Research and Teaching Access to a Large Clinical Picture Archiving and Communication System

Marc Harrison, Justin Koh, Steven Tran, Patanasak Mongkolwat, and David S. Channin

Purpose: To identify practical issues surrounding delivering digital images from picture archiving and communication systems (PACS) for research and teaching purposes. The complexity of Digital Imaging and Communications in Medicine (DICOM) access methods, security, patient confidentiality, PACS database integrity, portability, and scalability are discussed. A software prototype designed to resolve these issues is described. **System Architecture:** A six-component, three-tier, client server software application program supporting DICOM query/retrieve services was developed in the JAVA language. This software was interfaced to a large GE (Mt Prospect, IL) Medical Systems clinical PACS at Northwestern Memorial Hospital (NMH). **Conclusion:** Images can be delivered from a clinical PACS for research and teaching purposes. Concerns for security, patient confidentiality, integrity of the PACS database, and management of the transactions can be addressed. The described software is one such solution for achieving this goal.

Copyright © 2001 by W.B. Saunders Company

NORTHWESTERN Memorial Hospital (NMH) has deployed a large picture archiving and communication system (PACS).¹ After approximately 2 years of operation, there are over 350,000 studies, consisting of more than 12 million images, stored in the system. One frequent request from users of the system is for research and teaching access to the images in the database. Although the system provides for the creation of "teaching folders," there is currently no facile mechanism for transferring images from the PACS to a user's personal computer.

The PACS does provide for Digital Imaging and Communications in Medicine (DICOM) query/retrieve at both the study and patient root levels as defined by the DICOM standard.² This means that it is possible to query the PACS for a list of patients or studies and subsequently ask the PACS to deliver these images via a DICOM transfer. There are several off-the-shelf software packages that offer DICOM query/retrieve functionality that can be configured to work with the PACS. Unfortunately, there are several problems to be overcome in using these packages.

The nature of the DICOM query/retrieve services requires configuring the Move "service class provider" (SCP), in this case the PACS, with the application entity title (AETitle, equivalent to the name) and network internet protocol (IP) address

of the destination Store SCP. For any significantly large group of users, this rapidly becomes unmanageable.

For research and teaching purposes, it is often necessary to de-identify the patient information prior to delivering the images. Although some commercial packages offer de-identification, this is an additional manual step that must be performed and there is no mechanism for insuring that this step is performed. Also, commercial packages typically do not provide for management or auditing of the query/retrieve service.

Some commercial packages also provide for DICOM storage. If the DICOM gateway(s) connected to the PACS are configured to accept DICOM transfers from any calling "service class user" (SCU), then users could breach the integrity of the PACS database by sending unauthorized studies or images to the PACS.

For these reasons, we opted to develop our own DICOM query/retrieve proxy.

SYSTEM ARCHITECTURE

The NMH PACS is a PathSpeed system (GE Medical Systems Information Technology, Radiology Solutions, Mt Prospect, IL). One component of the system is the modality integration unit (MIU), essentially a DICOM gateway. This gateway provides a variety of DICOM services, including Storage and Query/Retrieve.

All of the software was developed in the JAVA language (Sun Microsystems, Mountain View, CA), running in Sun's Java Runtime Environment, version 1.2.2. The software also makes use of Sun's Java Spaces (version 1.0) and advanced imaging (version 1.0.2) packages. Certain graphical user interface elements came from the HiGrid

From the Department of Radiology, Imaging Informatics Section, Northwestern University Medical School, Chicago, IL.

Address reprint requests to David S. Channin, MD, Department of Radiology, Imaging Informatics Section, Northwestern University Medical School, 448 E Ontario, Suite 300, Chicago, IL 60611. E-mail: dsc@radiology.northwestern.edu

Copyright © 2001 by W.B. Saunders Company

0897-1889/01/1402-1030\$35.00/0

doi:10.1053/jdim.2001.23835

(version 4.0.1) package (KL Group, Toronto, Canada). JAVA was chosen for its platform independence, its numerous available component packages such as JAVA spaces and JAVA advanced imaging, and for its rapid development cycle. The Emageon JAVA DICOM (version 1.0) developer's kit (Emageon Solutions, Birmingham, AL) provided DICOM services for the software package.

The system consists of six agents: a client application, which includes a graphical user interface, a DICOM Query/Retrieve agent, a DICOM Storage agent, an Image agent, an Authentication agent, and a Logging agent.

The client application runs on the individual user's desktop computer. The DICOM, Image, Logging, and Authentication agents, as well as the JAVA space, run on a Sun Enterprise 250 UNIX server (Sun Microsystems, Mountain View, CA) in our data center. The flow of the system is shown in Fig 1.

Referring to the steps identified in Fig 1: When the client starts, it connects to a web server to obtain configuration parameters. This feature allows us to change configuration of the software

clients without having to change the client application or configuration files on the user's computer. The user is required to authenticate to our software that does an initial validation and performs a secondary validation to the PACS.

In Fig 1, step 2, the user's query is delivered to the JAVA space on the server. The DICOM query/retrieve agent retrieves the request from the space (Fig 1, step 3) and performs a DICOM Query to the GE PACS (Fig 1, step 4). The GE PACS is configured to only respond to query/retrieve requests from this agent.

The GE PACS returns the results of the DICOM Query to the query/retrieve agent, which returns the result to the JAVA Space. The user application retrieves the result of the query and displays it to the user. When a study, series, or one or more images are selected for retrieval, the query/retrieve agent instructs the PACS to perform a DICOM C-MOVE of the appropriate data to our DICOM storage SCP (Fig 1, step 5). The DICOM storage agent removes patient identifying information and then writes the de-identified DICOM image into the JAVA space (Fig 1, step 6). The Image agent retrieves the DICOM image from the JAVA Space (Fig 1, step 7) and converts it to either a 16-bit tagged image file format (TIFF) or a 16-bit portable network graphics (PNG) format. This converted image is returned to the JAVA Space (Fig 1, step 8). The user program then retrieves the image from the JAVA Space (Fig 1, step 9) and writes the image to the user's local disk in a directory chosen by the user.

All operations performed by all software components are logged by sending status, timing, and error messages to the JAVA Space. These messages are retrieved from the JAVA Space by a component, the Logging agent, and written to a log data repository, which can be mined for this information.

DISCUSSION

Research and teaching are two of the four pillars of radiology. In a film-based radiology environment, images can be photographically copied, made into 35-mm slides, and otherwise included in research and teaching activities. One of the most common requests from the users of our PACS is for research and teaching access to the images stored in the PACS.

Providing this access is nontrivial when one considers the complexity of the DICOM standard,

Fig 1. Transaction model for the system.

and the need for security, auditing, and management of the transaction. In addition, patient confidentiality and the integrity of the PACS database must be insured while providing access to full-fidelity images. When it came time to design our system, we were also faced with the challenges of making the system portable and scalable.

Query/retrieve functionality is defined in the DICOM standard.² Actually, a query/retrieve is a combination of several operations. The query client (SCU) executes a C-FIND. This causes the PACS to search its database for the patient or study in question. The PACS returns the results of the query to the query SCU. The retrieve SCU requests the PACS to perform a C-MOVE operation, which moves the specified images from the PACS to a destination. The destination is determined by a specified "name" (AETitle) of the Store server (SCP) to which the image should be moved. The PACS must look up the IP address and port that maps to the AETitle of the destination and attempt to connect to it. Thus to support query/retrieve, every PACS DICOM gateway must be configured for each potential destination. In the case of research and teaching access, the number of users could be very large and managing this configuration effort is not feasible.

In our software architecture, one software agent acts as a query/retrieve client and one software agent acts as a move recipient. This allows us to configure and manage the PACS for only one destination. It also adds a level of security in that the user is not in direct communication with the PACS and does not have access to such information as the IP address of the PACS or its components.

It is also important to prevent unauthorized storage of information to the PACS. Many DICOM gateways are configured to accept DICOM storage requests from any calling AETitle (so-called "promiscuous" mode). This feature of PACS should be disabled if users are going to be given access to other DICOM services.

In addition to removing the user from direct network contact with the PACS, we also require a user name and password in order to authenticate the user. The user name is checked against a list of authorized users of the software. The user must also be within a restricted, specific set of IP addresses. If the user name and IP address of the user are valid, then the user name and password are authenticated against

the NMH PACS by means of a JAVA database connectivity call (JDBC) to the NMH PACS.

The authentication process, all of the query/retrieve activity, and any system errors are logged via the JAVA space to a Logging agent. This allows monitoring of the system for unauthorized use and the gathering of usage statistics.

When providing research and teaching access to images, it is imperative to protect patient confidentiality. De-identifying digital images for projection or publication typically involves removing overlay or annotations and can be performed in many ways, typically using desktop image manipulation packages. This manual process can be tedious and may not always be performed. The Image agent in our system de-identifies the DICOM header information and delivers the images in formats that do not contain patient information. This solution is imperfect in that images can contain patient information "burned into" the image pixels, and in this case that information would still be visible.

Though de-identified, the images are otherwise full-fidelity, although for purposes of speed we offer the user the opportunity to download subsampled images that are smaller than the original. Often, half- or even quarter-size images may be sufficient for presentation. On the other hand, images to be processed for research purposes must be of full spatial resolution and contain all the gray scale information in the original image. It is for this reason that we provide 16-bit TIFF and 16-bit PNG images, as most medical images contain 8, 10, 12, or 16 bits of gray scale information.

With respect to the software architecture, the desktop client needed to run on the same machines that the faculty use for their office computing and research. This means the client has to run on a number of platforms, namely Windows, Solaris, and HP-UX. The server side applications needed to run on Windows and Solaris. We chose to develop the software in JAVA because of its platform-independent features and its network-centric architecture. The latter provides for a thin client that facilitates supporting a large number of users.

The system also needed to be able to scale from one user to potentially hundreds of users accessing the system simultaneously. The JAVA Space architecture provides for this scalability. A JAVA Space represents a shared, temporary object repository. The key feature is that the JAVA Space is addressable and therefore accessible by software

agents from anywhere on the network. As demand on the system increases, other query/retrieve, storage, and image conversion agents can be cloned to handle the increased load, without further software development.

CONCLUSION

Providing clinical, research, and teaching access to images in a clinical PACS is vital to support the clinical, research, and teaching missions of radiology. Many PACS can provide this

access via standard DICOM services. If this access is provided, consideration must be given to access control, security, and integrity of the PACS database. Patient confidentiality must be maintained.

At the same time, images must be delivered in a timely fashion, at resolutions up to and including the full fidelity of the acquired image. Meeting these requirements is feasible using existing software technology.

REFERENCES

1. Channin DS, Hawkins RC, Enzmann DR: North by Northwestern: Initial experience with PACS at Northwestern Memorial Hospital. *Proc SPIE* 3980:275-280, 2000

2. National Electrical Manufacturer's Association: Digital Imaging and Communication in Medicine (DICOM). Rosslyn, VA, NEMA, 1996, PS 3.1-1996-3.13-1996