Medical Information Retrieval and WWW Browsers at Mayo

Christopher G. Chute, MD, DrPH
Douglas L. Crowson
James D. Buntrock

Section of Medical Information Resources
Mayo Clinic/Foundation
Rochester, MN

Medical information retrieval from "Master Sheet" entries specially indexed for research retrieval has been part of the Mayo culture since 1909. Providing easy to use and universally available WWW access to these and other patient information databases at Mayo via browsers, shines a bright light on issues of privacy and confidentiality, user authentication, need to know, data transmission security, and technical details of interfacing disparate databases on a spectrum of platforms to many types of workstations using a variety of browsers. We review our recent experience, and generalize pertinent issues.

INTRODUCTION

The practice of medicine has become increasingly information intensive over the near century during which the Mayo Medical Records have functioned. The historical problems of identifying and retrieving patient diagnoses and procedures was operationally solved at Mayo by the introduction of a paper based "Master Sheet" where a succinct description of all major in-patient and out-patient is maintained[1]. These entries are then indexed, using a variety of computer assisted tools[2,3] for research and practice management inquiry[4]. Thus, Mayo investigators and practitioners have had access to the repository of patient experience at Mayo as part of our corporate culture; these experiences form the basis of approximately 2,000 papers, reports, reviews, or abstracts published each year.

Traditionally the retrieval and review of patient data has been a leisurely research undertaking, rarely contributing to patient care in real time. The specter of inquiring whether the institution has, in the vision of Blois[5], "ever seen a patient like the one I am seeing now" is not yet a reality. This is in part due to a limited, albeit ever growing, amount of electronically accessible patient data; our main repository of experience still resides in 4.6 million *paper* patient dossiers.

Nevertheless, patient data presently archived in machine format remain awkward and tedious to access, due to a fragmented electronic database legacy. While Mayo's design for a comprehensive Electronic Medical Record (EMR) will rectify this database access problem for prospectively collected information, we presently confront over 200 major clinical databases linked in some way to our Master Sheet indices. Further, these databases reside on heterogeneous hardware and data repository software, making a common interface to them a substantial challenge. Remote client support on a bilateral basis (user to database) generates a huge combinatorial burden of installs, updates, and support. While we have previously attempted to provide common interface solutions[6], this did not avoid the difficulty of maintaining the remote client (SAS) or developing the application specific interface.

The introduction of the World Wide Web (WWW) protocol was in response to the need for information sharing over disparate platforms[7]. The later introduction of graphical browsers, notably Mosaic[8] and Netscape, addressed needs for intuitive, easy to use interfaces. Their subsequent emergence as the *de facto* remote client standard poses an even greater attractiveness. As providers of data and information, we can concentrate on creating WWW interfaces to Mayo data repositories, with no need to concern ourselves about installation, maintenance and updates of remote client applications; increasingly the "browser" tend to be already there.

Our task then became reduced to how we might compose HTML (HyperText Mark-up Language, the

0195-4210/95/$5.00

lingua franca of WWW applications) interfaces to our databases and indexed patient information. This report overviews our preliminary experiences, and highlights the critical problems we confronted in this process. While the technical development issues were interesting, the dominant consideration remains that of privacy and patient confidentiality concerns in this new world of ready and faceless electronic access to profoundly sensitive information.

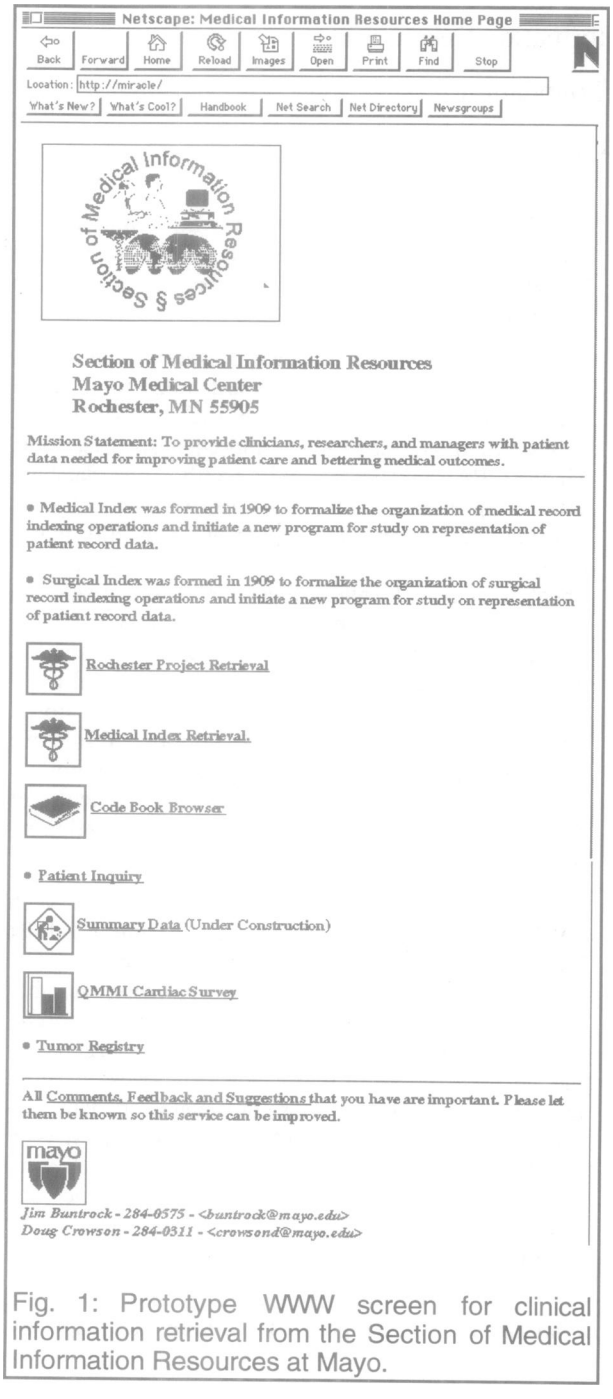

Fig. 1: Prototype WWW screen for clinical information retrieval from the Section of Medical Information Resources at Mayo.

TECHNICAL DEVELOPMENT

The Section of Medical information resources has for several years relied on the powerful prototyping and robust performance of the PERL language[9]. Although it is an interpreted environment, execution speed on modern UNIX workstations is more than acceptable for virtually everything we have implemented. Application development speed is greatly leveraged by the rich system command and statement library of the language, and its interpretive nature. Most importantly, the language can support SQL interfaces to a number of relational database environments, including DB/2, Ingres, Sybase, and Oracle (e.g. the IngPerl extension set). We have substantial experience writing these database interfaces for research and development of information retrieval tools, and thus regarded PERL with SQL extensions as the obvious tool for the generation of database interfaces.

We created a prototype interface, which collects some of our database linkages onto a single home page (Figure 1). Among the advantages of WWW interfaces is the ability to enter users at any point in the Web structure, for example at this common entry point home page, or deeper into the system at a user interface page of direct relevance to them, such as the interface to Mayo's Tumor Registry database (Figure 2).

HTML readily supports linkage to programs external to the HTPPD server (the program processing requests from the remote client), evidenced by the spawning of a graphical viewer to process an image file. We created PERL scripts which were the target of mouse clicks from the remote Web users (clients) that would spawn and process user inquiries. PERL excels at parsing text strings, a property we leveraged by designing screens that would collect user responses, specify data fields, and query values. Figure 2 illustrates some of the more common variables users tend to specify when asking for the frequency and breakdown of cancer treatment experience at Mayo. The "Submit" button collects the responses on this forms-enabled screen, and passes them back to the HTPPD server. They are then handed to the PERL script, which generates and executes the appropriate SQL code on the fly.

The response to these inquiries can be a complex event. The PERL scripts process the "answer" from the SQL databases, and depending on their content, generate an HTML reply on the fly. These replies can indicate something as simple and static as an error condition, or can respond to the size of data content response and pose another HTML form that asks about refining the query, or otherwise

modifying the volume of returned data (just passing on the whole mess is almost always a form option). Additionally, options about data format representation can also be posed as a function of the data, including graphical plots of data frequencies, or formats optimized for printout. The flexibility of the PERL script enables it to pass the SQL response onto a graphing utility (e.g. SAS/Graph® with GIF output extensions) or to simple cross tabulation utilities. There is essentially no limit to the elements of inquiry/retrieval processing that cannot be coordinated by these PERL scripts, interacting with the user via dynamically generated HTML screens and a suite of UNIX post-query processing tools.

We are not the first to publish on the utility of PERL interfaces for dynamically generated HTML screens. At least one other group has implemented this approach[10]. They too concluded that the HTML versions which support forms are the minimal level required for practical functionality. We agree with Willard *et al.* that most of the subsequent HTML extensions, save perhaps proposed security enhancements, add limited utility.

Standards "Under Construction"

With any new, innovative introduction of technology, the devil is in the details and the ever changing "standards." The original HTML v1 was limited to simple hypertext links, and lacked the notion of forms and responding user data or specification. Reports could be requested, but not customized with these early variants. Not surprisingly, some WWW browsers such as early Mosaic releases, still suffer for these limitations.

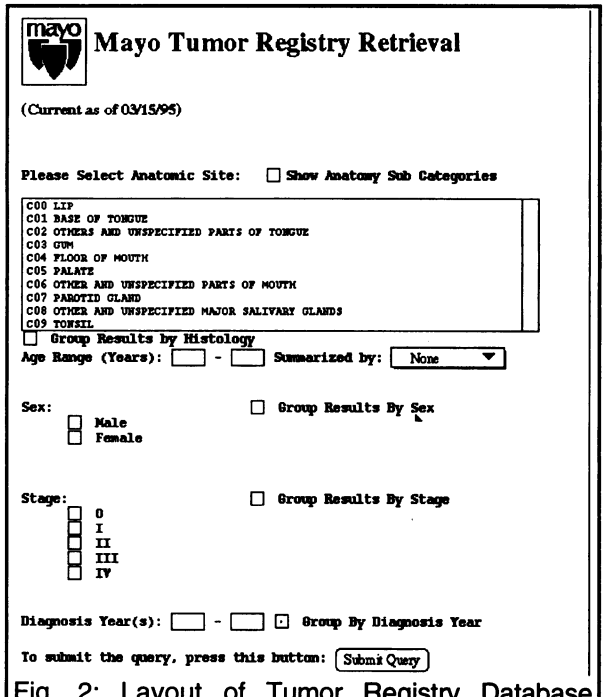

Fig. 2: Layout of Tumor Registry Database interface. This navigates inquiries against the 150,000 cancer registry patients maintained in DB/2.

HTML v2 introduced forms that support user responses and field entries that are returned to the HTTPD server. This is the functionality that we have found so critical to the creation of useful PERL scripts. Variants of forms supported include toggle button, radio (exclusive) switches, non-exclusive selections, option lists, and text entry. We have set our minimum standard at this level, and simply ask users that have WWW browsers that do not support these extensions to upgrade their tools. This is the least common denominator for practical support of SQL inquiry.

Version 3 of HTML introduces cosmetic elegance, in the way of font and format specifications. If you observe blinking text, you are looking at a version 3 combination of browser and page. More interestingly, table layouts and improved interfaces to external tools (e.g. jpeg graphical viewers) are supported. The ability to specify which server routine will process the remote form contents, essentially a multivalued submit button with a menu of options to which you can send your data, is an especially powerful tool. As of this writing, these features are in beta releases of Netscape only, and cannot be trusted to exist on the average users' desktop.

Performance

A question we asked early in the process, was how much performance are we sacrificing for the advantages of a generalizable tool. The answer, somewhat to our surprise is effectively none. We measured response times at several points of the process: WWW to server, PERL execution, SQL execution, and network latency at all segments. The results, in our environment, yielded the following observations:

- Network latency from WWW browser to server was invariably negligible (albeit, this was on Mayo's internal network. Outside Internet access to any of these servers is presently prohibited.
- SQL execution was the single largest bottleneck.
- For some applications, latency to the mainframe DB/2 databases degraded performance profoundly, to the extent that we simply chose to replicate these data on UNIX Ingres

From these measures and observations, we conclude that we experienced no appreciable

degradation in time/response performance using the WWW interface to clinical databases.

The WWW - PERL - SQL approach to remote information retrieval does have its disadvantages; functionality is always compromised by a generalizable tool. In our experience, these limitations are substantial, but are overwhelmingly outweighed by our improved development time and the unburdening of remote client support. Specific limitations pertinent to remote data retrieval we observed include:

- HTML forms are static. Forms or fields cannot be updated once they are displayed.
- Values and field content can only propagate forward between forms, they cannot be sent backwards to previous forms in a WWW browser stack.
- Multi-window communication is not supported. For example one cannot easily browse a code book in one window, and use code values in another. Cut and paste will work, but inter-window process communication is preferable.
- Field level validation is not supported. The entire form must be submitted en block to the server, and validated. The entire form is then accepted, or rejected for content/query errors.

User authorization/verification cannot be practically accomplished on a field or data element level. The resolution of security and privileged access is at the level of an entire form.

Security

The hardware level of security afforded by Mayo's Internet firewall gave us substantial confidence in performing these experiments. Nevertheless, we permitted *no* access to patient identifying information whatsoever (see Confidentiality below.)

The useful introduction of WWW technology for confidential patient data must await the release of secure protocols, and the ready availability of browsers that support them. This issue is being driven by the economic requirement for online transaction processing using credit cards and other sensitive financial data. These tools will enhance the security and authentication of legitimate clinical users with a need to know patient information, initially in aggregate form and eventually at the patient specific level. The recent convergence of Secure Socket Layer (SSL) and Secure HyperText Transport Protocol (sHTTP) is a welcome step forward toward achieving this functionality[11].

CONFIDENTIALITY

Perhaps deserving a manuscript in its own right, the importance patient confidentially considerations in remote database access cannot be sufficiently emphasized. Historically, Mayo research investigators and clinicians have had free access to patient information, with strongly stated policies that it should be exercised only on a need to know basis. This policy has operated satisfactorily with our unified medical dossier of in-patient and out-patient records for nearly a century.

Electronic access to the same information has many different considerations, the most notable being its relative ease and the perception of anonymity. This later point bears examination, since the reality could not be more different; virtually all electronic access to patient data at Mayo is irrevocably logged. Footprints cannot be avoided using any technology of which we are aware, although the question of whose footprints are left (simulated or real) remains.

In Mayo's emerging Electronic Medical Record System, great attention is paid to user authorization. The physical token/secret combination of a user card (the consultants ever valuable Parking Card with magnetic stripe) and a password serve this purpose. Authentication is scheduled to be undertaken using Kerberos and encrypted information transfer. We will draw upon this model, modified perhaps only to the extent that SSL and sHTTP might replace Kerberos, in our implementation of secure information retrieval tools. All these protocols and security issues pertain to access within Mayo's Internet firewall.

Internet access and security will be addressed very simply: we just say no. The existing firewall will not permit ftp, rlogin, telnet, gopher, or WWW packets to pass. To the extent that we support or allow any Internet access, it is on sentry machines that have stripped kernels and limited binary libraries outside the firewall. The information these machines are permitted will never include patient identifying data.

CONCLUSION

The emergence of the WWW and ensuing HTML standards has practical and positive consequences on remote client access to clinical databases. Performance impacts occur at the functional interface level, but have negligible contribution to response times. Nevertheless, these functional restrictions are overwhelmed by the development and support advantages which accrue from adopting Web tools for remote database inquiry and retrieval. Security and confidentially remain the

most important issues, and hold promise with the convergence of WWW standards for secure information interchange. The issues of user authentication are not intrinsically solved with existing Web technology, but precedents in Mayo's Electronic Medical Record application pertain.

ACKNOWLEDGMENTS

Supported in part by NIH grants LM05416, HS/LM08751, and AR30582. We thank Karen Elias for manuscript assistance.

[1] Kurland LT, Molgaard CA. The patient record in epidemiology. *Scientific American* 1981;245:45-53.

[2]Yang Y, Chute CG. An application of Expert Network to Clinical Classification and MEDLINE indexing. Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care. *JAMIA* 1994; 18(Symp.Suppl.):157-61.

[3]Chute CG, Yang Y, Buntrock J. An evaluation of computer-assisted clinical classification algorithms. Proceedings of the 18th Annual Symposium on Computer Applications in Medical Care. *JAMIA* 1994;18(Symp.Suppl.):162-6.

[4]Chute CG. Clinical Data Retrieval and Analysis: I've Seen a Case Like That Before. *Annals of the New York Academy of Sciences* 1992;670:133-40.

[5] Blois MS. *Information and Medicine. The Nature of Medical Descriptions*. Berkeley, CA: University of California Press, 1984.

[6] Van Grevenhof P, Chute CG, Ballard DJ. A common and clonable environment to support research using patient data. *Proceedings of the 15th Annual Symposium on Computer Applications in Medical Care* 1991;358-62.

[7] Berners-Lee T, Cailliau R, Luotonen A, Nielsen HF, Secreat A. The World Wide Web. *Communications of the ACM* 1994;37(8):76-82.

[8]Schatz BR, Hardin JB. The NCSA Mosaic and the World Wide Web: Global Hypermedia Protocols for the Internet. *Science* 1994;265:895-901.

[9]Wall L, Schwartz RL. *Programming PERL*. Sebastopol, CA: O'Reilly & Associates, Inc., 1990.

[10]Willard KE, Hallgren JH, Connelly DP. W3 Based Medical Information Systems vs. Custom Client Server Applications. *Proc. of the Second International WWW Conference* 1994:*641-51*.

[11]A New Safety Net: Top Internet vendors agree to online security protocol. *INFORMATIONWEEK* 1995;April 24, 1995:14-15.