OpenRIMS: An Open Architecture Radiology Informatics Management System

Steve Langer

The benefits of an integrated picture archiving and communication system/radiology information system (PACS/RIS) archive built with open source tools and methods are 2-fold. Open source permits an inexpensive development model where interfaces can be updated as needed, and the code is peer reviewed by many eyes (analogous to the scientific model). Integration of PACS/RIS functionality reduces the risk of inconsistent data by reducing interfaces among databases that contain largely redundant information. Also, wide adoption would promote standard data mining tools—reducing user needs to learn multiple methods to perform the same task. A model has been constructed capable of accepting HL7 orders, performing examination and resource scheduling, providing digital imaging and communications in medicine (DICOM) worklist information to modalities, archiving studies, and supporting DICOM query/retrieve from third party viewing software. The multitiered architecture uses a single database communicating via an ODBC bridge to a Linux server with HL7, DICOM, and HTTP connections. Human interaction is supported via a web browser, whereas automated informatics services communicate over the HL7 and DICOM links. The system is still under development, but the primary database schema is complete as well as key pieces of the web user interface. Additional work is needed on the DICOM/HL7 interface broker and completion of the base DICOM service classes.

KEY WORDS: Radiology information system, picture archiving and communication system, web, health leve7, digital imaging and communications in medicine, open source

C ONSIDERABLE MONEY, TIME, and effort are expended to electronically integrate the modern radiology practice. Much of this is owing to redundant efforts by vendors duplicating each other and also within the department itself by multiple overlapping databases that must synchronize with each other via multiple protocols. Fortunately, the efforts of a dedicated group of people have at least standardized those protocols. The Digital Communications Standard in Medicine (DICOM) is the native tongue of image acquisition devices and a PACS (picture archiving and communication system), which stores those images.[1,2] Other health-related information systems (such as hospital information systems) communicate via Health Level Seven (HL7) protocols.[3,4]

In the evolving filmless radiology department, examination orders are placed via a radiology information system (RIS). The RIS has detailed knowledge of the department resources and schedule, examination billing codes, and reports. It passes scheduled examination orders to the PACS and patient work lists to the imaging modalities. This messaging often is accomplished via a broker that converts RIS HL7 messages to the DICOM protocol for the imaging modalities and PACS. At least 3 independent database servers are involved, largely populated with redundant data. Some investigators have questioned whether it may be possible to collapse the various data repositories into one system.[5] That is the goal of the work described in this reports, and some vendors also are pursuing the concept (IDX Imaging Suite,

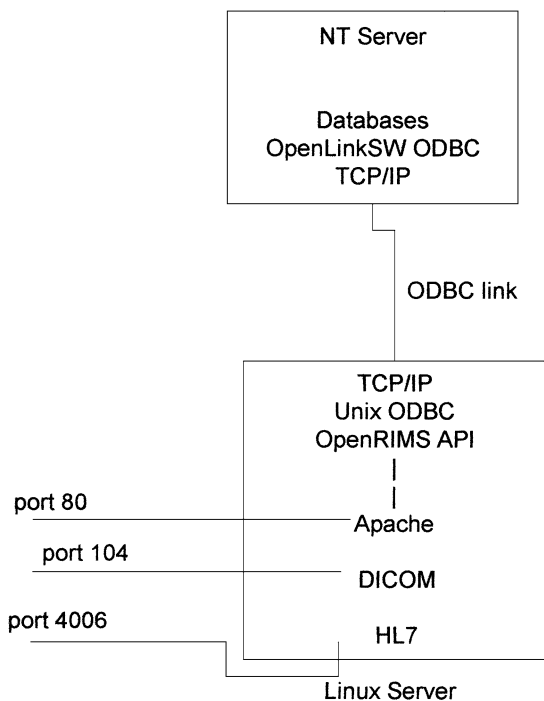*From the Department of Radiology, University of Washington Medical Center, Seattle, WA.*

Correspondence to: Steve Langer, PhD, Mayo Clinic and Foundation, Department of Radiology, 200 First St SW, Rochester MN 55905; tel: 507-266-4418; fax: 507-266-0884; e-mail: sglanger@ppsa.com

Fig 1. A total of 15 database tables are displayed. Thirteen of them are related and are linked by various keys. The 2 outstanding tables in the upper right are used to perform dictionary lookups to map HL7 and DICOM segments to the corresponding database fields. The general format of the data schema is as follows: a patient has procedures performed on them that are linked to images and reports. Images are collected under an examination that is further subspecified into series and individual images. The report is the patient's link to the radiologist (although the same schema could be used in laboratory medicine or pathology in which case the report source simply would be the pathologist for example). The report table then links to the radiologist, but the radiologist is just an employee, and all employees are instances of a particular role within the institution. Furthermore, not all roles are filled by humans (as shown by the link from the "resources" table to the "role_privs" table).

Burlington VT; GE Medical Systems, Mt Prospect, IL). The concept of the "open radiology information management system" (OpenRIMS) is to provide an integrated, open source solution to the management of all radiology data flow in one centralized, multilingual database application.

BACKGROUND

Despite its prevalence, many people are unaware of their reliance on software developed by open source or free software projects. Richard Stallman, one of the founders of the movement for free software, founded the GNU (Gnu's Not UNIX) Project in 1984.[6] The principal goals of that organization were to develop a free alternative to the UNIX operating system, which was (and still is) prevalent among colleges and universities as well as more demanding computer users. A guiding principle of that group is the so-called Gnu public license (GPL). Software copyrighted under the GPL is released under the promise that the software will be distributed freely with its source code intact so that other individuals may modify it and add their contributions back to the free software community. In a sense, the open source philosophy toward software development is no different than the concept of peer review in scientific research. Namely, peer review and open sharing of information leads to more robust outcomes.

A partial list of the free software that many use unknowingly in their daily business induces the following.

1. Sendmail: the premier routing program that still carries the majority of e-mail across the Internet. It was developed in 1981 by Eric Allman at the University of California at Berkeley.

2. Apache: the free Web server. According to a 2001 survey by Netcraft, Apache is in use by over 62% of all web sites, making it more common than all other web servers combined (Netcraft, Bath England, http://netcraft.com).

Fig 2. The database currently lives on the Windows NT server. The remaining software resides on a Linux computer. Communication between the servers is established via an ODBC link across an ethernet network. Within each computer, the software can be considered as layered. For instance, within the NT server, the database is layered on top of the ODBC driver, which communicates to the network. In a similar manner, the Linux server utilizes a layer of Python procedures to form an application program interface (API) to the DICOM, HL7, and web services that reside above them. Each of the top 3 interfaces (DICOM, Web, HL7) then communicate to the outside world on their own private TCP port, but all of them call on translation functions within the same API.

3. LINUX: another basis for a free alternative to the UNIX operating system. It was initiated by Linus Torvalds in Helsinki Finland from 1992 through 1993. What makes it somewhat different from other free software projects is that it was among the first to leverage the capability of the Internet to allow tens or hundreds of developers to cooperatively develop the software (Transmeta Corp, Santa Clara CA, http://kernel.org).

4. PERL: practical extraction and reporting language, is a scripting language often used by web sites to communicate web user input to background databases that provide information to the web user via the Common Gateway Interface. PERL was developed by Larry Wall.

5. Python: a very high level, object-oriented scripting language developed by Guido van Rossum. It can be used in web applications like PHP or PERL, but is increasingly used in more complex tasks.

6. PHP: personal home-page, is another scripting language usually used by web developers to construct dynamic content in web sites.

However, open source programs would be of limited use without open protocols by which different programs could communicate with each other. As opposed to open source programs wherein a single individual or team shares their intellectual property, open protocols are agreements among parties about how to represent information in transit and typically are developed by multivendor consortia. Examples are: HL7, DICOM, IP (Internet protocol), and HTML (hypertext markup language).

As a business model, open source raises some interesting questions; namely, how does one profit from giving away the software? Conversely, whom does a customer call for technical support? There are several answers to these questions, but a model that seems to be evolving is that companies arise that provide consulting services for users of free software (Linuxcare, San Francisco, CA; Sendmail Inc, Emeryville, CA). Indeed, even the medical industry now has a news site for open source projects.[7]

METHODS

It is a design goal to render the OpenRIMS project in the open source LAMP paradigm familiar to most open source World Wide Web developers. The acronym LAMP represents "Linux Apache MySQL Python/PERL/PHP." Linux provides the basic operating system and hosts the Apache Web server. MySQL provides the database interface to Apache and the resulting CGI (Common Gateway Interface) code can be written in any one of the 3 "P" languages. However, it proved expedient to develop the early database schema using the graphic database tools provided by Microsoft's Access (Microsoft Corp, Redmond, WA). Figure 1 shows the current database schema, and the detailed caption describes the complex interrelationships among various data tables.

In its current configuration, the database resides on a Windows NT server, while the Web server and DICOM/HL7 Python translation code resides on a RedHat Linux server (RedHat Corp, Durham, NC). Figure 2 shows the high-level data flow among various components. Communication between the Linux server and the database on the Windows server is accomplished via an ODBC (open database connectivity) bridge (OpenLink Software, Burlington MA).

Fig 3. The screen is divided into a left frame and a right frame. The left frame displays the users name and role for the institution in which they are currently working. Below that is a list of cascading menus. Only those for which the user has privileges will respond when clicked on.

The majority of human interaction with OpenRIMS (system administration, report creation and editing, exam scheduling) will be processed through the HTML interface. The user must authenticate to the system (Fig 3) and then access control rules within the database regulate what services the user can interact with. The system categorizes users as belonging to one or more roles: administrator, clerk, scheduler, transcriptionist, radiologist, or referring physician. Available operations are creation, cancellation, editing and signing of patients, examinations, reports, department resources, and users. Also, it is possible that a given user may work at several different sites and occupy different roles depending on the site at which they currently are working. During authentication, the system also requires that the login specify the current site at which the user is working. Depending on the user and the role at that site, the Apache server can be set to examine a bit mask encoded with the privileges that user has. This bit mask determines what menus are active to the user on the web pages. However, the actual permissions check is performed on the database at the time an operation is attempted. In this manner, HIPAA (Health Insurance Portability and Accountability Act) compliant security can be enforced on software agents as well as web users and stymie human agents who attempt to pass corrupt URL encoded CGI commands to the server. Operations performed via the web interface communicate with the database via CGI scripts written in Python (the OpenRIMS application programming interface or, OpenRIMS API).

It is assumed that other software agents such as order entry systems, hospital information systems, imaging modalities, diagnostic review workstations, or speech recognition systems will communicate to OpenRIMS via either the HL7 or DICOM interfaces. The interface engine (broker) is responsible for parsing the messages and submitting them to the OpenRIMS API to support such services as DICOM C-STORE, C-GET, and worklists or HL7 operations to support examination creation. Customization tables in the data base permit interfacing to nonstandard implementations as they are identified.

RESULTS

The completed OpenRIMS database design presents a consistent interface to the user, and user rights (whether human or software) are enabled or disabled based on a given user's role. In addition, the database design supports a multisite campus, where user roles may change across sites. Finally, because a user may have multiple roles at different sites, the privileges are reconstructed at each login based on a superset of all roles at that site. To date, OpenRIMS consists of the database design, introductory stages of the API, and the lion's share of the web interface for human users. Key parts of the user interface are described below.

Figure 3 shows the simple but consistent user interface (UI). The browser window is divided into 2 frames, the left of which always displays the information on the current user and contains the navigation window for all functions. The

Fig 4. In this example, a scheduler is scheduling a chest x-ray for a patient. The database populates the web form with the correct time block for this examination. For a more complex examination, the resource would be blocked out longer, perhaps 60 minutes rather than 30.

right frame varies with the selected function (login in this example). Note that the login requires not only a user name and password but also defines a drop down list that is populated with all the sites under the control of the OpenRIMS database. It is entirely possible that the same user may not only work in multiple sites but have differing privileges at those sites. Therefore, the login process requires that piece of information.

Figure 4 shows the appearance of the user interface for a key operation—examination scheduling (assuming the user has permissions to reach this page). The upper area provides a means to lookup all the examinations for a given patient and their status (ie, scheduled, completed, reported). The lower area permits the actual scheduling of a new examination. The data entry areas permit the user to "request" an examination, preferred time, and location. The database then will use a search using a "nearest neighbor" algorithm to either match that criteria or return the best alternative. The alternatives then are presented, and the scheduler may accept or decline and request another time or location.

Figure 5 shows the radiologist examination reporting page. The upper part allows setting up worklist filters by modality, body part/service, report status, time, or any combination of the former. Further, a specific patient can be searched by name, medical record number (MRN), or master patient index (MPI). By selecting a given examination in the scroll list, the examination data are synchronized to the lower window to permit report dictation. Conversely, entering an examination accession number in the lower area brings up the matching patient and all their examinations in the upper area of the window.

Figure 6 shows the user administration interface. The upper area permits viewing and updating global information about the employee (and much of these data would likely be obtained from the payroll system). Of more interest to the radiology administrator is the lower half of the window. This permits setting the roles (and implicitly the permissions) of the current user account. By selecting multiple roles, the privileges enjoyed by that user are defined by a union of the privileges assigned to

Fig 5. The upper table lists all the variables available for the radiologist to configure their worklist. For instance, they could choose to select Status "Unread," Anatomy "Body," Modality "CT," and leave the services set to "All," and the list will be populated with all unread body CT studies. In this case, the user has selected a single patient. By entering the Accession number in the lower table, patient demographics for that examination appear and a report window (lower on the screen) opens for the report.

each individual role. By selecting different sites in the drop down menu, the roles at each site can be set individually. The login authentication scheme assures that the administrator assigning these rights has access only to the access controls for employees at their institution.

CONCLUSIONS

Continuing development of OpenRIMS requires completion of the OpenRIMS API and completing the HL7/DICOM broker procedures. Where possible, existing libraries will be used such as the HL7 parser engine developed by James Rogers (HL7 lib, http://hl7lib. sourceforge.net). An advantage of the application architecture is its generality and ease in porting to other departmental information systems such as laboratory medicine or pathology with minimal effort. The completed application will act as the archive for both radiology reports and images but will not be tasked with other DICOM functions such as display and printing. For these applications, it is assumed that other third party specialized solutions will be used. Although some may not consider a PACS without display and printing to be truly PACS,

it is the intent of the author to provide a standards conformant archive to which third party, best-of-breed display workstations can be connected. In current PACS offerings, it usually is the case that the PACS workstations communicate with the archive in a non-DICOM manner. Rather, proprietary database channels usually are implemented between the workstations and the archive. This is the manner in which most vendors produce a physician's work list. A standards compliant archive, however, encourages use of DICOM standards to implement physician worklists and control workflow.

Radiology is no stranger to the open source process. In fact, in 1992 through 1994, Mallinckrodt's Institute of Radiology authored the central test node software that showed the first DICOM archive at the RSNA (Radiological Society of North America) annual meeting. Later, Mallinckrodt posted that software for free download from its computer site.[8] The tradition of developing open protocols continues with today's developments in the IHE (Integration in the Health-care Enterprise) project, which is being supported by RSNA and HIMMS (Healthcare Information and Management Systems Society).[9]

Fig 6. New users, resources, and examinations can be added via the Administration submenus. This example shows the User Administration tool. The user's "home" information is global, but other data (such as the user's role) can vary by site. Therefore, the lower table permits selecting a given site (in this case the University Medical Center) and assigning privileges and other data for that site. The lower right cell shows all the other sites for which the user has privileges.

The OpenRIMS project is a continuation of the above-mentioned tradition and is intended to show a reference platform for a simple, robust and inexpensive alternative to the complex multitiered database systems currently deployed throughout the radiology market. It is based on commodity hardware and software, and it is intended to release the completed code under the Gnu public license.

REFERENCES

1. Bidgood WD Jr, Horii SC, Prior FW, et al: Understanding and using DICOM, the data interchange standard for biomedical imaging. J Am Med Inform Assoc 4:199-212, 1997

2. Horii SC: Primer on computers and information technology. Part four: A nontechnical introduction to DICOM. Radiographics 17:1297-1309, 1997

3. Creighton C: A literature review on communication between picture archiving and communication systems and radiology information systems and/or hospital information systems. J Digit Imaging 12:138-143, 1999

4. Hammond WE: Health Level 7: An application standard for electronic medical data exchange. Top Health Rec Manage 11:59-66, 1991

5. Langer SG, Wang J: An evaluation of ten digital image review workstations. J Digit Imaging 10:65-78, 1997

6. Stallman R: Gnu's Not UNIX. 1984, http://www.gnu.org

7. Linux Medical News. http://www.linuxmednews.com

8. Horii SC: A non-technical introduction to DICOM. http://rsna.org/REG/practiceres/dicom/nontechintro.html

9. Parisot C, Channin D, Avrin D, et al: Management and presentation of grouped procedures: Has the IHE integration profile cracked the toughest radiology workflow nut? Proc SPIE Medical Imaging, 2001

OTHER RESOURCES

1. Minoru Development Corp. at http://minoru-development.com has numerous open source medical informatics programs.

2. Source Forge at http://www.sourceforge.net has 4 HL7 parser projects, 9 DICOM archive and image viewer projects

3. Clunie D: Information and tools at www.dclunie.com

4. HL7 standards are discussed at length at www.hl7.org