Secure, Web-Accessible Call Rosters for Academic Radiology Departments

Andrew Van Nguyen, Wyatt M. Tellis, and David E. Avrin

Traditionally, radiology department call rosters have been posted via paper and bulletin boards. Frequently, changes to these lists are made by multiple people independently, but often not synchronized, resulting in confusion among the house staff and technical staff as to who is on call and when. In addition, multiple and disparate copies exist in different sections of the department, and changes made would not be propagated to all the schedules. To eliminate such difficulties, a paperless call scheduling application was developed. Our call scheduling program allowed Java-enabled web access to a database by designated personnel from each radiology section who have privileges to make the necessary changes. Once a person made a change, everyone accessing the database would see the modification. This eliminates the chaos resulting from people swapping shifts at the last minute and not having the time to record or broadcast the change. Furthermore, all changes to the database were logged. Users are given a log-in name and password and can only edit their section; however, all personnel have access to all sections' schedules. Our applet was written in Java 2 using the latest technology in database access. We access our Interbase database through the DataExpress and DB Swing (Borland, Scotts Valley, CA) components. The result is secure access to the call rosters via the web. There are many advantages to the web-enabled access, mainly the ability for people to make changes and have the changes recorded and propagated in a single virtual location and available to all who need to know.

Copyright © 2000 by W.B. Saunders Company

THE PURPOSE OF THIS PROJECT was to research the possibility of using a Java applet to access a database versus a web-enabled database, using HTML. Although conceptually simple, we encountered many hurdles, ranging from differing Java Development Kit (JDK) versions to the lack of Java Database Connectivity (JDBC) drivers for common desktop databases, which proved to be very difficult. Currently, web-enabled databases seem to be the easiest solution for providing database access over the web, even though they require a commitment to a particular scripting language.

With web-based access to one "copy" of the schedule, selected personnel can modify the schedule and all users access an updated schedule. This eliminates the need for an individual or group to be responsible for keeping everyone updated on changes to the call schedule. In doing so, we

eliminate any confusion and chaos that occurs when people do not know about schedule modifications.

METHODS

For this project, we used Borland's JBuilder 2.01 development kit which employs Sun's JDK 1.1.6. We initially wrote the applet using Borland's DB Swing and DataExpress (Borland, Scotts Valley, CA) components. Upon initial use of the applet using Netscape Navigator 4.61 on a PC with a Pentium II 200-MHz processor, and Windows 98 (Microsoft, Redmond, WA), we encountered an error because a necessary file could not be loaded. The problem seemed to be with the components we were using to display the table (a JBCL component) so updating the DB Swing and JBCL components to JBCL_X and DB SwingX,¹ and migrating to Sun's JDK 1.2 plug-in resolved the problem. Figure 1 provides an example of the interface used to access the call schedule. Depending on their access privileges, users are permitted to view, post, and modify entries for their section. Figure 2 presents the dialog box used to filter call entries by section.

RESULTS

We were now dependent on Sun's Java plug-in, which was only available for the Windows and Solaris platforms. Consequently, we were faced with a new problem—the applet was no longer cross-platform compatible, which is the primary advantage of Java. By using Java, we were also committed to using JDBC² to access the database. What we did not anticipate, however, was the difficulty in obtaining an easy-to-use database with fully compliant JDBC drivers. We originally had the tables in a Microsoft Access table but were unable to find a suitable JDBC driver. Currently, we are using Interbase as our database and Interbase's InterClient as our JDBC driver. The downside, however, is the lack of a graphical user interface (GUI) for Interbase.

From the Department of Radiology, Laboratory for Radiological Informatics, University of California, San Francisco, CA.

Address reprint requests to Andrew Van Nguyen, C/O David E. Avrin, MD, PhD, UCSF Department of Radiology, 505 Parnassus Ave, Campus Box 0628, San Francisco, CA 94143-0628. E-mail: nguyen@smartpage.net.

Copyright © 2000 by W.B. Saunders Company

0897-1889/00/1302-1064\$10.00/0

doi:10.1053/jdim.2000.6912

Commands	Date	AM Readout	NeuroCrug Attending	NeuroCrug Fellow	Ultrasound	Interventional Attending	Interventional Fellow
1	7/1/99	Solito	Norman	Chuang	Meux	Gordon	Bhagat
2	7/2/99	Amin	Norman	Chuang	Goldstein	Gordon	Bhagat
3	7/3/99	Hincak	Norman	Chuang	Goldstein	Gordon	Bhagat
4	7/4/99	Reddy	Norman	Chuang	Goldstein	Gordon	Bhagat
5	7/5/99	Ominsky	Chin	Bone/Chau	Nguyen	Gordon	Bhagat
6	7/6/99	Steinbach	Chin	Kuo	Feldstein	Wilson	Bhagat
7	7/7/99	Nguyen	Chin	Nguyen	Feldstein	Wilson	Bhagat
8	7/8/99	Solito	Chin	Stufflebeam	Meux	Wilson	Bhagat
9	7/9/99	Gooding	Chin	Gronn	Meux	Wilson	Bhagat
10	7/10/99	Nguyen	Chin	Gronn	Meux	Wilson	Bhagat
11	7/11/99	Thoeni	Chin	Gronn	Meux	Wilson	Bhagat
12	7/12/99	Aranson	Fewell	Bone/Kuo	Meux	Loh	Bhagat
13	7/13/99	Coakley	Fewell	Nguyen	Nguyen	Wall	Bhagat
14	7/14/99	Higgins	Fewell	Patel	Meux	Wall	Bhagat
15	7/15/99	Solito	Fewell	Chau	Meux	Wall	Bhagat
16	7/16/99	Nguyen	Fewell	Stufflebeam	Meux	Wall	Bhagat
17	7/17/99	Higgins	Fewell	Stufflebeam	Meux	Wall	Bhagat
18	7/18/99	Steinbach	Fewell	Stufflebeam	Meux	Wall	Bhagat
19	7/19/99	Iwasaki	Fischbein	Bone/Orian	Meux	LaBerge	Bhagat
20	7/20/99	Geldberg	Fischbein	Nguyen	Carlen	LaBerge	Bhagat
21	7/21/99	Brach	Fischbein	Kuo	Meux	LaBerge	Bhagat
22	7/22/99	Webb	Fischbein	Chau	Meux	LaBerge	Bhagat
23	7/23/99	Amin	Fischbein	Patel	Meux	LaBerge	Bhagat
24	7/24/99	Hincak	Fischbein	Patel	Meux	LaBerge	Bhagat
25	7/25/99	Gahn	Fischbein	Patel	Meux	LaBerge	Bhagat
26	7/26/99	Aranson	Dillon	Bone/Patel	Meux	Kertan	Bhagat
27	7/27/99	Webb	Dillon	Nguyen	Goldstein	Kertan	Bhagat
28	7/28/99	Gooding	Dillon	Stufflebeam	Meux	Kertan	Bhagat
29	7/29/99	Thoeni	Dillon	Gronn	Meux	Kertan	Bhagat
30	7/30/99	Reddy	Dillon	Kuo	Meux	Kertan	Bhagat
31	7/31/99	Ominsky	Dillon	Kuo	Meux	Kertan	Bhagat

Fig 1. Call schedule list.

DISCUSSION

This project initially had a well-defined and simple practical purpose for addressing difficulties in posting and notifying radiology personnel of

changes to the call schedule; however, it proved to be very illuminating regarding the current status of Java and the “write once, run anywhere” philosophy. Our department happens to be dual platform (Windows 98 and Macintosh OS), and as described above, this uncovered significant technical problems that have to be overcome to successfully complete this project.

We investigated other web-enabled database applications currently deployed in our department to determine what the significant differences were from this project. One is our radiology department high interest case file server,³ running on Windows NT using 4th Dimension, a web-enabled database. Being web-enabled, the database accepts input and queries from multiple platforms; however it requires a commitment to the proprietary 4th Dimension scripting language. The call schedule application, however, takes full advantage of the power and flexibility of Java, which gives us access to a multitude of development tools and server platforms.

As of this writing, JDBC drivers for desktop database applications are somewhat scarce. We were successful using Interbase and its JDBC driver, but this database lacks the GUI design components of other products such as Access and Paradox. Finally, the applet required use of the most recent versions of JDK, which were only available for Windows and Solaris platforms. Macintosh JDK development lags, and renders the applet inoperable on the Macintosh.

Fig 2. Dialog box used to select sections.

In summary, we found that dual-platform deployment of Java applet web client/server applications that require integrated database connectivity for query and input/posting remain problematic at this time for the reasons described. We would hope and anticipate that the software industry will resolve these problems in the near future.

REFERENCES

1. JBuilder update to JFC 1.1: <http://www.borland.com/devsupport/jbuilder/downloads/java2update.html>
2. Sun JDBC driver page: <http://www.javasoft.com/jdbc>
3. Tellis WM, Andriole KP, Avrin DE, et al: Web technology in the integration of a digital teaching file at the diagnostic workstation. *J Digit Imaging* 11:117-119, 1998 (suppl 1)