Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

PACS through Web Compatible with DICOM Standard and WADO Service: Advantages and Implementation

George V. Koutelakis, Dimitrios K. Lymperopoulos, *Member, IEEE*

***Abstract*-All users of informatics applications need rapid and reliable access to the kind of information that they are interested in. Web technology provides these capabilities. DICOM Standard committees recognized the necessity of a web medical standard. They specified WADO (Web Access to DICOM Object) Service, so that system interaction takes place through web, in a standarized way, allowing interoperability and proper information management inside PACS. The advantages of a web PACS comparatively with a compatible PACS are multiple and they are detected in different fields of functionality. The authors have run a project of a WADO compatible web PACS development. A web portal platform with enhanced security has been implemented. Over it, DICOM applications have been developed. JavaServer Pages (JSP) technology is mainly used to satisfy design specifications and dynamic data exchanges. Furthermore, Java applets have been developed and introduced in the whole project to serve specific demands. Evaluation results confirmed our considerations about the improvement of DICOM Services, when they are provided through web.**

I. Introduction

Users of medical information systems require rapid and reliable access to reports and images. Within computerized environments this access is increasingly based on web technologies. Clinicians need to have access either in native DICOM (Digital Imaging and Communication in Medicine) format for advanced use or into a generic format (e.g. jpeg, txt) that is able to be presented without requirement of specific medical applications. WADO specifies a web-based service for accessing and presenting "DICOM persistent" objects (images and reports). This intends to distribute medical information to healthcare professionals providing a simple mechanism. The use of web browsers at their private office or home PC is enough for obtaining and reviewing any medical information, which is stored in computer systems of laboratories, hospitals, or archiving centers anywhere in the world, provided they are authorized to view or process it.

WADO is a joint effort of DICOM (WG10) and ISO (TC215/WG2). Therefore, it is published by both organizations (NEMA04a, ISO04c). It is not a new medical communication protocol, but a "web view" in DICOM. Scope of this Standard is a new web-based Service for accessing and presenting of "DICOM persistent" objects and a simple mechanism for viewing and processing of them from web pages or XML documents, through HTTP/HTTPS protocol, using the DICOM UIDs (Unique IDentifiers for every study, series, image - instance).

Additionally, we have to take account of medical information distinctiveness. Nobody must have access to medical data without a certification of authority. The distribution of this information must become in secure environment for the protection of patient privacy. That means extra effort for the integrity and the security of data, when they are shared thought World Wide Web.

In this paper we depict a general consideration of the advantages of a web PACS (Picture Archiving and Communication System) in compare with a conventional PACS. Then, we describe the architecture and the methodology of implementation of a new web PACS, which has been constructed in our laboratory. This development mainly uses JSP technology in combination with Java applets and it is compatible with DICOM Standard and WADO Service. A short evaluation of the implementation effort follows them.

II. Web PACS Advantages

Any computer, which has the capability of running a web browser and supporting software for Java applications, can serve as a physician workstation. Physicians may change workstations in a non-predictable manner, or they may consult with each other at various workstations. A central storage design can give them better performance. Web PACS permits an aggregate record of the whole information that concerns the treatment of a medical incident, wherever it takes care. Thus, the web PACS is able to be additionally used beyond the limits of the hospital intranet using only a security protocol for the Server protection and the proper privileges for the different clients according to their specialization in the area of health.

Web-based PACS has "thin clients". As long as only a browser is needed to view images, there is no special software for clients. Clients should run a browser to access the server and it does not usually require any special hardware. All functions (window width and level, zoom, etc.) are typically provided as controls within the server application. This tends to result in workstations that are much lower cost.

The use of multiplatform technology of Java in applications development assures portability, permitting

access with any existing computer hardware system or electronic appliance (e.g. palmtops, mobile phones).

Java is an architecture neutral language. Java compiler generates Java Virtual Machine (JVM) code instead of machine code specific to the computer system, which is used. This Virtual Machine code, or Java bytecode, is in the whole architecturally neutral. The compiled code is executable on any processor and system, provided of course, that the JVM (interpreter) is installed in those systems. So, basically, the compiler generates bytecode instructions, which have nothing to do with particular computer architecture. Rather, they are designed to be both easy to interpret on any machine and easily translated into native code (C and C++).

This difference is responsible for at least two more advantages of web PACS. Native code is bigger than bytecode. That means even longer download times. Working with Java, bytecode is used instead of native code, therefore resulting in a faster download process. Also native code makes validation impossible or at least impractical, which means any hope of being protected against intentional or accidental attack on the system by misbehaving applets has given up. Java, working with bytecode, has the upper hand, accomplishing the task by being secure. Also Java permits programs to run on the client system, rather than being dependent on the server. This feature not only prevents the server from crashing in the case of error, but also avoids network congestion, giving a much faster result on applications

In web-based PACS design, images are delivered to the client PCs only on demand, and the browser does not store the images locally. The Server is able to keep up with "just in time" delivery of images and this service is able to be provided in high quality, when there is a broadband network. Of course, compression methods are often used in the web-based systems, to get around problems comes from narrow-band communication channels and telephone lines.

The Server, that accepts images from the Modalities, is distinguished from the computational machines provide services to terminal users. Therefore web PACS achieve multitasking in DICOM Services. That means PACS Server has to serve Store requests and Web Server has to serve Query/Retrieve requests.

III. WADO SERVICE

Part 18 of DICOM standard is referred, in the whole, in WADO Service specifications. Below are defined the fundamental terms of this new Service and the interaction between the communication entities of the protocol in the web environment.

DICOM Persistent Object is an instance of a data object that has been allocated a unique identifier in the format specified for SOP (Service Object Pair) Instance UID and has been chosen as an object to be saved securely for some period of time. Within the DICOM Standard, a DICOM Persistent Object is referred to as a Composite SOP Instance.

Web Client System is a system using Internet technologies (web, e-mail …) in retrieving DICOM Persistent Objects from a Web Enabled DICOM Server, through HTTP/HTTPs protocol.

Web Enabled DICOM Server is a system archiving and managing DICOM Persistent Objects and able to transmit them on request to the Web Client System.

Web Access to DICOM Persistent Objects is the Service enabling Web Client System to retrieve DICOM Persistent Objects managed by Web Enabled DICOM Server. HTTP/HTTPs protocol is used for this communication [1].

Fig. 1 shows the Request-Response Model for communication between the Web Enabled DICOM Server and a Web Client System. Query parameters are sent to the Server through the http GET and the server will respond sending one or more objects in a proper MIME (Multipurpose Internet Mail Extension) type. The MIME types, corresponding to each persistent object, which should be supported in the WADO Server response, are presented in the Table I.

In this communication architecture the Web Client System becomes very flexible. The WADO functionality supports access to "native" DICOM data and common data format (e.g. JPEG image, TXT reports). So, in PACS including WADO (fig. 2), client design can be either complex, providing advanced functionalities by taking "raw" DICOM data to be applied (for PACS Client), or simple, relying on basic features of the web browser (for web PACS Client). All functions (viewing, processing and others) are typically provided as controls within the server applications. These may be downloaded to the workstation as JavaServer Pages or Java applets, so that the only necessary software on the workstations is a web browser and Java application supporting platforms (thin software, easy to download and install). There is no special client application.

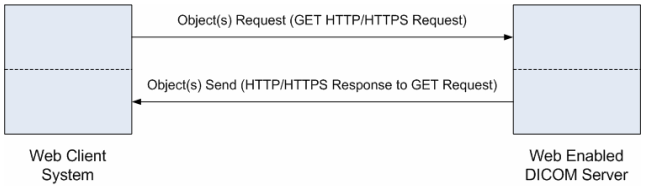

fig. 1. WADO Request-Response Model

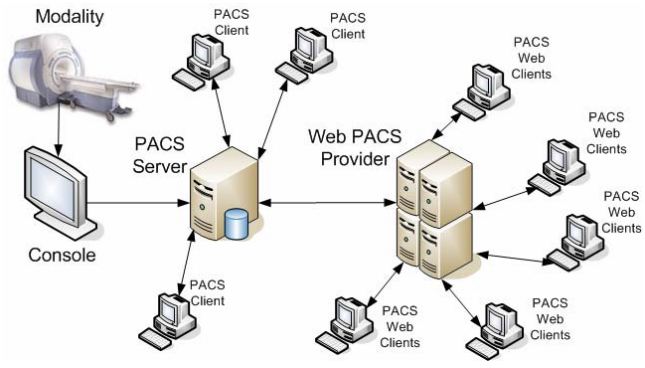

fig. 2. Web PACS network architecture

TABLE I
MIME TYPES OF WADO SERVER RESPONSE

applications	DICOM
images	jpeg, gif, png, jp2
video (for multi-frame images)	mpeg
text	plain, html, xml, rtf, CDA

IV. PORTAL PLATFORM ARCHITECTURE

The implementation of the web PACS architecture is based on Oracle web based services and applications platform, called Oracle 9i Application Server and is implemented based on a three-tier architecture (fig. 3).

The database tier comprises of databases that include data, metadata and enterprise management rules. In our system this tier includes all databases concerning the DICOM information. The user is not in direct contact with the data storage level. Every user has to retrieve the required data through the middle tier. This method is used to preserve data security and integrity. The end-user has no knowledge of the underlining processes cooperating in order to render data to him.

PACS Server, which implements the database tier, accepts images or studies (including or not report) from the modality or from the clients and has the role of the DICOM Service Provider. PACS Server has three different databases in order to serve all requests following the specifications of the protocol.

Control DB is the database, which has the responsibility to perform the communication between the PACS Server and the whole of clients (PACS clients/web PACS clients) in a formal and efficient way. Image DB is the database, where DICOM files can be stored in an efficient way. In DICOM files, information is structured in five different levels. Each one of the levels represents a DICOM entity. These are Patient, Study, Series, Image, and Instance. Report DB is the database, where DICOM Structured Reports are stored. Additionally, all information, which is essential for the association between the report and the proper study, is preserved there [2].

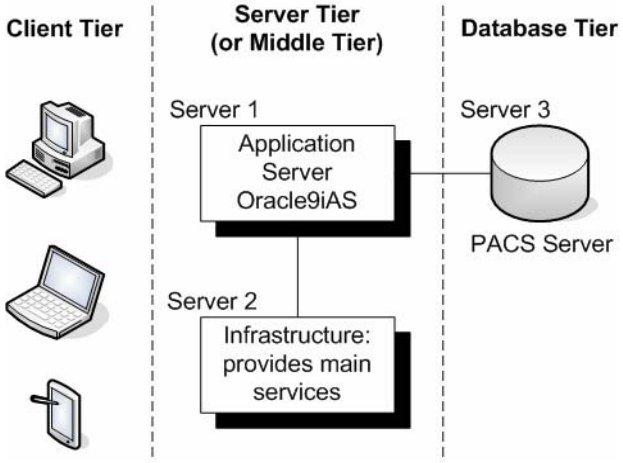

fig. 3. Three-tier architecture

The server (or middle) tier consists of the Application Server and the Infrastructure Server. It is developed using the Application Server Oracle9iAS for formalizing application and web services. This tier processes the service requests coming from the client tier. The service of a client tier request may demand access or change of the database tier's data. The server tier is the most important and complicated part of this architecture providing the technology for web application's development.

The Application Server creates an expandable, secure and embedded platform [3] permitting web content provision, web application support and communication with back-office applications. It improves the performance, scalability and availability of frequently used web sites that run on the Oracle9i platform, using the following discrete components:

1. Web Cache: storing frequently accessed URLs in memory and eliminating the need of repeatedly process requests for those URLs on the Web Server.
2. HTTP Server: based on Apache web technology, handling the http requests [4].
3. J2EE Platform: supporting applications development using: JSP (Java Server Page), Java servlets and Enterprise JavaBeans (JVB) [5].

The Infrastructure Server handles diverse procedures (e.g. the users' authentication) using data and metadata included within an internal database. This database supports also procedures of the Application Server [6].

The client tier comprises all different types of end-users that may run web browsers, web services and back-office applications. Any end-user may request to access server tier's services using different networks and telecommunication protocols. The server tier provides the end-user with all necessary information to fulfill the desired tasks through dedicated front-end interfaces of the applications [7].

The three servers of the proposed web PACS architecture are developed as follows:

1. PACS Server is an Intel Pentium IV 3.2GHz, 2GB RAM and 250 GB Hard Disc, Solaris 5.8 x86 operating system, Oracle 8.1.7 database, CTN 2.12.0 open source software for PACS functions from Washington University of Saint Lewis properly modified to support Oracle databases.
2. Application Server is a Compaq Proliant 1600, 2GB RAM, 80GB Hard Disc, MS-Windows 2000 Server operating system, hosting the web applications (Java DICOM Libraries, JSPs and Java applets).
3. Infrastructure Server is a SUN Enterprise 250, 2GB RAM, 80GB Hard Disc, Solaris 9 operating system.

V. DICOM APPLICATIONS DEVELOPMENT OVER THE PORTAL PLATFORM

The static HTML, by itself, cannot support dynamic information. Static pages cannot be configured by the data, which user enters, or come from the server resources. JSP is an easy and proper tool for expanding of HTML capabilities.

We achieve to consruct dynamic pages in our web application using JSP [8]. Essentially, we create pages in Java, however without using only Java code as in servlets. The important advantage of JavaServer Pages is the capability of distinguishing the creation of a page in two parts, the static and the dynamic. Thus, we distinguish the presentation and the business logic using JSP. In this way, we can make changes in the view of portal (colour, fonts, page structure, etc.) without affecting the basic components, as the logical procedures. On the contrary, every change in the dynamic part of the page, (thus in the JavaBean file, too) make changes in every page that is used by the specific application.

The main logic of DICOM applications' operation, over web, is based on receiving a set of commands from the users through a HTTP request [9], while they access a database for information. In such a way, they compose and process the information based on their demands and they present results in a proper format (HTTP), based on dynamic connections.

Except from JSP, we use Java applets in the implementation of the DICOM image viewer. A Java applet is a software component written in Java programming language. Java applets can run in a web browser using a JVM. Unlike a usual program, DICOM viewer applet cannot run independently. It features display and often interaction with the human user. The applet of DICOM viewer runs in a container, which is provided by a host program. The whole development of the DICOM applications constructed using Oracle Jdeveloper 10g environment.

Fig. 4 depicts the Request-Response Model of WADO Service as it is configured in our application. According to this, the user makes a request that is transferred to the server in a HTML document format. Thence, it is leaded to a server extension, as it is Jakarta for Apache Server, used in our implementation, which processes the above procedures of the server. Behind of them, there is Tomcat, which deals with page management. In more detail, the page initially goes to JSP engine. This component distinguishes the dynamic part (JavaBean) from the static part. The dynamic part, which is the code that represents it (the corresponding JavaBean file), goes to servlet engine. It will be compiled, there, to produce the desired result, which is given to the end user in a proper presentation.

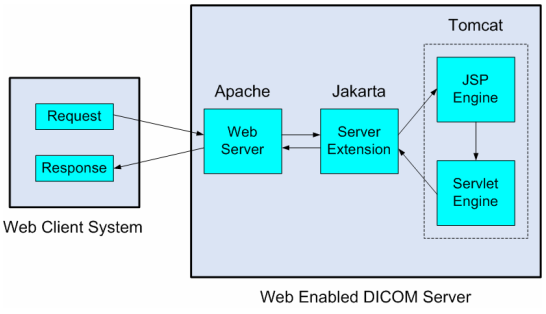

fig. 4. WADO Request-Response Model as it is configured in our web application

The design and the integration of PACS Services over the Application Server implemented using DICOM libraries developed in the Pixelmed DICOM toolkit. This is a series of libraries developed in Java and provides classes for the most of DICOM services. It also provides full support for integration of services based on WADO. Our application implements reading and transferring of DICOM data. It supports DICOM network and files. It creates an environment for display of DICOM objects, images, reports and specifications of medical objects. It also uses XML in Medical Report structuring and in other parts of the application.

In order to reduce stored-memory usage and network transfer times, we integrated "bzip2" compression utility into PACS application development. "bzip2" is a freely available pure Java tool. This compression is used to send a collection of DICOM images to another web PACS station, which should use the corresponding decompression algorithm to retrieve the images one by one (fig. 5).

VI. THE SECURITY ISSUE

The submission of the requests and the service of them, in our web PACS, are implemented over the HTTPS protocol. The part of exchanges between Web Enabled DICOM Server and Web Client System are updated with the X.509 standard, which demands the eavesdropper has the root CA (Certification Authority) and the certificate chain file. This demands a little more than a simple certificate creation for a web browser. However, this security addition comprises a great advantage. When HTTPS is used, a JNI call (Java Native Interface – communication between Java and C) happens and the C program executes the communication. This permits better performance and preserves the certified memory use of JVM.

We use HTTPS protocol in exactly the same way as the HTTP protocol. The differences are that HTTPS uses a default port number of 443 (instead of 80 for HTTP) and that HTTPS automatically performs SSL (Secure Sockets Layer) negotiation. Thus HTTPS always sends data in encrypted form. SSL uses a 40-bit key size for the stream encryption algorithm, which is considered an adequate degree of encryption for medical data exchange.

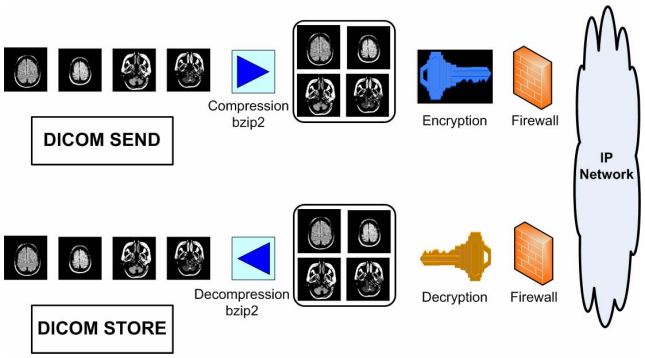

fig. 5. DICOM Send and Store using bzip2 algorithm

Moreover, Single Sign-on (SSO) part of the Infrastructure Server supports the aggregate user certification in the whole network, permitting the access in the different accounts and applications of Oracle9iAS. SSO offers a user identification process, through a log in screen. At this point a unique id and a unique password are required for any user to enter the application. On a second layer, a container, responsible for translating user's request in applications logging, accomplishes the whole logical process. A cross reference procedure with a user catalogue (Oracle Internet Directory - OID), which also belongs to Infrastructure Server, takes place. The users already registered to the catalogue are authorized to continue navigation inside the application platform. The user catalogue implements the LDAP (version 3) protocol.

VII. Evaluation

The web PACS was developed in independent steps. On a first step, the implementation of the PACS Server [2] was completed, using DICOM compatible architecture. On a later step, we developed the web portal platform [6]. In a further development (third step), we expanded the portal platform by implementing the DICOM services over web, taking into account the WADO specifications [5].

The developed system has been installed into the Patras University Campus and tested by doctors of the Patras University Hospital, doctors of a private medical center, as well as by private doctors using wireless connections. A group of evaluators performed the test process, filling appropriate questionnaires and noting their own remarks. The issues of the evaluation concerned the web PACS in comparison with compatible PACS and they are depicted in Table II [10].

TABLE II
Evaluation Measurements

Evaluation Topics	Benefit	Con	Fixed
Speed of image availability	•		
Speed of clinical decision making	•		
Overall report turnaround time	•		
Time devoted to image searching	•		
Overall personal productivity	•		
Radiologists/clinicians relations	•		
Clinicians/patients relations	•		
Overall quality of patient care	•		
Quality of work life	•		
Speed of image availability	•		
Speed of clinical decision making	•		
Time devoted to quality control		•	
Accuracy of diagnoses			•

The total evaluation process proved that the use of "bzip2" compression method improved the transferring speed of images and reports. According to test results, "bzip2"gets over the performance of other compression formats such as gzip or zip. "bzip2" gets within ten to fifteen percent of the "best" class of compression algorithms, although it is roughly twice as fast at compression and six times faster at decompression.

VIII. Conclusion

DICOM, the dominant of the medical imaging communication, already provides the standard for web development applications, which is WADO. Thus, using Java technology in different modes (JSP, applets) and Oracle portal development technology, we achieved to put out of the laboratory LAN, in the web, the services that DICOM Server can provide in an efficient way and high speed. Additionally, taking care of the importance and the privacy of medical images and reports, we used enhanced security in information access and management. This architecture guarantees for accurate performance, in combination with flexibility and functionality in providing services. Moreover, it provides a lot of capabilities for further development of extra services in the field of healthcare.

References

[1] National Electrical Manufacturers Association, Digital Imaging and Communication in Medicine, PS 3.3 - 2003, PS 3.4 - 2003 & PS 3.18 - 2004, http://medical.nema.org

[2] L. Kolovou, G. Koutelakis, G. Mandellos, A. Darras, D. Lymperopoulos, "Innovative Cooperative Scheme of Novel RIS and PACS", IASTED Int. Conf. on Biomedical Engineering, BioMED 2005, pp. 578-583, Innsbruck, Austria, February 16-18, 2005.

[3] M. Geueke, J. Stausberg, "A meta-data based learning resource server for medicine", Elsevier Science, Computer Methods and Programs in Biomedicine, vol.72 (2003), pp. 197-208.

[4] N. Nakata, N. Suzuki, Y. Fukuda, K. Fukuda, "Accessible web-based collaborative tools and wireless personal PACS: feasibility of group work for radiologists", Elsevier Science, International Congress Series, vol. 1268 (2004), pp. 260-264.

[5] G. Koutelakis, G. Triantafyllou, G. Mandellos, M. Koukias, D. Lymperopoulos, "A Web Portal that Supports PACS Applications Based on WADO Service of DICOM Standard", WSEAS Transactions on Information Science and Applications, December 2005, Vol. 2, Issue 12, pp. 2107-2114.

[6] G. Triantafyllou, G. Koutelakis, C. Boukouvalas, G. Mandellos, M. Koukias, D. Lymperopoulos, "A Web based Telemedicine Portal for centralized access to Patient Health Records", 5th WSEAS Int. Conf. on Multimedia, Internet and Video Technologies (MIV '05), Corfu Island, Greece, August 17-19, 2005.

[7] M. Shepherd, D. Zitner, C. Watters, "Medical Portals: A Web-Based Access to Medical Information", 33rd Hawaii Int. Conf. on System Sciences, HICSS-33, IEEE, January 4-7, 2000, Island of Maui, Hawaii.

[8] Duane K. Fields and Mark A. Kolb, Shawn Bayern "Web Development with JavaServer Pages", 2nd edition, Manning, 2002.

[9] I Zeng, D. Y. Fei , C. T. Fu, K. A. Kraft, " Internet (WWW) based system of ultrasonic image processing tools for remote image analysis", Elsevier Science, Computer Methods and Programs in Biomedicine, vol. 71 (2003), pp. 235-241.

[10] Guy Paré, Luigi Lepanto, Claude Sicotte, "Evaluating PACS Success: A Multidimensional Model", Proceedings of the 38th Hawaii International Conference on System Sciences, HICSS-38, IEEE, January 3-6, 2005, Big Island, Hawaii