Internet Tools Access Administrative Data at the University of Delaware

by Carl Jacobson

The introduction of NCSA’s Mosaic browser ignited a fire of interest that is changing the face of the Internet and the way we deal with networked information. The scramble for commercial success on the Internet has brought many technology vendors into the Web trade, resulting in the development of new tools and methods. As these advances define the role of commerce on the Internet, they will also change the way we conduct routine business on our networked campuses.

The World Wide Web offers a new model for application development in colleges and universities. At the University of Delaware, for example, Web tools are effectively being employed to produce multi-platform administrative applications. Web applications are quickly and easily crafted to interact with administrative databases, providing powerful, new functionality. Web applications cross most client platforms and can be simultaneously GUI- and character-based, reaching users of both old and new desktop hardware. Web tools are particularly suited to customer outreach efforts, delivering direct service to students, faculty, and staff. The capabilities of the Web’s HyperText Markup Language (HTML) facilitate new classes of applications, including hyper-reporting, mixed media, electronic forms, and kiosk services.

Administrative systems and customer service

The University of Delaware provides widespread access to its administrative systems, delivering improved customer service to students, faculty, and staff. The Internet’s free, public, outreach tools (World Wide Web, Gopher, and e-mail) have been merged with the institution’s closed, proprietary administrative systems (student records, human resources, and financial management).

Private, personal information, including student and employee records, is integrated with the public, general information of the campuswide information system. Freely distributed clients for DOS, Windows, Mac, UNIX, and timeshare users allow access to official, production data on both MVS and UNIX platforms. The methods employed to achieve this success are simple, inexpensive, and easily adapted.

While the administrative systems of the University can be characterized as closed, proprietary, controlled, and secure, the student view of computing is open, pedestrian, public, and wide-reaching. In keeping pace with trends toward a more student-centered campus, Delaware’s administrative systems have been reworked to place an emphasis on self-service. Self-service technologies have been applied to deliver timely information directly to the customer. These technologies empower the customer and provide cost-effective, automated services that know no geographic boundaries. Self-service technolo-

Carl Jacobson is Director of Management Information Services at the University of Delaware, where he has worked for the past eighteen years. Management Information Services has responsibility for acquisition, development, and maintenance of administrative and library systems. Mr. Jacobson’s emphasis is on expanding the boundaries of the University’s administrative systems beyond the realm of administrators to students, faculty, researchers, high school students, and parents. carl.jacobson@mvs.udel.edu

“While the administrative systems of the University can be characterized as closed, proprietary, controlled, and secure, the student view of computing is open, pedestrian, public, and wide-reaching.”
Technologies merge

With a healthy portfolio of mainframe-based administrative systems, Delaware chose to adapt existing information resources to open, network technologies, in order to meet the goals of improved customer service.

It is impossible to grant the large, expanding customer base direct access to mainframe-based information systems. Faculty and research users of “academic” machines have little desire to log on to “administrative” machines and navigate through unfamiliar territory in search of needed information. Nor is it feasible to allow 22,000 students to log on to the administrative mainframe to review grades on the day they are posted. These closed, proprietary systems must be opened to allow such “pedestrian” use. Administrative information services must be adapted to behave more along the lines of a publicly available campuswide information system (CWIS).

To meet these goals, Delaware chose to leverage existing resources by merging

- the established, closed, proprietary mainframe-based administrative systems
- the emerging, open, public, client/server-based campuswide information systems

in order to

- deliver customer service in the environment of the customer,
- do “administrative things” in “the student way,”
- allow the free, public access tools of the Internet to be used to do official university business.

The key to successfully merging these technologies is compromise. It is necessary to bring the security of the administrative environment to Internet tools, while opening the administrative systems to Internet protocols.

As Delaware first turned to the Web for administrative support, official institutional data were maintained using Software AG’s ADABAS database system and processed by programs written in COBOL and NATURAL. At the same time, CWIS information was collected, maintained, and delivered on the World Wide Web. The use of Web browsers was widespread among campus customers, while existing Natural/ADABAS systems were robust and useful. These disparate resources were combined in a unique but simple way to deliver improved information service to students, staff, and faculty.

This combination requires the transformation of the “host” of a host-terminal system into the “server” of a client/server system. The host and its associated applications become part of a client/server network enabling outreach and supporting diverse data types.

Opening closed systems

The opening of such closed systems focuses on the need for secure servers to translate Internet protocols into the languages of the administrative systems. Web HTTP (HyperText Transport Protocol) servers meet this need, functioning as effective gateways between the Web browsers and administrative programs and databases.

Such Web gateway servers may be built or bought. Several HTTP servers are available commercially at surprisingly low cost. Apple’s Internet Server and Netscape’s Commerce Server are examples of general-purpose HTTP servers that provide packaged sets of tools needed to develop Web applications. They are popular, inexpensive, vendor supported, and utilize economical hardware.

While commercial gateway servers provide the convenience of packaged toolsets, they may require additional hardware, new communications protocols, and unfamiliar programming languages. As an alternative, special-purpose HTTP servers can be developed in-house to perform these translations directly on existing hosts. Interpretive servers may be written on any networked platform, using any language supporting Internet communications interfaces.

This approach would, for example, allow COBOL programmers to open legacy systems to the Web using the tools, techniques, and training of the legacy environment. While Web browsers expect information to be packaged using HTTP, they are not concerned with how that packaging is performed.

Whether built or bought, gateway servers use standard HTTP to communicate with Web browsers on the user side. On the application side, these servers employ common gateway interfaces (CGIs) to communicate with external programs and databases. CGIs are programs or scripts, and may be written in many languages, including C, Perl, and AppleScript. CGIs allow Web servers to communicate with other servers, DBMSs, external programs, screen scrapers, and a variety of network program interfaces.

CGIs may be used in conjunction with DBMSs and programming languages to build complete, new administrative applications, or CGIs may play the role of transforming closed,
proprietary administrative systems into compelling Web applications.

With many Delaware administrative systems residing on an MVS mainframe, interpretive servers were developed to run in this environment, accept Internet packets, recognize Web HTTP protocol, and call administrative application programs based on the content of these packets.

With interpretive servers speaking to administrative programs, existing tasks such as transcript production can be reused rather than redeveloped. Upon request from a student client, the server simply invokes the existing COBOL transcript program. However, instead of printing or displaying the results, they are packaged in a Web packet and sent out onto the network.

Authentication, authorization, and encryption

In order to provide the levels of security needed in conducting personal business, authentication, authorization, and encryption routines must be employed.

With an overall design goal of “using existing resources whenever possible,” security schemes used for touch-tone registration were enlisted at Delaware to provide similar protection to the Internet clients. Student-ID and PIN (Personal Identification Number) authentication was already known and in use by students and staff. PIN-based authorization tables were already in place in existing administrative systems. (Exhibit I illustrates the Web SID/PIN authentication and access to records.)

In order to protect the authentication information as well as the private records of students, faculty, and staff, Netscape’s Secure Socket Layer (SSL) encryption protocol was adopted. This protocol was selected because of the popularity and success of the Netscape’s Web browser and because its socket-level encryption is ideal for supporting the re-use of existing authentication and authorization schemes.

SSL uses encryption to enhance user privacy by providing a communications channel that is secure against eavesdropping. When an SSL-aware browser connects to an SSL-secured server, all information passing between browser and server is fully encrypted. This secure data circuit allows existing authentication and authorization information to be safely exchanged on the network.

SSL is not the only security alternative available to those wanting to do business on the Web. Secure HTTP, Digest Access Authentication, Shen, and DCE-Web security are several examples of current Web security efforts.

Stateless client/server relationships

A significant advantage to adopting the Web-server model to provide student services lies in the “statelessness” of these servers. The transactions may be viewed as “stateless” in that a server has no lasting connection with each requesting client. The server “comes alive” upon receiving a request message, interprets and fulfills the request by passing a message back across the network, and returns to a “wait state” until the next user request comes along.

Since students do not log on to the administrative system, there is no data communications overhead. A single task monitors an Internet port and responds to customer requests. This “stateless” client/server relationship allows many customers to effectively use administrative resources without becoming members of that environment.

Without the overhead of CICS or TSO sessions, a mainframe server performs its simple tasks with little impact on the overall system. Response is immediate, even for longer packages such as student transcripts. In addition, due to the nature of the Web itself, the response time expectations of Web users are lower than those of interactive, transaction-based systems, so that if a delay is encountered, it is unremarkable.

Such interpretive servers have the advantage of accessing production data directly. They need not rely on data extracts, but instead return timely and accurate information from official production records. As students perform touch-tone drop-add, they can immediately confirm schedule changes. As students pay bills, they can quickly print summaries of charges and payments. With many business transactions reaching databases in real time, it has become necessary to report these changes in real time. “Just-in-time” production of course schedules and transcripts calls for this level of timeliness. The stateless Web server allows this to be accomplished easily and inexpensively.

At Delaware, servers have been deployed to run on MVS, UNIX, and MacOS platforms to allow information to be gleaned from various databases across campus and to take advantage of the relative merits of each operating system.

Training and support

With Web browsers already in the hands of students, faculty, and staff, the issues of training, support, and software distribution are minimized. Student grades and transcripts may be accessed in a manner familiar to all existing Web users, allowing students to use these tools to conduct institutional business (see Exhibit I), as well as to explore academic frontiers.
Client-side development costs are usually a large portion of a client/server budget. However, Web applications differ from the popular client/server model in that all Web development effort is on the server side. Since Web client tools are free and widespread, client-side costs have been kept to a minimum.

Server-side development may be as simple as re-routing the formatted-text output of a COBOL report program to a routine to place the output in an HTTP packet. In many cases, there is no need to add HTML codes to a formatted text document and no need for application programmers to learn the details of HTML.

However, HTML syntax is easy to learn and enables application developers to transform simple, pre-formatted text reports into powerful hypertext documents supporting multimedia and user input.

Software distribution

One strength of the client/server model of computing is the increased functionality provided at the desktop. Not only can Internet browsers access grades and course schedules, but they can also retrieve and display images, sounds, and even brief video clips. Any “digital object” of reasonable size can be delivered to any client workstation. This includes the delivery of client software itself.

In keeping with the goal of “self-service,” Delaware's Internet client software is stored on a Web server and made available to anyone in the campus community across the network. A simple point-and-click causes the newest version of a program to be loaded across the network to the user's hard drive.

For Web applications themselves, the bulk of processing code remains on the “server side,”
and version-control is centralized. HTTP markups are, in effect, software code that is delivered and interpreted in real time, ensuring that the most recent code changes are invoked by every user.

The Web’s hypertext capabilities provide for easy access to associated documentation for all network-delivered software.

**Classes of application**

The powerful capabilities of the Web enable the rapid develop of new classes of administrative applications. While formatted text reports such as course schedules and transcripts can easily be delivered to Web browsers, the hyperlinking and multimedia features of the Web offer exciting new potential. The Web’s hypermedia model expands the potential of administrative computing.

**Hyper-reporting**

An HTML document may be linked to any other document on the Web, creating a powerful hypertext application that may be used to produce hyper-reports. Hyper-reporting can be used to link existing summary reports and detail screens to produce effective executive information systems. Institutional executives may receive regularly generated summary reports with built-in “drill-down” capability, with links to official, detailed, production data from administrative databases.

**Mixed media**

Web hyperlinking also supports diverse data types, such as photographic or document images. Student demographic data may be gleaned from a legacy student information system, while student photographs are retrieved from a UNIX-based image server. Both could be merged seamlessly by the desktop Web browser.

**Electronic forms**

Web browsers support fill-in-the-blank forms with ease-of-use features such as scroll boxes and radio buttons. Paper forms used for routine campus business may be effectively replaced by electronic documents, available to users on all platforms and routed and processed on the campus network. (See Exhibit II, an illustration of the University’s forms home page.)

**Touch-screen, multimedia kiosks**

PO DIUM, a multimedia authoring tool developed at the University of Delaware, has been made “Internet aware,” allowing it to “speak” Gopher and Web protocols. This tool, originally designed as a classroom technology, is now used by several institutions to develop compelling multimedia, touch-screen kiosks—merging image, sound, and video with administrative information. PODIUM is an early example of an emerging class of tool, facilitating the construction of special-purpose browsers for custom Web applications.

**Evolving Web capabilities**

One of the primary strengths of the Web is the ability to deal with diverse data types—the ability to support multimedia objects. Complex data objects may be sent across the network and “unwrapped” and “displayed” at desktop browsers. In the future, these objects will become even more complex. For example, an electronic form and its associated processing rules might be delivered directly to, and processed locally on, the client workstation.

Vendor efforts, such as Sun Microsystems’ Hot Java, demonstrate the ability to deliver secure program code as an integral part of a Web transaction. This capability will redefine distributed computing, allowing host servers to deliver machine-independent code to desktop clients for just-in-time processing.

---

1 PODIUM is an object-oriented multimedia application generator developed by Professor Fred Hofstetter. For more information, see http://www.udel.edu/lynam/fth/podium.html.

---

**Exhibit II**

Groupware applications requiring collaboration, routing, and approval are ideal candidates for the Web.
Conclusion
Rapid advances in the development of tools for the Internet will impact the processes of teaching, learning, and research at our institutions. Many of these same advances will contribute to the way we conduct business and affect daily campus life for students, employees, and visitors.

The World Wide Web is emerging as a new model for administrative service on our campuses. With the application of emerging tools and technologies, existing resources can be reused effectively to return immediate benefits for small investments. Each early adopter of these technologies will gain valuable experience and insight into the issues of delivering networked services and will establish a foundation for controlled growth and change.

Development Checklist
As the Web capabilities listed below demonstrate, Web development offers many advantages over traditional application development methods.

✓ Multi-platform
Web clients exist for DOS, Windows, Mac, UNIX, and other popular operating systems.

✓ Low cost
Commercial Web browsers are available to educational institutions at no cost.

✓ GUI
Web applications may be simultaneously GUI- and character-based, delivering functionality to users of older desktop hardware.

✓ Mixed media
Web protocols support images, sounds, and video clips as well as text, allowing character-based administrative data to be merged with these rich data types.

✓ Common user interface
Although Web browsers run on disparate platforms, a certain look and feel is maintained across platforms, providing an easy-to-support common user interface.

✓ Software distribution
Web browsers themselves may be easily and inexpensively distributed across the network, using the Web itself.

✓ Self-documenting
Hypertext capabilities allow application help and tutorial routines to become an integral part of any Web application.

✓ Distributed servers
Web browsers merge information from several servers onto a single screen, without specific user knowledge of these servers.

✓ Network security
Socket-level encryption provides a secure network communications channel that can be employed to protect any existing or emerging campus authentication scheme in addition to all user data.

✓ Local processing
As Web browsers employ "helper applications" to display and process information, Web applications can therefore make use of local processes, such as spreadsheet or word processing programs.

Additional Resources
Visit Delaware's administrative Web site (http://www.mis.udel.edu/admin.html) for live demonstrations of secure business transactions. A Web version of this paper with hyperlinks to demonstrations and other relevant resources can be found at the URL http://cause-www.colorado.edu/cause-effect/cem95/cem9533.html

“The Web: A New Model for Application Development,” a CAUSE95 pre-conference seminar to be held November 28 in New Orleans, will provide an opportunity to explore, in depth, the use of the Web for application development. Seminar leader Carl Jacobson will target a general audience, addressing technical issues of interest to programmers and DBAs in the non-technical language of managers and directors. For more information about CAUSE95 activities, visit the CAUSE Web server (http://cause-www.colorado.edu/) or Gopher server (gopher://cause-gopher.colorado.edu/) or call 303-939-0315.