Viewer integrates the process of site review with the administrative documentation required for managing sites by displaying both functions simultaneously on the screen. Because the system is Web-based, sites can be accessed directly for review. Additionally the Result Viewer dynamically prepares the list of sites to be reviewed in response to the interest of the user. The status of a site relative to its review is also clearly evident so that the sites needing to be reviewed are readily identifiable. Storing search results in a database allows the user to review sites at his/her leisure rather than immediately after they are returned.

We encountered several challenges in developing this Web-link management tool. These challenges predominately revolved around the interface between the Web-link manager and the commercial search engines. The commercial search engines have proprietary methods through which search strings are submitted and search results are returned. As a result, we required the user to select a search engine prior to submitting a search. In addition, we had to develop a unique parser for each search engine we included in the search manager. Each parser had to recognize the proprietary grouping of returned search results, e.g., sets of 10 or 20; identify the pre-formatted section of the results HTML page containing the site information, and decode the URL and other site information from the HTML code. We used the JAVA string manipulation functions to handle the parsing.

Figure 3. Screen shot from the Result Viewer
requirements in each search engine-specific parser we developed.

While our Web-link management system has been extremely useful, the tool we developed has several limitations, which point to areas for future development. The current version of the Web-link manager does not make any associations between sites returned in a set of search results. As a consequence, the review of search results could necessitate multiple independent visits to the same site. In a future version, we plan to develop a database engine to associate pages from the same site and, where appropriate, identify parent-child relationships between pages from a single site. The site tracking system of the current version is relatively limited. It records whether or not a site is used in a local Web-based application. A future expansion of the tracking system could include more extensive information about the review status and use of a site. Such information could document that the site was visited, reviewed, approved but not currently in use locally, or in use in a specific application(s) along with comments regarding the site content. While the current system can monitor if sites have been updated since last reviewed, it is often difficult to know what has changed without an extensive re-evaluation of the site. We are considering the addition of a page archiving system that would enable the reviewer to directly compare (perhaps even in an automated fashion) the site when it was last reviewed and archived relative to the revised site. Lastly, the current version only uploads into the database sites returned from a search. We plan to modify (e.g., a site discovered during review of search results) the Result Viewer to allow the user to access a site and upload it into the database without having identified the site through a search.

CONCLUSION

We have developed and implemented a Web-based tool to identify and to manage Web links for use in clinical decision support systems and other health care applications. This tool allows repeated searching, identifies sites not previously reviewed and tracks the status of sites that are reviewed and used in local applications. Tools such as this Web-link manager will facilitate the use and maintenance of the distributed knowledge resources of the Web in healthcare applications.

Acknowledgements

Project supported in part by a grant from the Agency for Health Care Policy and Research (R01 HS09436). We thank Naomi Pratt for excellent clerical assistance.

References

15. <http://www.compulink.co.uk/~allied-display/redhog/> (03/09/99)