Integrating DXplain into a Clinical Information System using the World Wide Web

Gai Elhanan, M.D., Socrates A. Socratous, James J. Cimino, M.D.
Department of Medical Informatics
Columbia Presbyterian Medical Center
New York, New York

The World Wide Web(WWW) offers a cross-platform environment and standard protocols that enable integration of various applications available on the Internet. The authors use the Web to facilitate interaction between their Web-based Clinical Information System and a decision-support system - DXplain, at the Massachusetts General Hospital - using local architecture and Common Gateway Interface programs. The current application translates patients' laboratory test results into DXplain's terms to generate diagnostic hypotheses. Two different access methods are utilized for this model; Hypertext Transfer Protocol (HTTP) and TCP/IP function calls. While clinical aspects cannot be evaluated as yet, the model demonstrates the potential of Web-based applications for interaction and integration and how local architecture, with a controlled vocabulary server, can further facilitate such integration. This model serves to demonstrate some of the limitations of the current WWW technology and identifies issues such as control over Web resources and their utilization and liability issues as possible obstacles for further integration.

INTRODUCTION

Amongst the consequences of the rapid growth of the Internet is the ongoing creation of a worldwide bio-medical information network that provides a vast array of services and databases, accessible to all by personal computers (PCs), using the World Wide Web (WWW)[1]. Although some applications may interact with the user, most of the resources currently available are informative, may be browsed through and downloaded, but are not intended to interact actively with other applications. The Common Gateway Interface (CGI) allows external programs to interface with WWW servers and thus, sequences of CGI programs may be used to integrate different Internet-based applications. A Web-based Clinical Information System (CIS) has been developed and is in daily use at Columbia Presbyterian Medical Center (CPMC)[2] while the Laboratory of Computer Science at Massachusetts General Hospital (MGH) has recently provided a Web-based interface to their computer-based diagnostic system - DXplain[3], as well as provided access to DXplain functions via TCP/IP. This paper examines the use of CGI programs to integrate these two applications in order to produce an automated method to create a diagnosis list based on a patient's laboratory data.

BACKGROUND

The CPMC Clinical Information System (described elsewhere[4,5]) has several features which make the task more attainable. Among these features are the centralized patient database server that supports queries via the Internet and the controlled vocabulary server using the CPMC Medical Entities Dictionary (MED)[6]. The MED provides translation capabilities to and from coded data and query capabilities within its structure. Essential to our task is the structure of the MED; each term has its unique integer code (MED code) as well as data about its name, synonyms, units and other information all contained in various slots to create a frame representation of the term. The various frames are linked semantically to create a semantic network that relates terms in meaningful ways. The MED also supports a classification structure that allows multiple hierarchies.

DXplain is a computer-based decision-support system designed to provide diagnostic advice based on clinical and laboratory data in the field of internal medicine.[3] DXplain accepts a list of clinical manifestations, laboratory results and other findings to suggest diagnostic hypotheses.

It can further be used to explain and justify components of its proposed diagnostic hypotheses. DXplain can be accessed via various TCP/IP function calls or through a Uniform Resource Locator (URL) link to a Web-based DXplain interface. The TCP/IP connection has no state and every call requires a new connection. The Web-based application offers pre-defined interface, simplicity of use and no need to reconnect, while maintaining full functionality. As suggested by the developers of DXplain[7] and in an analysis by Berner et al.[8], in their current state such programs should serve a prompting function, to remind physicians about related or overlooked possibilities.

METHODS

By selecting the "Laboratory" button, a user of the Web-based CIS at CPMC is presented with a date-sorted list of various laboratory procedures available for a specific patient. The user must then choose the procedures of interest to be presented win a test-specific display, organized by procedures. Test results are accompanied by normal ranges (if applicable) and a flag for normal / abnormal results. Each button push on the screen is associated with a CGI program which generate HL7 query to the clinical database server. All queries and results are in MED codes. The results of the queries are returned in HL7 format to be parsed by the CGI and be presented as an Hyper Text Markup Language (HTML) document by the user's Web browser[2].

By prior decision, the model integrating DXplain into CPMC-CIS, was limited to a single laboratory panel, the CHEM-20. This panel includes laboratory data for 20 blood chemistry tests, such as electrolytes, various enzymes and other tests (see Figure 1). We modified our CHEM-20 HTML display to include two additional buttons, one for each connection method. Each button is associated with an HTML "ACTION" that points to the specific CGI script that will process the form that will pass along a hidden string of CHEM-20 laboratory findings captured from the original HL7 returned result for that display. The data string is composed of the tests' MED codes coupled with coded status flags (normal, elevated, decreased, irrelevant) (35456!2! 35455!3!...35451!2!). No patient identifying data are included in the string.

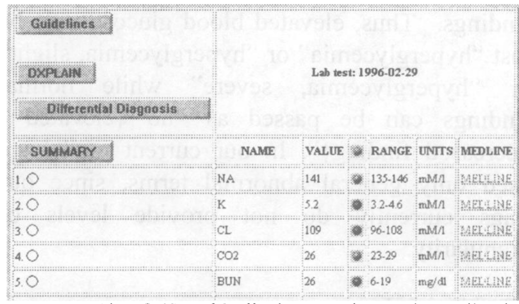

Figure 1: The CHEM-20 display on the Web application showing connections to DXplain's Web interface ("DXPLAIN") and TCP/IP calls ("Differential Diagnosis").

DXplain has two controlled vocabularies (for diseases and clinical findings) and supports functions which can match free-text entries and retrieve lists of possible matches or DXplain's exact matching term. The final string submitted to DXplain (via either method) must be composed from exact DXplain terms. Before submitting a query to DXplain, translation must take place at two levels, from MED codes to actual MED terms and matching those terms to exact DXplain's terms. For the second step, DXplain functions could be used, but this would involve many recurrent connections and direct user interaction. Instead, we utilize our local controlled vocabulary server, built around the MED, for this purpose. DXplain terms which match CHEM-20 test's terms were incorporated into the MED. A new slot was created in the MED and was defined for the tests included in the CHEM-20; this slot pointed from the specific test to a MED "abnormal finding" class specific for the substance measured in the test ("Abnormal Blood Level of {substance measured}"). Each finding class grouped all possible abnormal findings associated with a test (e.g. "hyperglycemia" & "hypoglycemia") and was central to our translation scheme (see Figure 2). In addition, each specific finding (e.g. "hyperglycemia") was made a child of another one of two general MED codes; "Elevated Abnormal Finding in Body Substance" or "Decreased Abnormal Finding in Body Substance". Thus the combination of MED code and status flag translates numeric values into an exact DXplain match from within the MED, utilizing built-in MED query functions available on the server. Findings with "Normal" flag were added to the string as "no {elevated / decreased substance}", as appropriate, by the same process. DXplain also supports some level of granularity regarding degrees of abnormality for some findings. Thus, elevated blood glucose may be just "hyperglycemia" or "hyperglycemia, slight" or "hyperglycemia, severe" while normal findings can be passed as "no {elevated / decreased finding}". In our current model we used only general abnormal terms, since our flags currently do not provide levels of granularity.

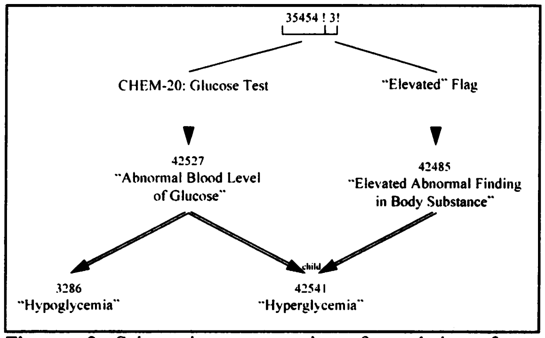

Figure 2: Schematic representation of translation of test results into DXplain terms using the MED.

Selecting either of the DXplain buttons on the laboratory results display triggers an HTML action that "POSTED" the data string along with gender and age definitions to a specific CGI named in "ACTION". All CGIs were written in the C programming language. A short description follows for each connection method:

DXplain's TCP/IP application

This application requires a series of recursive CGIs with many recurrent connections since all processing is done locally. We do not take advantage of the full spectrum of DXplain's functions, but rather intentionally limit it only to the functions that will enable us to query DXplain for terms, present the user with a diagnoses list and display DXplain's text for each disease on the list. The user is presented with an HTML document that displays the findings from the CHEM-20 automatically included in the search string. The user can then choose between submitting the list as is, check out some of the terms, or add other terms by entering them as free text (see Figure 3). If free text is entered, the "SUBMIT" action triggers a CGI that parses that string and uses DXplain's TCP/IP calls to match the items on the free text string with DXplain's exact terms. If a single exact match is found for a term, it is automatically added to the final query string. If the free text item invokes several DXplain matching finding, or if a DXplain finding is associated with several more specific findings, the user is presented with HTML screens that let him choose the appropriate term or delete it from the final query. When all the items from the free text string are processed, the final query is automatically sent to DXplain to produce the diagnoses list. The returned string is parsed and displayed as an HTML document. Each listed diagnosis is associated with a button that triggers a CGI utilizing another DXplain function enabling the user to query DXplain for textual data about any of the listed diseases, displayed as a simple HTML text screen. The user can switch between the CIS and the local DXplain applications using his Web browser backward and forward buttons.

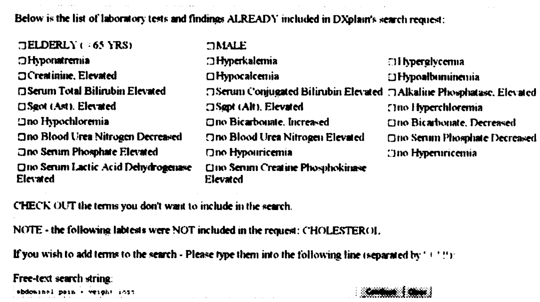

Figure 3: the TCP/IP application main screen containing exact DXplain terms and free-text entry option.

DXplain's Web-based application

This application requires some initial login parameters (provided by the Massachusetts General Hospital) to be passed by the CGI as well as the query string, as "hidden" variables attached to an HTML form as part of the CGI. The user is presented initially with a local CPMC disclaimer about DXplain followed by the main DXplain screen. This screen includes the Massachusetts General Hospital (MGH) disclaimer and all the functions of DXplain, with the CHEM-20 variables already in place as genuine DXplain terms (see Figure 4). The user may choose to add terms to the search request or use any of DXplain's other functions. The returned result is similarly displayed by the DXplain application, and allows the user to further interact with DXplain. The user can switch between the CIS and DXplain applications using his Web browser backward and forward buttons.

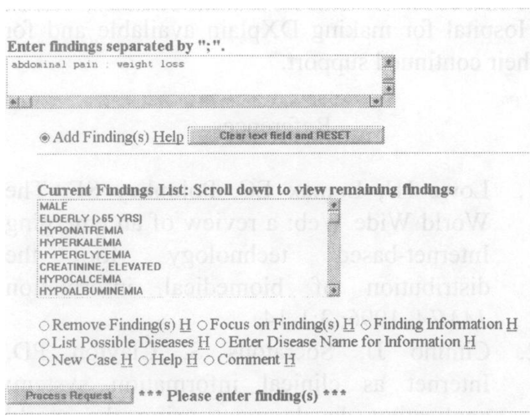

Figure 4: Web-based DXplain's main screen with CPMC CHEM-20 results already translated into DXplain terms.

RESULTS and DISCUSSION

Two separate applications were incorporated into the CPMC Web-based CIS, designed to access and interact with DXplain - the computer-based medical decision making program from MGH. The applications use CGI programs to translate patients' specific CHEM-20 tests results into specific DXplain terms using CPMC local architecture and controlled vocabulary server. The principal outcome of this model is the availability of an external decision support system from within our Web-based clinical information system. The environment of the Internet with the WWW makes such tasks relatively simple with its easy, inexpensive, multiple platform compliant, standard graphic interfaces and protocols[2,9,10]. The model demonstrates how the CPMC Web-based CIS can reach out and access the vast medical and other resources available on the WWW. In addition, a well designed local architecture, with a vocabulary server mediating requests between the user and the clinical database server, can further simplify and augment such tasks.

The Web paradigm is not without its limitations. The most significant in the current model is the "statelessness" of the TCP/IP connection. This poses a problem at several levels and locations, depending on the application: local connections to vocabulary and data servers and remote Web connections. This is most obvious with our local application. For each CGI's function call to DXplain a new communication session must be established, but with various unrelated applications competing for system resources connection timeouts are not uncommon, exemplifying the need for dedicated servers.

Another problem is the limited navigational capabilities of Web browsers; although the user can "surf" backward and forward through historical lists, or be sent to various screens via "return" buttons, he cannot navigate freely within the same application and between different applications. Exiting an application poses another such problem and, at times, can be done only in a backward fashion. Locally developed applications can cope with such problem by "returning" the user to the initial screen, but utilizing a remote interface does not allow for that. In the future, standard support of Web browsers of frames and Java scripts may alleviate this problem. Although local security measures may further complicate navigation within and between applications, we were not faced with any significant security issues. Since local security measures at both ends were in place and no identifying data was passed between applications, this model does not pose any additional security risks.

Several interesting issues arise from the use of different access methods to a remote resource. Connecting directly to a URL offers simplicity of use for the designer as long as he complies with the log-in specifications, at the price of conforming with the design of the remote application, which might not always suit local needs and format. Using TCP/IP calls burdens the local designer with recreating parts of the distant application, but offers much greater flexibility in terms of seamless incorporation into local applications and formats and can enhance integration between more than two applications. Not all providers of distant applications will allow various methods of connection, since providing full accessibility to all the applications' functions means some loss of control over the way their application/ database is being used. To what degree a provider must conserve his control depends on the nature of the database and the application, but liability aspects, especially in the field of medicine, will certainly have a significant effect on access options and thus indirectly may affect usability and level of integrality when using remote applications. Such issues will certainly affect the every-day use of available resources. If our model is to be used to incorporate more than just one distant application, the aspect of going through multiple, site-specific, disclaimers may deter any potential user from ever using such application.

DXplain and other diagnostic decision support systems have not been extensively evaluated in clinical settings[11] and the current model only allows us to test the plausibility of using the Web to integrate distant applications, but is not comprehensive enough to evaluate and field test the integration of such a decision support system or to test the comprehensiveness or relevance of the results. For such an evaluation, coverage of a larger spectrum of laboratory tests, radiology and other text-based reports will be required. Such coverage will enable to test whether decision support systems, such as DXplain, can serve as automated gate-keepers, prompting physicians for related or overlooked diagnoses.

CONCLUSIONS

The World Wide Web and its browsers offer a relatively easy and flexible cross-platform interface which enables interaction between Web-based applications and incorporation of one application within another. Local architecture can significantly simplify the design of such applications, while issues of liability and control over the original resource may limit flexibility and connectivity, especially when more than two applications are involved. Future evaluations will be needed to assess the effect of incorporating decision support systems into Clinical Information Systems and Electronic Medical Records.

Access to a demonstration site

http://www.cpmc.columbia.edu/intermed/dxplain.html

ACKNOWLEDGMENTS

Dr. Elhanan is supported in part through a National Library of Medicine (NLM) Training Grant LM07079-01. This work was supported in part by a High Performance Computing and Communication contract from the NLM.

The authors wish to thanks Kathy Famiglietti, Rich Kim and Octo Barnett at the Laboratory of Computer Science, Massachusetts General Hospital for making DXplain available and for their continued support.

References

1. Lowe HJ, Lomax EC, Polonkey SE. The World Wide Web: a review of an emerging Internet-based technology for the distribution of biomedical information *JAMIA*, 1996; 3:1-14.
2. Cimino JJ, Socratous S, Clayton PD. Internet as clinical information system: application development using the world Wide Web. *JAMIA*, 1995; 2(5):273-284.
3. Barnett GO, Cimino JJ, Hupp JA, Hoffer EP. DXplain - an evolving diagnostic decision-support system. *JAMA*, 1987; 258(1):67-74.
4. Clayton PD, Sideli RV, Sengupta S. Open architecture and integrated information at Columbia-Presbyterian Medical Center. *MD Comput*, 1992; 9(5):297-303.
5. Clayton PD. Integrated advanced medical information systems (IAIMS): payoffs and problems. *Methods Inf Med*, 1994; 33(4):351-357.
6. Cimino JJ, Clayton PD, Hripcsak G, Johnson SB. Knowledge-based approaches to the maintenance of a large controlled medical terminology. *JAMIA*, 1994; 1:35-50.
7. Feldman MJ, Barnett GO. An approach to evaluating the accuracy of DXplain. *Comput Methods Programs Biomed*, 1991; 35:261-266.
8. Berner ES, Webster GD, Shugerman AA et al. Performance of four computer-based diagnostic systems. *N Engl J Med*, 1994; 330:1792-1796.
9. Schatz BR, Hardin JB. NCSA Mosaic and the World wide Web: global hypermedia protocols for the Internet. *Science*, 1994; 265:895-901.
10. McKinney WP, Wagner JM, Bunton G, Kirk LM. A guide to Mosaic and the World Wide Web for physicians. *MD Comput.*, 1995; 12(2):109-114,141.
11. Kassirer JP. A report card on computer-assisted diagnosis - the grade: C. *N Engl J Med*, 1994; 330:1824-1825.