Architecture for a Web-Based Clinical Information System that Keeps the Design Open and the Access Closed

James J. Cimino, M.D.,[1] Soumitra Sengupta, Ph.D.,[1] Paul D. Clayton, Ph.D.,[1]
Vimla L. Patel, Ph.D.,[2] Andre Kushniruk, M.S.,[2] and Xiaoli Huang, M.S.[1]
[1] Department of Medical Informatics, Columbia University, New York, New York, USA
[2] Centre for Medical Education, McGill University, Montreal, Quebec, Canada

We are developing the Patient Clinical Information System (PatCIS) project at Columbia-Presbyterian Medical Center to provide patients with access to health information, including their own medical records (permitting them to contribute selected aspects to the record), educational materials and automated decision support. The architecture of the system allows for multiple, independent components which make use of central services for managing security and usage logging functions. The design accommodates a variety of data entry, data display and decision support tools and provides facilities for tracking system usage and questionnaires. The user interface minimizes hypertext-related disorientation and cognitive overload; our success in this regard is the subject of on-going evaluation.

INTRODUCTION

The World Wide Web is becoming a popular medium for providing access to clinical information systems (see, for example a recent review article[1] and dozens of examples in the previous AMIA Fall Symposium[2]). One advantage of the Web environment is the ability to link disparate components,[3],[4] servers,[5] applications[6] and even clinical information systems themselves[7],[8] into more complex systems. There are many challenges presented by the Web, including the provision of privacy and confidentiality.[9]

Based on previous work with a physician-oriented system,[10] we are developing the Patient Clinical Information System (PatCIS) to provide patients with access to their own medical record, permit them to contribute clinical data, give them access to educational resources, and provide customized medical advice.[11] The Web-based nature of the system has proved ideal for allowing a variety of faculty, systems programmer/analysts, students and fellows to develop components which can be incorporated easily into the application. However, our distributed approach presents challenges to building a robust, secure and flexible system. This paper describes our architecture and shows how it supports the goals of open design and limited access.

DESIGN GOALS AND CHALLENGES

We have many expectations for the project which seem, by their nature, destined to conflict. We have therefore identified several general features in order to identify potential discord.

Multiple Functions

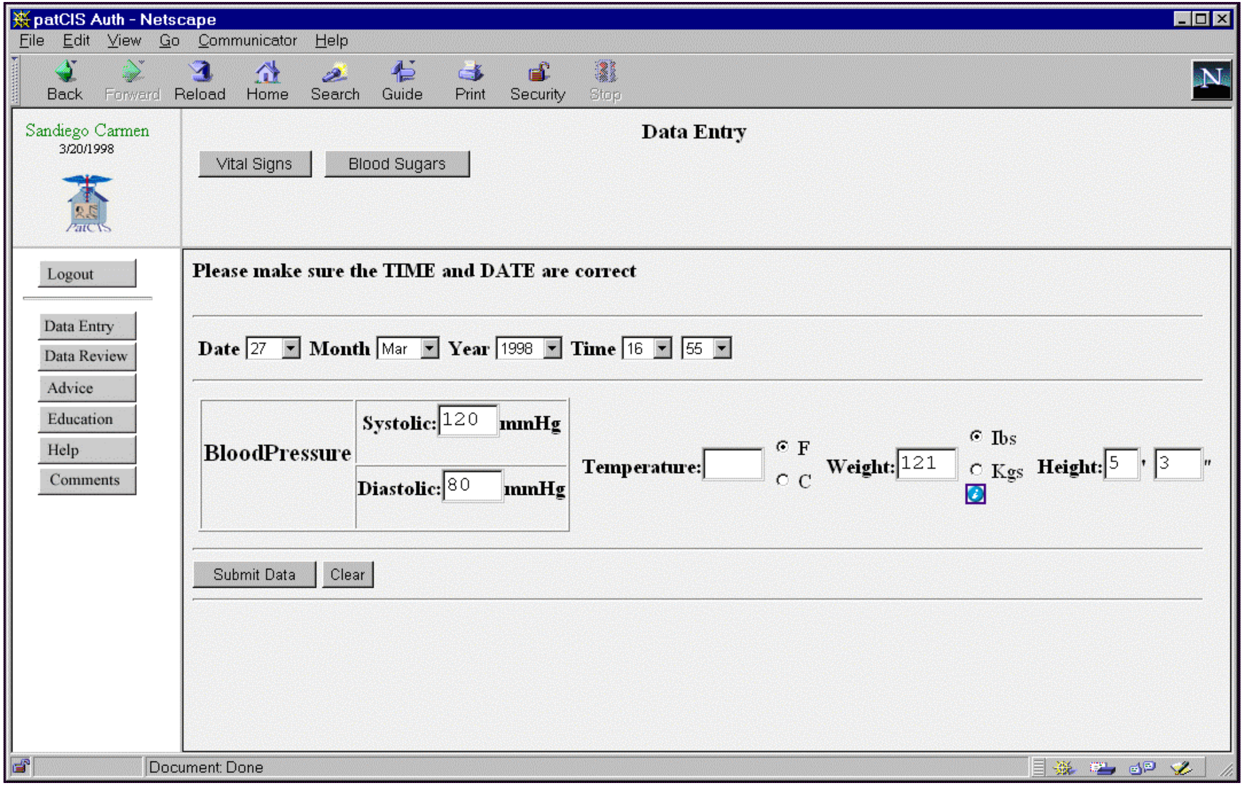
Within the four main functions of PatCIS (data entry, data review, education and advice), we envisioned multiple subordinate functions, some of which would be designed locally and some of which would be drawn from a variety of Web-based resources. We desired an architecture which would allow us to add components as quickly as we think of them.[12] In some environments, this requirement would suggest an object-oriented approach. The computing architecture at CPMC, however, is a server-based infrastructure which supports basic functions (e.g., security, database and vocabulary).

Ease of Use

The system is intended for use by patients with varying levels of computer training and experience. Our design-by-agglomeration approach seems doomed to result in a system which will have many different interaction styles, leading to the disorientation and confusion typical of hypertext systems.[13] To the extent we use outside Web-based resources, this is unavoidable and part of the price we pay for using applications "Not Invented Here". We therefore planned a user interface which provides some consistent look-and-feel as the user traverses among local and remote applications.

Security

Our desire to operate in the Web environment complicates security tasks. The traditional approach of having users access an application through an identification-authentication-authorization process once, at the beginning of the session does not work in the connectionless Web paradigm where there is no "session". Coupled with this is the need to keep track of all access to patient data as part of our institutional data security policy.



Please make sure the TIME and DATE are correct

Date 27 Month Mar Year 1998 Time 16 55

BloodPressure Systolic: 120 mmHg Diastolic: 80 mmHg

Temperature: ● F ○ C

Weight: 121 ● lbs ○ Kgs

Height: 5 ' 3 "

Figure 1: Sample PatCIS screen. Main buttons appear in the left-most frame; sub-buttons (in this case, for Data Entry) appear in the upper frame. The main frame shows a form for entry of vital signs. An infobutton (depicted by the "i" icon) is available to calculate body mass index from the patient's data.

Evaluation

We need to understand not only the usual issues of usability and utility, but also PatCIS's influence on patient decision-making. Evaluation is complicated by the remote nature of the users: patients will be using the application in their homes, far from interviewers and observers. Security logs can provide some information about the users' activities, but we require more detailed and personalized information in order to understand the cognitive processes involved in using PatCIS.[14]

SYSTEM ARCHITECTURE

User Interface Design

We expect our users to be a diverse and dispersed group. We are therefore making PatCIS compatible with Netscape Navigator, Version 4 (Netscape Communications Corporation, Mountain View, CA) because it is ubiquitous and freely available. We use a frame-based design, with a constant set of four frames, shown in Figure 1. The top-left corner is a logo so the patient will recognize that he or she is using PatCIS. The left frame contains a fixed set of "main buttons", each of which produces a set of "sub-buttons" in the top frame.

Each sub-button is associated with a Uniform Resource Locator (URL), which points to either a static file (a document in the Hypertext Markup Language (HTML) format) or a Common Gateway Interface (CGI) program somewhere on the Web. The specific function is determined by the developer of the sub-button; however, the HTML or CGI document to be displayed must appear in the large frame in the middle of the screen, and subsequent files to be displayed (through links contained within the first document) must be displayed either in the same frame, or a new browser window must be opened. Outside applications we link to will appear in new browser windows, since it will be difficult to guarantee that they will not alter other frames.

Security

In order to maintain the privacy of the patient's record, we include four security features in PatCIS: encryption, identification, authentication, and authorization. We make use of the standard Secure Socket Layer encryption built into the Netscape browser. Other functions are addressed through an initial log on function and on-going surveillance of all accesses to patient data.

Initial access to PatCIS is accomplished through a "Log On" form, requiring a user ID and password. If the user is accessing the system from a location outside the medical center campus, a second password is required, taken from a SecurID card (Security Dynamics, Bedford, Massachusetts) - a credit card-sized device which displays an ever-changing six-digit number.[15]

The log on form contains, as a hidden variable, a unique identifier for the form which corresponds to an entry in a registry on the Web server. Once the user enters the required information and hits "submit", the log on process verifies that the identifier appears in the registry and then removes it permanently. Subsequent attempts to log on with the identifier will fail. This prevents other users from logging on with the first user's identity (assuming the SecurID number was not needed).

If this log on process (identification and authentication) succeeds, the user ID is used to determine the medical record number (MRN) which corresponds to this user. This step suffices for authorization: each user may see his or her own record, and only that record. The system records the Internet address of the user's browser (the "host") and also provides a session number which is recorded in a session registry. This information is used subsequently to maintain system security, as will be described below. The session ID can be invalidated through several mechanisms: the user clicks on the "Logout" button, the same user establishes another PatCIS session (on either the same or different host machine), the user fails to re-log in when requested to do so (see below), or a prolonged period of inactivity has passed (a "long timeout", currently set to an hour).

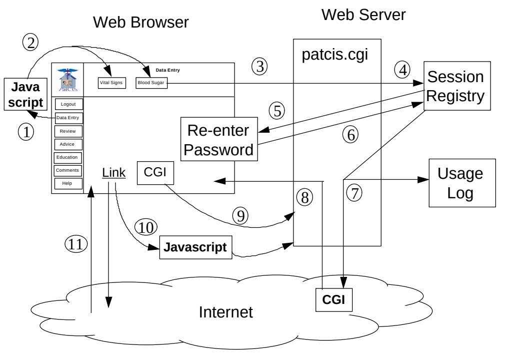


Figure 2: Flow of control in PatCIS. The user selects a main button (1) and a Javascript function generates a set of sub-buttons for the top frame (2). The user then selects a sub-button which calls patcis.cgi (3). As described in the text, the session registry is checked for validity and timeouts (4). In this case, PatCIS has determined that the user has timed out and needs to re-enter his password (5). The session registry is updated (6) and PatCIS then calls the CGI associated with the sub-button (7), recording the event in the usage log. The document returned by the CGI is passed back to the user's Web browser (8). If the CGI complies with PatCIS design, any subsequent CGI calls will be passed to patcis.cgi (9) as in step 3, and links will call patcis.cgi using a Javascript function (10). Links not conforming to PatCIS design will be directed by the browser to the appropriate Web resource (11), bypassing the usage log and other PatCIS functions.

Central Management of HREFs and CGIs

As previously noted, each sub-button is associated with a specific URL. However, clicking the sub-button does not result in direct access to the URL by the browser. Instead, the URL is passed to a central program called simply "patcis.cgi". This program provides central management for all HREFs and CGI calls, and carries out security and usage log functions. The steps in its operation (depicted in Figure 2) are:

1) patcis.cgi is called with several standard parameters (User ID, MRN, Session ID), the URL (as a parameter called "CGI") and any parameters which might be needed if a CGI is being called.

2) The User ID, MRN, Session ID and host are all checked in the registry to make sure that (a) the session is still valid and (b) these four parameters have not been altered in any way.

3) If the session is no longer valid, a message to that effect is returned.

4) If the session is valid but no activity has been noted recently (a "short timeout", currently set to 5 minutes), a form is returned to the browser, requesting the users' password. If this is satisfied, all the same parameters are passed back to patcis.cgi and it starts over again, this time passing quickly through these first four steps.

5) If the session is valid and either the "short timeout" period has not passed or the user provided the correct password, patcis.cgi obtains the proper document to return to the user's Web browser. If the value of "CGI" is a URL for an HTML document, the document is obtained. If the value is actually a CGI, that CGI is called.

6) If the document or CGI call was successful, the resulting document is passed back to the browser; if not, an error message is returned.

Integration of Components

Application components created by a developer or student can be readily incorporated into PatCIS in one of two general ways. The first is through creation of a sub-button. As previously stated, each main button is associated with a list of sub-buttons. This association is recorded in a configuration file which identifies the main-button, the name displayed on the sub-button, the order in which the sub-buttons appear, and the URL associated with the sub-button. When a user logs onto PatCIS, this file is read and the list of main buttons is generated dynamically for display in the left-most frame. Also generated dynamically is a set of Javascript functions, one for each main button, which causes (on clicking a main button) the HTML needed to display the relevant sub-buttons in the uppermost frame. Thus, incorporation of a new sub-button function requires making entries in a text file.

The second way in which components can be added is through the use of "infobuttons".[16] These are URLs (usually CGI calls) which link to independent resources that may or may not take patient-specific input and provide context-specific relevant information. For example, one data entry component of PatCIS allows patients to enter height and weight information. These data can be submitted for storage in the central clinical information system. Also available, however, is an "infobutton" (see Figure 1) which passes the data to a body mass index calculator (in this case, at a commercial site on the Web). The calculator, in turn, provides the user with customized information about the measure of his or her body fat and appropriate weight. In this way, infobuttons will be used to help users understand their own clinical data.

Exploitation of Web-Based Resources

The components being integrated into PatCIS are all Web-based, but they may be one of several forms: HTML documents, locally-developed CGIs, and CGIs developed elsewhere. The HTML documents in our system may present information to the patient directly and/or provide links to other resources. All CGIs in these documents are routed through patcis.cgi as previously described. HREFs to HTML documents are also passed to patcis.cgi through the use of a simple Javascript function which is associated with one of the frames. When content developers wish to include a link in a document, rather than specifying:

```
<A HREF="mydoc.html">Click Here</A>
```

they must specify it as:

```
<A HREF='javascript: parent.titleWin.document.
    FF.CGI.value="mydoc.html";
    parent.titleWin.document.FF.submit();'>
    Click Here</A>
```

Evaluation

The evaluation of the use of PatCIS has many aspects, which are discussed elsewhere.[17] Two aspects of the system design are relevant to the collection of data. The usage log, discussed above, can be used to identify the features used, the sequence of their use, and the time between different actions. In addition, the log on function checks a local database to see if the user has any outstanding questionnaires to fill out. If so, the user is presented with an HTML form to complete at log on time.

DISCUSSION

The PatCIS project is still in evolution, with a pilot trial scheduled to begin in the second quarter of this year and a larger trial to begin in the last quarter. We therefore have no experience with actual usage by patients; however, we have moderate experience with several Web-based clinical information systems[10,12] and believe that we can now assess those aspects of the architecture intended to address the development requirements.

Integration of multiple components has been remarkably simple, with pieces contributed by ten different individuals thus far. The rules concerning the use of patcis.cgi as a "Grand Central Station" have not been difficult to follow and have alleviated developers (except the author of patcis.cgi itself) from all security and usage log concerns. The current architecture does not allow us to track the use of non-PatCIS resources. However, we are exploring the ability of patcis.cgi to modify documents it retrieves from the Web such that all links in these documents will be refer back to patcis.cgi (tracking of this type has been demonstrated successfully in the Lamprey system[18] at Stanford University).

Naturally, we believe that our user interface will be easy for patients to interact with. We will not know for sure until we have begun testing with real users. When we do begin that testing, we believe, based on our experience with evaluation of other systems, that the information collected through on-line questionnaires and usage logs will provide a large part of what we need for our assessment. We may find, for example, that different arrangements of main buttons and sub-buttons are appropriate for different users. If so , the architecture will permit us to customize the presentation simply by associating different configuration files with different users.

Security issues will always be an important consideration for PatCIS. We are attempting to strike a balance between safety and convenience. For example, we have not chosen to add any security features to the users' browser machines, found in other implementations.[19,20]

Despite the risks, we believe that the Web environment is ideal for developing applications such as PatCIS. New ways to solve old problems, and solutions for new problems, arise every day.[1] The PatCIS architecture appears ideal for exploiting them as they appear.

Acknowledgments

The authors express their appreciation for the students and fellows who have contributed components to the PatCIS project, including Joseph Finkelstein, Osama Alswailem, Mehmet Birgen, Eneida Mendonca, Shabina Ahmad, and Cui Lei. This work was supported in part by National Information Infrastructure contract N01-LM-6-3542 from the National Library of Medicine.

References

1. Cimino JJ. Beyond the superhighway: exploiting the Internet with medical informatics. J Am Med Inform Assoc. 1997;4:279-84.
2. Masys DR, ed. Proceedings of the 1997 AMIA Annual Fall Symposium (formerly SCAMC). Published as a supplement to J Am Med Inform Assoc, 1997;4.
3. Karson TH, Perkins C, Dixon C, Ehresman JP, Mammone GL, Sato L, Schaffer JL, Greenes RA. The PartnerWeb project: a component-based approach to enterprise-wide information integration and dissemination. J Am Med Inform Assoc. 1997;4:359-63 (suppl).
4. Chueh HC, Raila WF, Pappas JJ, Ford M, Zatsman P, Tu J, Barnett GO. A component-based, distributed object services architecture for a clinical workstation. J Am Med Inform Assoc. 1996;3:638-42 (suppl).
5. Klimczak JC, Witten DM, Ruiz M, Mitchell JA, Brilhart JG, Frankenberger ML. Providing location-independent access to patient clinical narratives using Web browsers and a tiered server approach. J Am Med Inform Assoc. 1996;3:623-7 (suppl).
6. Sittig DF, Kuperman GJ, Teich JM. WWW-based interfaces to clinical information systems: state of the art. J Am Med Inform Assoc. 1996;3:694-8 (suppl).
7. Kohane IS, Greenspun P, Fackler J, Cimino C, Szolovits P. Building national electronic medical records systems via the World Wide Web. J Am Med Inform Assoc. 1996;3:191-207.
8. Halamka JD, Safran C. Virtual consolidation of Boston's Beth Israel and New England Deaconess Hospitals via the World Wide Web. J Am Med Inform Assoc. 1997;4:349-353 (suppl).
9. Rind DM, Kohane IS, Szolovits P , Safran C, Chueh HC, Barnett GO. Maintaining the confidentiality of medical records shared over the Internet and the World Wide Web. Ann Int Med, 1997;127(2):138-41.
10. Cimino JJ, Socratous SA, Clayton PD. Internet as clinical information system: application development using the World Wide Web. J Am Med Inform Assoc. 1995;2(5):273-84.
11. Cimino JJ, Patel VL, Sengupta S, Clayton PD, Kushniruk AW, Huang X. PatCIS: support for informed patient decision-making. Proc of the 1998 AMIA Spring Congress: 47.
12. Cimino JJ, Socratous S. Just tell me what you want!: the promise and perils of rapid prototyping with the World Wide Web. J Am Med Inform Assoc. 1996;3:719-23 (supl).
13. Conklin, J. Hypertext: an introduction and survey. Computer 20(9), 17 (1987).
14. Patel VL, Kushniruk AW, Cimino JJ. Cognitive issues in the evaluation of patient-based information systems. Proc of the 1998 AMIA Spring Congress: 33.
15. Flanagan JR, Montgomery RR. Clinical communication among health providers and systems using Web tools. J Am Med Inform Assoc. 1997;4:354-8 (suppl).
16. Cimino JJ, Elhanan G, Zeng Q. Supporting Infobuttons with Terminological Knowledge. J Am Med Inform Assoc. 1997;4:528-32 (suppl).
17. Kushniruk AW, Patel VL, Cimino JJ.. J Am Med Inform Assoc. 1998 (this volume).
18. Feliciano RM, Altman RB. Lamprey: tracking users on the World Wide Web. J Am Med Inform Assoc. 1996;3:757-61 (suppl).
19. Masys DR, Baker DB. Patient-centered access to secure systems online (PCASSO): a secure approach to clinical data access via the World Wide Web. J Am Med Inform Assoc. 1997;4:340-3 (suppl).
20 . Raman RS, Reddy R, Jaganathan V. A strategy for the development of secure telemedicine applications. J Am Med Inform Assoc. 1997;4:344-8 (suppl).