BEYOND THE WEB: BUILDING INFORMATION SYSTEMS WITH CLINICAL CONTEXT

Scott P. Narus, Ph.D., T. Allan Pryor, Ph.D.
Medical Informatics, Intermountain Health Care
Salt Lake City, Utah

Much of the current research on clinical information systems concentrates on World Wide Web (W^3) development. Although the W^3 is well suited to many information applications such as nonsensitive data exchange, it has drawbacks that may make it inappropriate for certain clinical environments. We present a solution that addresses many of the W^3's limitations by focusing on clinical requirements first and then applying the appropriate technology. We also make a case for clinical information systems researchers to become more involved in the development of Internet standards.

INTRODUCTION

While this year's AMIA Fall Symposium focuses on exploiting the Internet, we would like to propose that the most successful clinical information systems (CIS) of the future will capitalize on the underlying technology and philosophy used in Internet applications ("boundless" information space, common transmission language, platform independence, mixed-media, and the client-server paradigm)[1], but will do so in a clinical context. Many of the Internet CIS applications currently under development utilize the World Wide Web (W^3 or Web) and its corresponding technologies such as Hypertext Markup Language (HTML), Common Gateway Interface (CGI) programs, and commercially available client W^3 browsers, to enter, retrieve and display clinical data. Researchers have noted limitations to the W^3 paradigm, however, that are often consequences of the clinical environment. Although nonsensitive information meant for general use (e.g., clinical guidelines and drug references) is suited to the Web format, patient specific data may be better handled by a paradigm that both retains the Web's best features and addresses clinical characteristics. Additionally, data servers and client browsers that are designed with a clinical context in mind may be more appropriate for the logical and procedural flow of a clinical practice, as well as for the type and amount of data stored and retrieved. At Intermountain Health Care, we have designed a system that shares many of the Web's best features, but has focused from the beginning on the clinical environment. This system allows for the collection and display of medical information, permits users to build data screens without requiring prior knowledge of the underlying des language or database structure, and adds command a control features that are peculiar to the medical setting which the system will be used.

LIMITATIONS OF THE W^3

The W^3 was designed as a general means disseminating disparate information to a multitude of si independent of the eventual client platforms. As a res the general nature of the paradigm has produ restrictions on specific applications. CIS research have noted several limitations that are a dir consequence of the environment in which they oper namely:

- *Security*: There is an intrinsic requirement in CIS maintain patient confidentiality and limit unauthori system access. The W^3 by its very nature is an op system. In order to limit potential security breach Web servers may restrict user access to client syste located within the domain of the server. This solut has been chosen for most W^3 clinical applicati studied[2,3]. Additionally, commercial browsers do provide direct user logon and maintenance faciliti CGI programs written by CIS personnel may supp these features. Through these programs, the ser provides a security ticket to the client. Since the is connectionless, the ticket must be maintained returned to the server on each subsequent cli access in the current session. A second problem w Web security is that, for performance reasc browsers such as Netscape Navigator can retai local cache of documents retrieved from a serv These cached documents can easily be viewed unauthorized users at the client machine.

- *Stateless environment*: Because the W^3 connectionless, once the server has responded t client request, any knowledge of that client is l Thus there is no knowledge of the current client s on subsequent requests. Commercial brows require that a data form be sent to the server discarded) before a new form can be receiv Therefore there is no facility for multipage for Parameter strings may be passed between the cli

0195—4210/96/$5.00 © 1996 AMIA, Inc.

and server in order to maintain state information[4], but these strings may become extremely long and cumbersome.

- *Hypertext linking and client caching*: HTML documents use embedded hypertext links as pointers to other documents. Clicking on a link forces retrieval of a new document[5]. The new document is unaware of the context in which it was called, and subsequent calls back to previous documents also lose context of who they called and how they were recalled. This produces a non-intuitive flow in a clinical environment where data are requested and/or supplied in order to support a higher-level concept, or in cases where data are retrieved in order to be modified[1,2]. In addition, because of the way in which Web browsers cache documents locally for efficiency, subsequent changes to patient data may not be reflected as documents are recalled from cache[1,3].

- *Visual presentation*: The HTML language is limited in how information may be displayed on a document[6]. This includes text formatting, document layout, and data collection fields. Thus, HTML documents are less visually appealing and graphically intuitive than custom solutions[7,8].

- *Field validation*: Client browsers are currently unable to perform data validation at the field level and must wait until the entire form is submitted before either typographic or range errors are found[3]. Similar criticisms apply to such features as field alerts and inline help. Proprietary extensions to HTML that allow embedded code (such as Sun MicroSystems Java and Microsoft Visual Basic Script) may solve this problem, but present further design complexity for the document author since the author must be familiar with the features of a particular extension. Additionally, since these extensions are proprietary, they may be unsupported by other browsers.

In our research, we have discovered further limitations to the W^3 paradigm. First, there are problems with both the underlying document language and document creation. The current HTML specification lacks certain clinically relevant features such as (1) the ability to attach "hidden" properties to input fields (e.g. coding information, valid domains, numeric formats, units, negation), (2) embedded logic for field alerts, (3) categorization of documents, and (4) user context information. Plans for a new HTML specification (version 3.0) have been scrapped in favor of piecemeal enhancements to the current specification. Concerning the creation of Web documents, a knowledge of HTML is required[8] (although new HTML authoring tools have limited this). During document creation there is often a lack of direct interfaces to the clinical vocabulary and data storage structure[8].

Second, we have identified features that are needed on our client systems but are unavailable in commercial browsers. These features include (1) dynamic retrieval of patient information after form display, (2) dynamic repetition of data entry fields, (3) dynamic linking of textual or dictated notes and time stamps to predefined fields, and (4) implementation of user preferences versus machine preferences. Not all browsers are equal, either, and several, including Netscape Navigator and Microsoft Internet Explorer, implement their own nonstandard extensions to HTML that may be unsupported by other browsers.

Finally, the Hypertext Transfer Protocol (HTTP) used by the W^3 is suboptimal for transferring many data types and for processing requests containing multiple objects. The short-lived connections used by HTTP interact badly with TCP, the transport layer used by HTTP. Since HTTP does not allow multiple objects per request, one HTTP client can severely load a server if the requested document contains many embedded objects; if several clients make nearly simultaneous requests to the server, the overhead caused by TCP's TIME_WAIT state may cause the server to dedicate enough resources to effectively hang itself for several seconds.

PROPOSAL

Several years ago, Intermountain Health Care (IHC) decided to transition its future CIS architecture from a mainframe system to a client-server strategy. We also began targeting our health care delivery to outpatient care clinics, physician offices, and patient homes. The diversity of sites, personnel and client systems necessitated the development of a new data entry and display method. The resulting system, called "Profiler[9]", originally addressed many of the drawbacks expressed above, but did so in a non-standard format. We have slowly evolved the system to more closely match many of the W^3 standards in use today.

We began by developing a graphical design tool for creating platform-independent clinical data entry and display documents, called "Profiles". Similar in concept to WHAM![8], an HTML authoring tool developed at MIT, our design tool is linked to our clinical vocabulary server, allowing Profile designers to use a standard terminology. It also ensures syntactically and semantically correct database queries.

Profile design begins at the lowest level by selecting a medical concept from the vocabulary server (e.g.

"systolic blood pressure"). As this is done, all the relevant fields for the concept are retrieved (e.g. "units", "measurement device", "measurement site", etc.). The designer may choose any of these fields for inclusion in the Profile. Designers are allowed the flexibility of several visual presentation types (objects) for each field. These objects may be placed anywhere on the Profile in a true WYSIWYG format. Some style customization is provided, such as font size and type, color, and borders. Many of the objects we use are peculiar to the clinical environment, and were developed after consultation with physician user groups. Objects can have additional properties such as default values, automatic selection, negation (i.e., the ability to say the finding is "*not* severe"), and mandatory designation (i.e., a value for the field *must* be specified by the user before any data from the Profile can be saved).

The underlying Profile Definition Language (PDL), which closely complies with the Standard Generalized Markup Language[10] (SGML) (from which HTML was derived), allows Profile hierarchy and embedding, similar to hypertext linking in HTML and the new "frames" capability in Netscape Navigator 2.0. Therefore, once a low-level Profile has been designed, as explained above, a higher-level Profile can be built that simply calls these previously defined documents. In order to call other Profiles, designers may choose between (1) hypertext-like links, which when clicked replace the current Profile with a new one, (2) pop-up Profile links, which construct a moveable Profile window on top of the current Profile, and (3) "open" Profile frames, which retrieve the Profile and embed it directly into the current Profile.

We have provided the capability from within a Profile to call executable programs located anywhere on the network. A Profile can thus be designed for collection and display of specific information on a problem such as Asthma. This Profile can contain embedded Profiles both for collecting pertinent history and physical information, and for displaying current medications, links to Asthma protocols and journal articles on the Web or a local CD-ROM, and access to the current patient's electronic medical record.

Once a Profile has been stored, it may be used as a template in the design of a new Profile. We have utilized this capability in order first to build detailed physical exam Profiles for specific medical specialties, and then to remove many of the more specific sections so that the Profile may be incorporated in a General Practitioner's exam.

For our client systems, we have designed a browser that can search for, retrieve and display Profiles. Navigation is handled in a similar fashion to current Web browsers by using "Back", "Forward", and "Home" keys, history lists, bookmarks, hyperlinks, and direct calls. User logon to the network is handled by the browser, and current patient information is retained at the client as the user navigates through several Profiles, providing us with useful security and state information. Access to information is limited to validated clinicians, and can be further limited by user or user role. While Profiles may be cached locally for performance reasons, patient data never are. This is possible because the PDL ensures that data are always kept separate from document structure and format, a standard for SGML. The separation provides an added level of security, and allows cached Profiles to display the most recent patient data. The browser can incorporate existing patient information from the electronic medical record whenever called for by the user.

The browser also provides unique functions and features such as pop-up calendars and calculators, text and dictation insertion, and field validation. Each user can build an individual hierarchical "bookmark" list that reflects the user's most frequently accessed Profiles. Default behaviors that are unique to a user may be predefined and will follow the user no matter where logon is initiated. Because of the ability to call executables from within a given Profile, we can use other information services such as Web browsers and local drug formularies in order to search for and incorporate additional forms of information.

We utilize a three-tier client-server architecture in order to store and retrieve patient data. Clients use local *data retrieve*and *data store* objects to communicate with the middle layer data services. This allows our client software to remain independent of a particular database architecture. The middle layer services communicate directly with the longitudinal data repository, our electronic medical record. The services provide data security and integrity, and they implement any business rules. At the database server end, time-driven and data-driven logic modules can notify clients (through the middle layer) of alerts and special information. We are unaware if the W^3 permits this possibility.

The current browser runs exclusively in Microsoft Windows, our *de facto* clinical operating environment. Implementing similar browsers on other platforms is possible, though, because of the platform-independent PDL. The Windows operating system lets us take advantage of Microsoft's OLE standard and their forthcoming distributed objects model, but implementing these may limit our client platform choices. One distinct advantage of our browser is that we can be certain that

any Profile built with our design tool will be viewable by any client in the enterprise in exactly the same format as it was built.

We store Profiles in a central SQL database that provides searching by Profile ID, name and category. The database also keeps a table of objects referenced by each Profile. This lets us, with one server request, retrieve all of the embedded components in a Profile, reducing our network traffic and speeding information retrieval. During Profile design, it also permits the author to know which linked Profiles may be affected by changes in the current document. The central location of the Profile database lets clinicians access their particular Profiles from anywhere in the enterprise network. Database storage also grants us additional Profile maintenance features such as specification of a Profile's owner and allowable users, usage tracking, and versioning. We require that the database maintain the unique Profile ID and version number. This prevents the breaking of links, a common problem on the Web. We also reserve only a small portion of the entire ID space for our enterprise. If other health care enterprises were to adopt our system, they would receive their own ID space, allowing enterprises to share Profiles without creating identification conflicts.

CONCLUSION

We have tried to point out some of the drawbacks of W^3 development, and we have proposed a solution that addresses many of these limitations. Unfortunately, even though we have the ability to incorporate Web technology into our system, our solution will likely be unavailable to clinicians outside our enterprise. We also have the added complications of developing and maintaining our own document definition language and its accompanying document design and display tools. This obviously increases our design cost, and has delayed implementation of our system.

Limitations of the W^3 may not require CIS researchers to build proprietary systems as we have done. W^3 developers have found solutions that utilize features available in the current technology. For example, researchers at Columbia Presbyterian solved the security problem of locally cached documents by attempting to flush the cache with blank documents sent by the server when the user logs off the system[2]. Another group has created an additional state layer on their server in order to maintain state information[6]. Software companies are also attempting to provide solutions. Companies like Netscape and Microsoft are proposing add-ins and enhancements for their client browsers and servers that, although meant as general solutions, can address many of the issues discussed earlier. For example, Web server extensions, called "cookies", can allow the server to return state information along with a range of document addresses for which the state is valid. The client then stores this information locally. Anytime the client accesses documents from the specified range, the state information is passed back to the server[1].

On the other hand, if our goal is a universal system for information exchange, then these individual solutions will hinder this purpose if they remain proprietary and are not incorporated into a common standard. Therefore, it is in the interest of CIS researchers in general, and AMIA in particular, to become a strong voice in the development of Internet standards such as HTML and HTTP. We can also provide input to Internet software providers in order to influence the design of client and server software so that new strategies reflect the clinical context in which systems will be used. If standards and technology are developed in this way, IHC may decide to forego our current development in favor of a more general system. We are already moving our own designs in the direction of Internet standards. As the two paradigms become more similar, there may come a time when there is no logical reason for us to pursue a separate path. But standards and technology must reflect the kinds of clinical features we have deemed necessary to the success of our CIS. By participating in the development of Internet standards and software, we may eventually see the deployment of a clinical information system that truly breaks institutional barriers and allows the secure flow of information anywhere it is requested.

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):1-14.
2. Cimino JJ, Socratous SA, Grewal R. The Informatics Superhighway: Prototyping on the World Wide Web. In Gardner RM, ed.: *Proc of the Nineteenth Annual SCAMC*; New Orleans, LA; October, 1995:111-115.
3. Chute CG, Crowson DL, Buntrock JD. Medical Information Retrieval and WWW Browsers at Mayo. In Gardner RM, ed.: *Proc of the Nineteenth Annual SCAMC*; New Orleans, LA; October, 1995:903-907.
4. Cimino JJ, Socratous SA, Clayton PD. Automated Guidelines Implemented via the World Wide Web. In Gardner RM, ed.: *Proc of the Nineteenth Annual SCAMC*; New Orleans, LA; October, 1995:941.

5. Widman LE, Tong DA. EINTHOVEN on the World Wide Web: A Tool for Analysis of Cardiac Arrythmias. In Gardner RM, ed.: *Proc of the Nineteenth Annual SCAMC*; New Orleans, LA; October, 1995:968.
6. Willard, KE, Hallgren JH, Sielaff B, Connelly DP. The Deployment of a World Wide Web (W3) Based Medical Information System. In Gardner RM, ed.: *Proc of the Nineteenth Annual SCAMC*; New Orleans, LA; October, 1995:771-775.
7. Vanzyl AJ, Cesnik B. A Model for Connecting Doctors to University Based Medical Resources through the Internet. In Gardner RM, ed.: *Proc of the Nineteenth Annual SCAMC*; New Orleans, LA; October, 1995:517-521.
8. Hinds A, Greenspun P, Kohane IS. WHAM!: A Forms Constructor for Medical Record Access via the World Wide Web. In Gardner RM, ed.: *Proc of the Nineteenth Annual SCAMC*; New Orleans, LA; October, 1995:116-120.
9. Narus SP, Pryor TA. Profiler: A Tool for Devloping User Specific Clinical Input Screens (abstract). In Hripcsak G, ed.: *Proc of the 1995 Spring Congress of AMIA*; Boston, MA; June, 1995:68.
10. Goldfarb CF. The SGML Handbook. New York: Oxford University Press, 1990.