LA-UR- 98-2490

Approved for public release; distribution is unlimited.

Title:

TELEMED: WIDE-AREA, SECURE, COLLABORATIVE OBJECT COMPUTING WITH JAVA AND CORBA FOR HEALTHCARE

Author(s):

David W. Forslund, Ph.D., CIC/ACL
James E. George, Ph.D., CIC/ACL
Eugene M. Gavrilov, Ph.D., CIC/ACL

Submitted to:

Seventh IEEE International Symposium on High Performance Distributed Computing

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

TeleMed: Wide-Area, Secure, Collaborative Object Computing with Java and CORBA for Healthcare*

David W. Forslund, James E. George, Eugene M. Gavrilov
Los Alamos National Laboratory

ABSTRACT

Distributed computing is becoming commonplace in a variety of industries with healthcare being a particularly important one for society. We describe the development and deployment of TeleMed in a few healthcare domains. TeleMed is a 100% Java distributed application build on CORBA and OMG standards enabling the collaboration on the treatment of chronically ill patients in a secure manner over the Internet. These standards enable other systems to work interoperably with TeleMed[1] and provide transparent access to high performance distributed computing to the healthcare domain. The goal of wide scale integration of electronic medical records is a grand-challenge scale problem of global proportions with far-reaching social benefits.

1: Introduction

The value of distributed computing in the commercial arena has become obvious with the explosion of the Internet. Although the domain of healthcare has been traditionally dominated by single-vendor, proprietary, mainframe systems, the interest in the Internet and the desire to provide more cost-effective healthcare solutions has resulted in a rapid movement toward more flexible, scalable, component-based systems. One of the leading organizations in this effort is the CORBAmed[2] Healthcare Taskforce of the Object Management Group (OMG), which is a non-profit organization of over 850 members developing standard interfaces for distributed object services and infrastructure at the platform and domain (vertical market) levels. The effort builds on existing standards (such as Health Level Seven, HL7 [3]) as well as advanced system development that is going on in research organizations. TeleMed utilizes and helps to drive these standards by demonstrating the utility of advanced distributed object systems in a non-commercial environment. We are attempting to work with industry to show the utility of collaborative computing in healthcare[4], thus greatly extending the traditional concept of telemedicine. In addition, the infrastructure enables the utilization of advanced, high performance serves in a transparent manner to a healthcare provider.

Although the OMG's CORBA (Common Object Request Broker Architecture) specification itself is language and platform neutral, we have chosen to develop all of the components in TeleMed with Java because of its portability and rapid development environment. We describe in this paper the architecture we have used, some of the relevant CORBA standards and how they are evolving, some details of the security implementation and the overall functionality and usability of the TeleMed software. Finally, we discuss the impact of this technology on solving some of the global healthcare issues.

2: Architecture

Although there are a number of efforts underway to deliver electronic medical records via the Web[5], we believe a more comprehensive architecture is needed to provide electronic medical records over a wide area network. The TeleMed project is an attempt to help define at least portions of this architecture by demonstrating the value of distributed objects in healthcare.

TeleMed is designed as a distributed object system in which the various healthcare components are dealt with as objects and distributed via the CORBA standard. Java has been used to implement both the client and server side to facilitate portability and ease of development. The strategy is to enable a client to discover and communicate with a variety of servers in such a way as to enable the assembly of a virtual patient record on the client from the

* Work performed under the auspices of the U.S. Dept of Energy and with funding from the U.S. Medical Research and Materiel Command

multiple sources. The information can have many modalities including images and sound.

In our current configuration, there are at least three servers that must be connected. The first is the authentication server that validates the user and manages the access policy for various servers. This is discussed in more detail in Section 4. The second is a Person Identification Server that utilizes a variety of demographic information to uniquely identify a patient who is the subject of treatment. The current implementation is based on the CORBAmed standard that has recently been adopted by the OMG. Following successful identification of the patient, the client connects to a MedLib server that provides the relevant clinical information for that patient. There might be multiple MedLib servers feeding information to the client. There is no patient identifying information stored in the MedLib server to facilitate epidemiology studies as well as outcomes analysis. The MedLib server is a CORBA interface that provides simple representations of the objects on the server that can be navigated from the client.

The server architecture is designed in such a way that multiple database technologies can be used with no change in the overall system behavior. At the lowest level we use JDBC to connect to relational databases in a portable manner. Above that level we use (as much as possible) the ODMG (Object DataBase Management Group) Java bindings so that various object databases can be easily used in addition to relational databases. Objects are mapped from the persistent store to the CORBA object model in the server before sending it to the client. This enables one to map from a data representation in the database that may be very different from that provided to the client. The adaptation of the system to a variety of data sources can thus occur with no change in the client's functionality.

3: CORBA Standards

The OMG effort in the early 90's was focused on providing the low level infrastructure and services to support the remote access of objects. In the last few years, however, the work has expanded to numerous vertical markets, including healthcare. The healthcare taskforce in the OMG is known as CORBAmed. The primary mission of CORBAmed is to improve the quality of care and reduce costs by use of CORBA technologies for interoperability throughout the global healthcare community. CORBAmed defines standardized object-oriented interfaces between healthcare related services and functions. These interfaces serve to promote interoperability between a variety of platforms, operating systems, languages and application.

The design of each Object Service uses and builds on these CORBA concepts:

- Separation of interface and implementation
- Object references are typed by interfaces
- Clients depend on interfaces, not implementations
- Use of multiple inheritance of interfaces
- Use of subtyping to extend, evolve and specialize functionality

The first approved standard coming from CORBAmed is the Person Identification Service. Its goal was to provide a standard method of locating person identifiers and their associated records across facilities and enterprises, subject to the confidentiality concerns and the right for anonymous care. This provides for a flexible set of traits that can be matched with varying quality as well as providing for correlation of person identification across multiple domains. TeleMed implements this standard so that multiple, diverse systems can be linked together for patient location information.

The second standard to be approved is the Lexical Query Service that supports the use of multiple vocabularies in a heterogeneous application environment. Other standards underway include Decision Support Services, Clinical Observation Access Services, healthcare-specific security issues, linking with HL7, biomedical imaging, and pharmacy.

Although far from perfect, we believe the CORBA architecture has great potential and provides an excellent venue for research groups to work together with industry to develop advanced distributed computing tools which can be useful both to industry and academia. There are a number of working groups and special interest groups (SIGS) that are adopting this model for the advancement of open systems architectures [6].

4: Security

Security is a interesting problem in healthcare because providing robust security is commonly perceived as being in opposition to usability and timely delivery of services. If security substantially delays the access to relevant information, health care providers tend to go around the system, further degrading the overall security. Any system to be used in healthcare must take this issue into account to provide robust security with ease of use. In addition, the complexity of varying roles of healthcare providers also makes the security management difficult.

Maintaining privacy and confidentiality when electronic information systems make data much more accessible is of primary importance, too[7]. Finally, the lack of standardized security policies in healthcare compromises the ability to provide a wide-area integrated electronic medical record.

The CORBA architecture has defined a security service called CORBAsec that is in the process of being implemented by a variety of vendors. The standard is very comprehensive and supports authentication, access control, security auditing, secure communication, non-repudiation, and administration[8].

However, because of the delay in delivering security services for Java applications, we have implemented our own security services in pure Java that are compatible with the security services when they become available. This design provides a high level model which will be adequate in the near term and can be enhanced with the security module when it is available. For security, the philosophy is to keep all of the user identification, passwords, and access policies on the server side and expose as little as possible directly in the IDL.

Along with the authentication, every server (including the authenticator) is expected to manage the policies of its data and a user list class is provided to implement policies, along with a simple encryption class for rudimentary secure communications. Further, the servers are expected to implement some of the authenticator IDL.

A secure server/client implementation can be created by:

1. Implementing the authenticator and servers with policy management and using SSL between all connections.
2. Implementing the authenticator and servers with policy management and using COSSecurity services for all connections.
3. Implementing the authenticator and servers with policy management and using encryption/decryption and CORBA::principal as a certificate on an open net.

We have implemented such a system with public/private key infrastructure in both hardware and software. The hardware example has the advantage of preventing duplication of the private key and enabling an easy to manage portable key infrastructure. The hardware we have used is the Cryptographic iButton from Dallas Semiconductor, but other smart-card technologies could also be used. Since the encryption is done selectively, we are able to control the performance penalty of the secure communication channels.

Although we must manually intercept the method invocations to validate the user policy, we are able to provide sufficient granularity of access to flexibly meet the various security requirements in healthcare. The mechanism we use is extremely portable and has no impact on a server's IDL for the specific service provided. In addition, a single login is supported so that the system has a minimal impact on the usability.

5: Functionality

The TeleMed system is designed to be fairly easily accessible by healthcare providers[9]. Although the client software can be distributed through a web browser at the time of use, we prefer to deliver it separately from the web browser to reduce the performance and security problems. Separate distribution of the client, enables the system to be easily used over the plain old telephone system (POTS), since only the relevant information needs to be moved over the low speed lines. In addition, the Java client can readily communicate with multiple servers which is more difficult when running inside of a browser. Finally, the session security is easier to manage, since one has more control of the Java Virtual Machine (JVM).

The initial communication of the client is with an authenticator to which the servers have also registered. Once the secure authentication has occurred, the user is presented with a screen for searching for the appropriate patients to be considered for treatments and the connection to the relevant medical record servers. Once connected, having been validated for access, the complex healthcare information is displayed in a graphical form to enable rapid comprehension of the treatment history of the patient, including the immunization history. Any individual elements of the medical record can be navigated to a greater depth by simply clicking on the appropriate icon.

Multiple users can simultaneous view the electronic medical record and view the information provided by another providers in near real time. Thus we have a true collaboratory within the electronic medical record itself.

The system is being tested at the National Jewish Medical and Research Center in Denver, Colorado for chronic, mycobacterial infections that must be maintained over a long time. It is also being used in a rural telemedicine project in rural northern New Mexico following clinical encounters and immunization records, as well as by the Dept of Defense at the US Medical Research and Materiel Command in Fort Detrick, Maryland.

The implementation of the system in Java has enabled us to reduce the footprint of the system so that it can run on a compact flash memory card for extreme portability. This enables the individual to carry a personal electronic medical record that is fully networkable with the more permanent long-term electronic medical record. The value of this is currently being explored in the military and for use in patient-managed home healthcare.

6. Conclusions

We have found that Java/CORBA infrastructure provides a flexible, good performing environment for managing multimedia electronic records in a variety of domains. In addition, the standards for sharing distributed objects is evolving rapidly so that the goal of being able to securely assemble a virtual electronic medical record appears to be within reach. With that a near reality, we can foresee the situation in which a physician can have a substantial assist in the treatment of their patients by being able to have the anonymous clinical histories of similar patients at their finger tips. This availability could enable them to determine in almost real-time the best clinical procedures and to discover the emergence of new infectious diseases in a timely manner. This new era of the use of advanced information technology by physicians should allow a widespread use of cost-effective high performance computing for automatic or semi-automatic diagnosis to become commonplace. We believe that improved quality of healthcare will be the clear result of these developments.

References

[1] D. W. Forslund, R. L. Phillips, D. G. Kilman, J. L. Cook, "Experiences with a Distributed Virtual Patient Record System" In *Proceedings of the 1996 American Medical Informatics Association Annual Fall Symposium, Symposium* (Washington, D.C., Oct. 26–30). J. J. Cimino, Ed. American Medical Informatics Assoc., Hanley & Belfus, 1996, pp. 483-487.

[2] Object Management Group, Ed. Common Object Request Broker Architecture. Wiley-QED, 1991. See also: The OMG CORBAmed web page: http://www.omg.org/corbamed.

[3] David G. Kilman and David W. Forslund, "An International Collaboratory based on Virtual Patient Records" *Communications of the ACM*, Vol. 40, No. 8 (1997), pp. 110-117.

[4] cf., http://www.hl7.org for charter and scope of the Health Level Seven organization

[5] Kohane, I. A., et al. "Sharing electronic medical records across multiple heterogeneous and competing institutions". In *Proceedings of the 1996 American Medical Informatics Association Annual Fall Symposium* (Washington, D.C., Oct. 26–30). J. J. Cimino, Ed. American Medical Informatics Assoc., Hanley & Belfus, 1996, pp. 608–612.

[6] A complete listing of the OMG Task Forces and SIGS can be found at http://www.omg.org/about/sigs.htm .

[7] Donaldson, M. S., and Lohr, K. N., Eds. Health Data in the Information Age: Use, Disclosure, and Privacy. Institute of Medicine, National Academy Press, Washington, D.C., 1994.

[8] cf: http://www.omg.org/docs/formal/97-12-22.pdf

[9] James L. Cook, Mimi N. Hackley, Richard L. Phillips, James E. George, and David W. Forslund, "NJC/LANL: Improving Patient Care with Java", *Java for Business*, Thomas Anderson, Ed., Van Nostrand Reinhold, 1997, ch. 8, pp. 173-199.