What Digital Imaging and Communication in Medicine (DICOM) Could Look Like in Common Object Request Broker (CORBA) and Extensible Markup Language (XML)

Andrew Van Nguyen, David E. Avrin, Wyatt M. Tellis, Katherine P. Andriole, and Ronald L. Arenson

Common object request broker architecture (CORBA) is a method for invoking distributed objects across a network. There has been some activity in applying this software technology to Digital Imaging and Communications in Medicine (DICOM), but no documented demonstration of how this would actually work. We report a CORBA demonstration that is functionally equivalent and in some ways superior to the DICOM communication protocol. In addition, in and outside of medicine, there is great interest in the use of extensible markup language (XML) to provide interoperability between databases. An example implementation of the DICOM data structure in XML will also be demonstrated. Using Visibroker ORB from Inprise (Scotts Valley, CA), a test bed was developed to simulate the principle DICOM operations: store, query, and retrieve (SQR). SQR is the most common interaction between a modality device application entity (AE) such as a computed tomography (CT) scanner, and a storage component, as well as between a storage component and a workstation. The storage of a CT study by invoking one of several storage objects residing on a network was simulated and demonstrated. In addition, XML database descriptors were used to facilitate the transfer of DICOM header information between independent databases. CORBA is demonstrated to have great potential for the next version of DICOM. It can provide redundant protection against single points of failure. XML appears to be an excellent method of providing interaction between separate databases managing the DICOM information object model, and may therefore eliminate the common use of proprietary client-server databases in commercial implementations of picture archiving and communication systems (PACS).

Copyright © 2001 by W.B. Saunders Company

DIGITAL Imaging and Communications in Medicine (DICOM) 3¹ has been remarkably successful and widely accepted throughout the world as a standard and protocol for digital medical image format, communication, and data model or structure. However, some deficiencies have been identified from experience in using the standard. Also, in the interval since its initial approval and distribution, new software technologies have developed that appear to have promise for digital medical imaging, particularly common object request broker architecture (CORBA; www.corba.com) and extensible markup language (XML).²

The purpose of this presentation is to report on the simulation and testing of some concepts using CORBA and XML for the picture archiving and communication system (PACS) problem, in anticipation of using these technologies for the next major revision of DICOM.

CORBA is a technology for using software objects and their methods distributed across a network, platform-independent. Platform independence distinguishes it from simpler technologies such as Active X and DCOM (Microsoft, Redmond, WA) and various technologies for remote method invocation (RMI). Using available CORBA development kits for various platforms, CORBA-compliant objects can be created and compiled, then deployed on a server by registering them with an object request broker (ORB) installed and accessible from the same network (local area network [LAN] or wide area network [WAN]).

The most obvious example use of this technology is to create a storage object(s) and test its performance in responding to requests to store images from a (simulated) imaging device.

The second issue addressed is the general agreement that the DICOM data model needs to be more compliant with existing database technologies. Over the past several years spanning the time that DICOM 3 was approved, until recently, the most common relational database technology was structured query language (SQL; www.inprise.com). Unfortunately, for various reasons, SQL was rejected as an approved standard by the DICOM Committee. Today, the choice of technologies for database interoperability includes XML. XML is a method that allows data to be essentially self-describing in terms of its content. An industry, such as medical imaging, can agree on a document type definition (DTD) that defines <tags>

From the University of California at San Francisco.
Address reprint requests to Andrew Van Nguyen, Department of Radiology, University of California at San Francisco, 530 Parnassus Ave, Box 0628, San Francisco, CA 94143-0628.
E-mail: avnguyen@ucsf.edu.

Copyright © 2001 by W.B. Saunders Company
0897-1889/01/1402-1022\$35.00/0
doi:10.1053/jdim.2001.23827

```

<!DOCTYPE CT_IMAGE[
  <ELEMENT IMAGE_ID (#PCDATA) >
  <ELEMENT Patient (MRN, DOB, Sex) >
  <ELEMENT Study (Date, Time, ID) >

  <ENTITY % Data "(MRN, DOB, Sex, Date, Time, ID)" >
  <ENTITY %Data; (#PCDATA) >
]>

```

Fig 1. XML example: document type definition for patient and study demographics.

for data items such as medical record number, patient name, examination type, examination number, date, etc, as shown in Fig 1. Databases can output records and reports in XML format. Parsers are available to then insert this "published" or transmitted data into a second database.

The model we explored was that of a simulated computed tomography (CT) scanner, looking for an available storage object to store the images and associated demographic and image data from a study into an archive and associated database.

This technology appears to offer promising solutions to several problems: simplifying the DICOM standard to tech advantage of new software technology to perform "associations" between components (application entities).

MATERIALS AND METHODS

This project was developed on the Windows 2000 (Microsoft) platform using JBuilder and Visibroker (Inprise, Scotts Valley, CA) for Java as the ORB. Initially, we wanted to send an image from one computer (the CT scanner) to another (the storage server) using CORBA. A "server" was created to provide a simple storage method which wrote the image to disk. The server would instantiate itself and then register itself with an ORB sitting somewhere on the network. CORBA allows the ORB to reside on any computer within the network so it was not necessary to know which physical computer was handling the storage operation. After registering itself with the ORB, the server is ready to handle any requests. A client program was also written to simulate the CT scanner. The client was sending a JPEG image of an orchid, which could have been any kind of image, in any format. The client would query the ORB for an object matching a specified description. The ORB would look through its repository, seeing that there was at least one registered "server" of the correct type. The ORB would then allow the client to call the storage method from the server. The image would then be transferred and the ORB would be ready to handle another request.

The "coordinator" and "storage servers" were first written in an interface definition language (IDL) file; this allows the implementation to be written in any CORBA-supported language and platform. In this case, we chose Java, though C++ could have been used without any difficulties. These files offered a skeleton for the implementation so that programs written in other languages would not have a compatibility program. The IDL files were run through a program provided with Visibroker called *IDL2Java*, which created *.java files. Basically, the *IDL2Java* program provides function prototypes and a conversion between data types, which vary from language to language and platform to platform.

Currently, the DICOM standard issues a header with the image being stored. The header contains information on the patient and the image being transferred. We wanted to investigate the usage of XML to transmit the header information. XML provides a simple way of transmitting data because of its self-describing nature. The following XML tag: "<MRN>1234</MRN>" could be used to transmit the medical record number of a patient without the need for complex SQL statements. For the header, we decided to include patient information (medical record number, date of birth, sex) and image information (date, time, ID); all images were simulated CT images. This information could be encoded with the image and extracted by the storage server, although we chose to send it separately as a string. The XML text could then be parsed and input into a local database on the CORBA storage server. A DTD provides a description of the content model. Similar to hypertext markup language (HTML), XML has "tags"; these tags are used to describe data instead of presentation, as in the case with HTML. The DTD basically provides the "rules" that the tags must adhere to. DTDs could be included within an XML file or stored in a separate file. Since we were transmitting the XML information as a string, the DTD was stored in a separate file.

The storage server can then be queried through its local database for retrieval.

RESULTS

The initial tests to store simulated CT scans on a CORBA storage object worked as planned. At this point, we wanted to know if it was possible to have a pool of servers that could handle the storage tasks, to address the distributed/single point of failure issues. A coordinator was implemented with which the servers would have to register, in addition to making itself known to the ORB, as shown in Fig 2. However, this did not have the effect we envisioned. Since the coordinator was not multi-threaded, it handled requests one at a time. This resulted in the same storage server handling the requests since the coordinator would not process the next storage request until the previous one was finished. Several methods for making the coordinator multi-threaded were attempted. One involved a pool of

Fig 2. CORBA distributes storage server objects with coordinator.

servers waiting to be called upon, but the coordinator itself was not multi-threaded and doing so was seemingly difficult and inefficient. Another method approached was to use a first-in-first-out (FIFO) queue. One queue would hold the available servers which would be removed as they were used and placed back in the queue as they were done with their assignment. Another queue allowed the coordinator to be multi-threaded. As storage requests came in, they

would be added to the queue. The coordinator ran in its own thread, removing items from the queue and creating a new thread each time. If the queue were empty (there were no pending jobs), the coordinator would wait until a request came through. This allowed the storage requests to run simultaneously, maximizing efficiency within the storage objects.

DISCUSSION

This project demonstrates the potential of new software technologies, specifically CORBA and XML, to address some of the deficiencies in DICOM 3. The complexity of association between application entities can be simplified with CORBA. In addition, CORBA can readily support a distributed storage architecture to help protect against single points of failure. XML shows great promise for enriching yet simplifying the structure and interoperation of the DICOM data model. These or similar technologies should be utilized in the next major revision of DICOM.

REFERENCES

1. Digital Imaging and Communications in Medicine (DICOM) Standard. Washington, DC, National Electrical Manufacturers Association (NEMA), 1996
2. Bosak J, Bray T: XML and the Second-Generation Web.