eWhiteBoard: A Real Time Clinical Scheduler

David A. Berkowicz M.D., G. Octo Barnett M.D. and Henry C. Chueh M.D.
Laboratory of Computer Science,
Massachusetts General Hospital, Boston, MA.

Hospitals often have unique workflow and high-pressure environments that rely on the timely and accurate display of data. One such example is the Cardiac Catheterization Center at the Massachusetts General Hospital (MGH). This facility has a high throughput of patients and it requires careful coordination to maximize the use of its resources. In order to reach the optimum level of efficiency the staff need to be able to track the whereabouts of all their patients as well as the statuses of the procedure rooms. We felt that these activities could be enhanced with the creation of a virtual whiteboard that would display an integrated view of patient data, derived from an electronic scheduler, and room status information.

As different groups of health-care providers would be using the displayed information to coordinate activities occurring in various areas of the suite, allowing multi-user input from many workstations was key. It was clear that these requirements could be met with a web-based application. Data integrity and reliability is also an absolute requirement, and therefore failsafe mechanisms are needed to ensure the application's continuous operation regardless of network or server states.

INTRODUCTION

The Cardiology division at the MGH recently opened a new Cardiac Catheterization Center. The suite has 6 procedure rooms, 2 large holding areas, and a reception area. The layout of the suite makes communication between different groups of health-care providers difficult, and made the prior form of distributing information unworkable.

In the previous catherization center the information was displayed on a traditional whiteboard that reflected the statuses of the procedure rooms and the whereabouts of patients who were waiting to be seen, or who had already been seen. Any special instructions regarding a patient were written on the board and so transmitted to other staff members. The whiteboard was populated with information obtained from a scheduling application used by the front desk. The data were not electronically available to the nursing staff and therefore constant communication between the different work areas and the front desk was needed to ensure the validity of the whiteboard data. Similarly, changes in a procedure room's availability necessitated a physical update to the data display.

A major drawback of this system was the reliance upon the manual posting of timely information and the absolute necessity that the board was readily visible from all locations. On the other hand, this system was unaffected by network downtimes and did not suffer from software glitches.

The requirements for an electronic or virtual whiteboard can be stated as follows:

- It needs to provide the same functionality and ease of use as the original.
- It needs to be able to disseminate information to all parts of the catheterization center and be accessible on any workstation
- It needs to be interactive, so that changes in patient location or information can be rapidly posted.
- It needs to by dynamic, reflecting any changes in the schedule or room states in near real-time.

Given the above considerations and the physical dimensions of the facility, a network-based solution was clearly needed. A web-based application would be available on all workstations in the center and would provide the users with a familiar interface.

As an electronic whiteboard is essentially displaying information collected by a scheduling application, we decided to embed scheduling functionality into our virtual whiteboard. A single application could then integrate the data coming into the system from many different sites.

The application would have to display real-time capabilities to maintain data integrity. Web browsers establish stateless connections to servers in order to retrieve information, a model that is contrary to the notion of continuous updating. We therefore needed to devise innovative mechanisms to simulate this activity and yet still keep the displayed information current.

In order to automate the updating of changes in procedure room status, the application needed to integrate data provided by the 3rd party application used to record patient hemodynamics (Series IV, Witt Corporation).

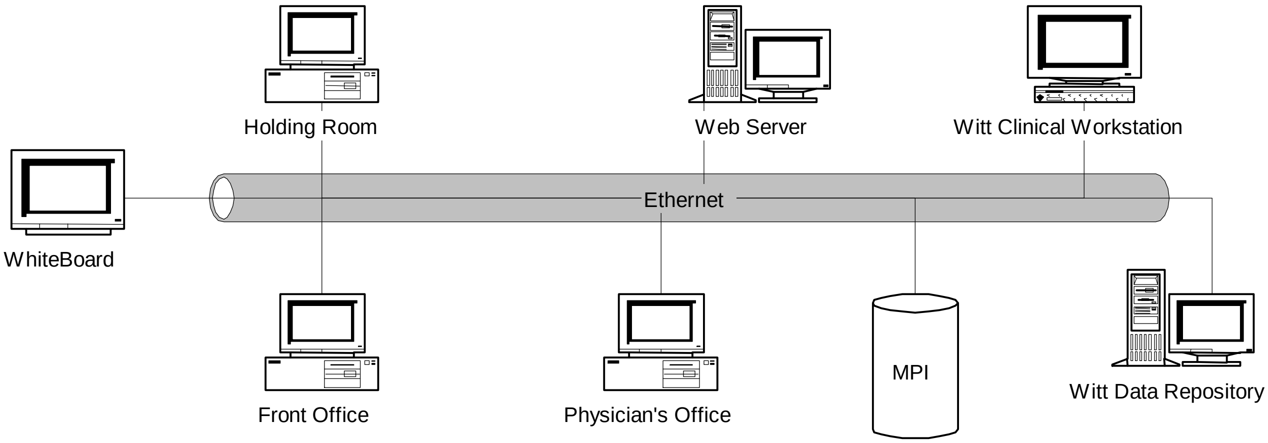

Figure 1. Overview of the Catheterization suite network.

Due to the reliance of the smooth operation of the facility on the virtual whiteboard display, a number of techniques were employed to maximize uptime and reduce the impact of network outages.

MATERIALS AND METHODS

An overview of the network showing the main components can be seen in figure 1. The master patient index (MPI) is an enterprise-wide patient database containing essential patient information needed by the scheduling application to ensure accurate identification.

The client portion of the application was written in dynamic HTML and was designed to run within a browser environment. We elected to use Internet Explorer 4.0 (Microsoft Corporation) as our standard as this is the de facto browser on all computers maintained by the computer support staff at the hospital. We used the Internet Information Server (IIS) (Microsoft Corporation) as our web-server with scripts written in JScript.

We decided not to use a relational database to store schedule data on the server, opting instead to save the information in text files. The data in the files is formatted as an eXtensible Markup Language (XML) tree[1]. Each file contains one tree and the data in that tree represents a single day. The use of XML files as a storage medium enabled us to leverage the emerging eXtensible Stylesheet Language (XSL) standard [2]. There are a growing number of XSL based parsing tools [3] that enable the transformation of XML formatted data into formats such as HTML and HL7. The transformation operation can be accomplished on either the server or the client computer and thereby proved us with tremendous flexibility in application design.

Not using a database software product reduces system complexity, as well as system cost. For reporting purposes the data will be electively driven into a relational database at set intervals so detailed analysis across trees can be performed.

RESULTS

We successfully designed a web-based application that stores schedule data in an XML format. The XML is processed according to rules defined in a set of XSL templates to produce the user interface. The modular nature of the software architecture (following a previously described design[4]) enabled the rapid creation of three distinct user interfaces (figure 2.).

Front Office Scheduler

This version of the scheduler is the most complete in that it includes a calendar and so enables the front office personnel to book appointments for future patients. The application includes a patient lookup tool that uses a server-side application to query the hospital’s MPI. These services return the information formatted in XML. Once the details of a patient’s appointment are finalized, the information is sent to the server in the form of a transaction. The server portion of the application updates the XML tree and thereby makes the information available to any other application requesting the data. The server will return an updated tree to the application if there have been any modifications to the data since the last transaction conducted by the client. The client

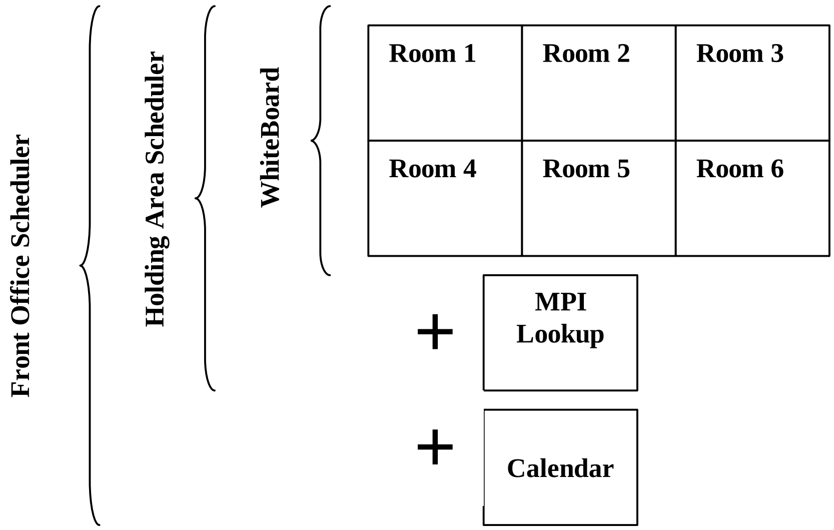

Figure 2. This diagram demonstrates the flexibility of the modular architecture of the application. The addition of extra function units to the basic whiteboard application creates two new specialized applications.

application therefore reflects the contents of the data tree at all times.

Holding Area Scheduler

This version of the scheduling application has the same functionality as the front office version but lacks a calendar component. The nursing personnel in the holding area are responsible for maintaining the smooth operation of the suite and they therefore need a tool that readily enables them to update the location of the patients. As this application is driven from the same data file as the front office application, the schedule information is keep synchronous and up to date. The application is accessible by any of the computers in the unit and so enables the free movement of the staff.

Virtual WhiteBoard

A computer with a 50-inch plasma screen display has replaced the traditional whiteboard. The computer is dedicated to run a web-browser that is constantly directed to a web-site containing the schedule data. The display is in the central holding area and provides the staff with a focal point for determining the status of procedure rooms and the location of patients.

This smaller version of the application only displays the schedule, i.e. it provides no editing facility, but due to its central role in workflow management, it is essential to ensure that the information it displays remains current. This is accomplished by forcing the browser window to refreshes itself every 30 seconds. However, this is not an adequate solution as a network or server outage can render the screen blank. Our solution to this problem was to automate the client to test the state of the network and the web-server prior to fetching the latest version of the data. If the application determines that there is a problem, it postpones the refresh and continues to test the network until it resolves that data can be retrieved.

Data Integration

A ‘transaction server’ (figure 3) handles all transactions sent via the client applications to the server. The transaction server manipulates the XML tree and ensures that the data is kept current. This is also the site where information is shared between the scheduler and the Witt system. The areas of interaction occur at two points:

1. The Witt system comes with a ‘listener’ port that is used to record admission discharge transactions (ADTs) and it uses this information to populate its database for future verification of patient identity. However, since the workflow of the facility does not fit this model our application is required to act as a conduit between the hospital’s master patient index (MPI) and the listener. The transaction server generates an ADT message by parsing XML formatted patient data into HL7 on the server and passing this via a socket connection to the “listener” device. This action occurs every time a new patient is added to the schedule.

2. Room and patient status information written to a text file by the Witt system is detected by a custom server component that passes the information to the transaction server.

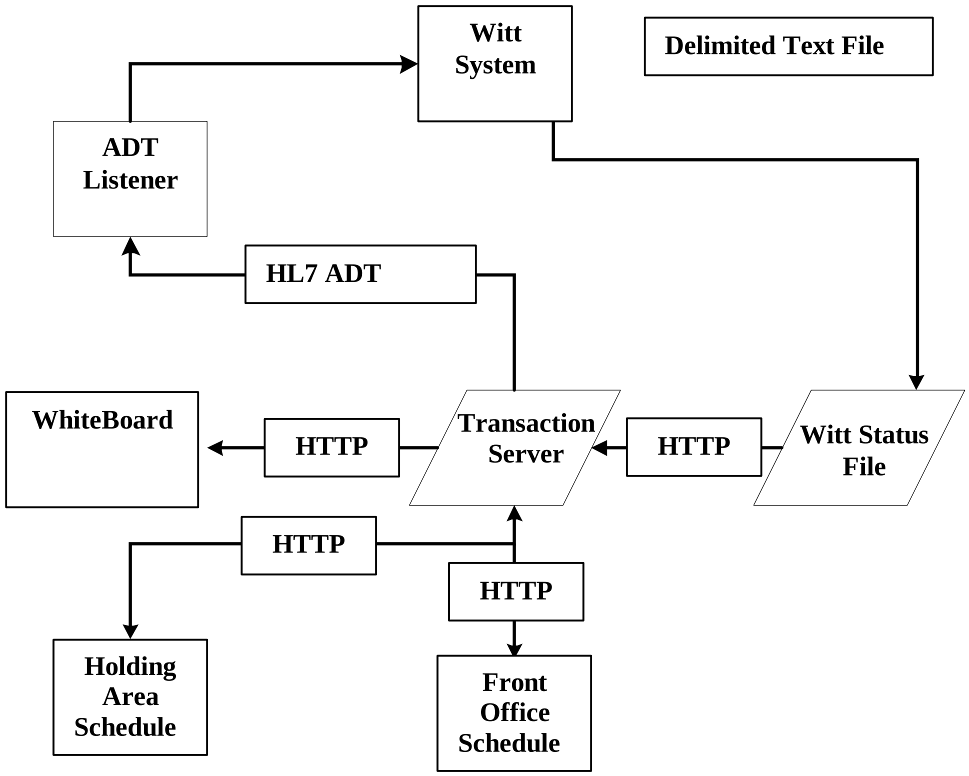

Figure 3. Schema showing the central role of the transaction server in integration information from multiple entry points.

Security

All clinical workstations require the user to logon using a hospital issued password. This information is used by the scheduling application to determine the privilege level of the user. Most physicians in the hospital are entitled to view but not edit the whiteboard, while only the catheterization staff is allowed to edit the day's schedule. Only a small handful of personnel in the front office have the ability to prospectively schedule patients.

Experience So Far

The new Cardiac Catheterization Center officially began operation March 1999 and the scheduler has been in continuous use since that time. The staff in the suite has transitioned smoothly to a web-based application. While the reliability of the whiteboard has been excellent, there have been some problems.

As expected, the integration of our software with the Witt system has been the most difficult part of the project. We have needed to work closely with their engineers to ensure that any changes made to either application did not result in unexpected and untoward result in the other. The relationship has allowed us to test and debug our software under simulated conditions that mimicked a full day of activity in the catheterization center.

Another issue revolves around network and server reliability. The whiteboard application needs to refresh its display very frequently so that the display is as current as possible. The mechanisms for detecting the integrity of the network and server can be confounded if the application attempts to refresh a page at the same time a network outage occurs. We are investigating other mechanisms to ensure that data loss does not occur.

DISCUSSION

The web has yet to be accepted and proven to provide the type of reliability needed by vital clinical systems on a 7 by 24 basis, even though there have been a number of successful attempts to devise such applications[5]. While a scheduling application is not a vital component of direct patient-care, its proper functioning can facilitate the smooth and efficient operation of a clinical unit. The demands of the cardiac catheterization center are severe with a stringent requirement for data integrity and so provide a fine test bed for this type of technology.

Since inception the application has been in continuous operation with only a minimal amount of "downtime". Over the next few months we will be closely monitoring the integrity of the network and web-server to determine the long-term reliability of the application. Routine maintenance of the enterprise-wide network mandates that there will be some unavoidable 'white-outs' but the fail-safe mechanisms we have placed on the white-board will ensure that there is no loss of operational information. Following a network "outage" the application will refresh and update itself once the network/server service is reestablished.

Machine faults are harder to predict, the dedicated computer displaying the whiteboard application has been configured to launch the web-browser application and navigate to the whiteboard URL upon reboot. In addition, the computer can be remotely administered so that minor system faults can be rapidly rectified. Should the computer suffer a catastrophic failure a replacement machine can be substituted with minimal disruption to the service.

We view this application as a starting point for the dissemination of clinically relevant data to physicians[6]. Without any major modification to the application we can allow any authorized physician access to the whiteboard. Their view can include a link to preliminary results and so facilitate rapid reporting. In addition the use of XML and XSL has enabled us to rapidly prototype different user interfaces without significant changes to core application code.

CONCLUSION

The integration of web-based applications in the clinical environment is in its infancy, yet we are seeing a greater need for the types of flexibility the web provides. The virtual whiteboard application will provide us with the opportunity to measure the acceptance and reliability of the web in high-volume high-pressure areas.

References

1. www.w3.org/DOM
2. www.w3.org/style/xsl
3. www.arborText.com
4. Berkowicz DA, Barnett GO, Chueh HC. Component architecture for Web based EMR applications [In Process Citation]. Proc Amia Symp 1998;:116-20.
5. Wang K, Kohane I, Bradshaw KL, Fackler J. A real time patient monitoring system on the World Wide Web. Proc AMIA Annu Fall Symp 1996;:729-32.
6. Chueh HC, Raila WF, Berkowicz DA, Barnett GO. An XML portable chart format [In Process Citation]. Proc Amia Symp 1998;:730-4.