

LA-UR- 96 - 34.92

Title:

TELEMED: AN EXAMPLE OF A NEW SYSTEM
DEVELOPED WITH OBJECT TECHNOLOGY

CONF-9610132--1

Author(s):

David Forslund
Richard Phillips
Al McPherson
Bob Tomlinson
James George
David Kilman
Francisco Reverbel
Juhnyoung Lee
Mohamad Ijadi
Jonathan Greenfield
Jim Cook
John Newell

Submitted to:

Conference on Object Oriented Programming Systems,
Languages and Apps. October 6-10, 1996,
San Jose, CA

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

Form No. 836 R5
ST 2629 10/91

***TeleMed*: An Example of a New System Developed with Object Technology: ⁺**

David Forslund, Richard Phillips, Bob Tomlinson, Al McPherson, James George, Francisco Reverbel, Juhnnyoung Lee, David Kilman, Mohamad Ijadi, Jonathan Greenfield, Jim Cook*, John Newell*

Los Alamos National Laboratory

Los Alamos, NM 87545

Introduction

The *TeleMed* application grew out of a relationship with physicians at the National Jewish Center for Immunology and Respiratory Medicine (NJC) in Denver, Colorado. These physicians are experts in pulmonary diseases and radiology, helping patients combat the effects of tuberculosis and other lung diseases throughout the Nation. These individuals are an expensive and scarce resource, who often travel around the country to share their expertise with other physicians.

To make the knowledge and experience at the National Jewish Center available to a wider audience, Los Alamos National Laboratory has developed a virtual patient record system¹ called *TeleMed* which is based on a distributed national radiographic and patient record repository located throughout the country. Without leaving their offices, participating doctors can view clinical drug and radiographic data via a sophisticated multimedia interface. For example, a doctor can match a patient's radiographic information with the data in the repository, review treatment history and success, and then determine the best treatment. Furthermore, the features of *TeleMed* that make it attractive to clinicians and diagnosticians make it valuable for teaching and presentation as well. Thus, a resident can use *TeleMed* for self-training in diagnostic techniques and a physician can use it to explain to a patient the course of their illness. In fact, the data can be viewed simultaneously by users at two or more distant locations for consultation with specialists in different fields. This capability is of enormous value to a wide spectrum of healthcare providers. It is made possible by the integration of multimedia information using commercial CORBA technology linking object-enabled databases with client interfaces using a three-tiered architecture².

Capabilities

Some of the capabilities of *TeleMed* can be illustrated by looking at a series of user interface components that are available to the user. The user begins a *TeleMed* session by selecting a database site from the interface shown in [Figure 1](#). This sets in motion an CORBA transaction for vending all patient record objects from the selected site to the requesting client, shown listed in [Figure 1](#) as Patient #1, Patient #2, etc. The databases have registered the fact that they have a particular patient's data with an ID-server which maintains the links between the various databases in which the patient information may reside.

⁺ Work performed under the auspices of the U.S. Dept. of Energy

* National Jewish Center for Immunology and Respiratory Medicine, Denver, CO.

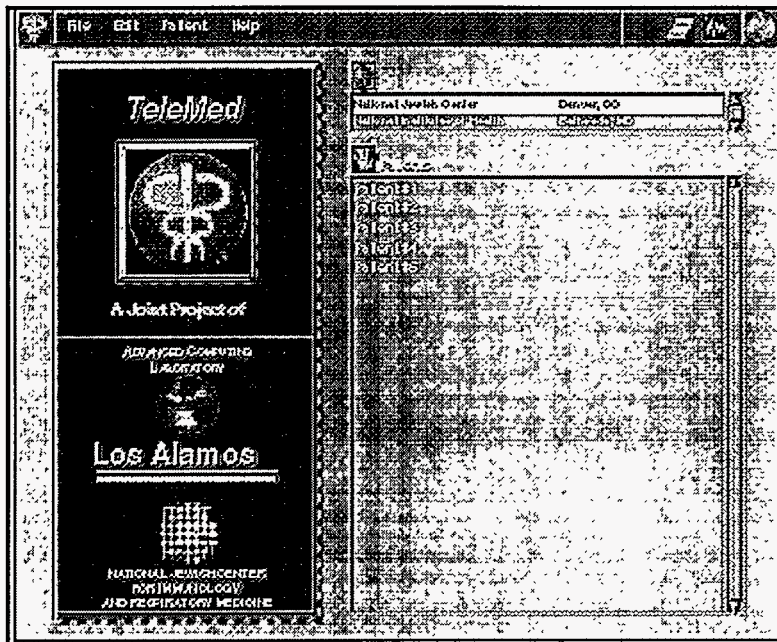


Figure 1 Initial TeleMed Interface

To understand the coordination of distributed object activities with user interface, [Figure 2](#) is a graphical representation of TeleMed objects. In this diagram the arrows represent an inheritance relationship and the other lines represent a reference or containment. Textual data from the Patient object, i.e., the patient's name, was retrieved and used to populate the Patient list in [Figure 1](#).

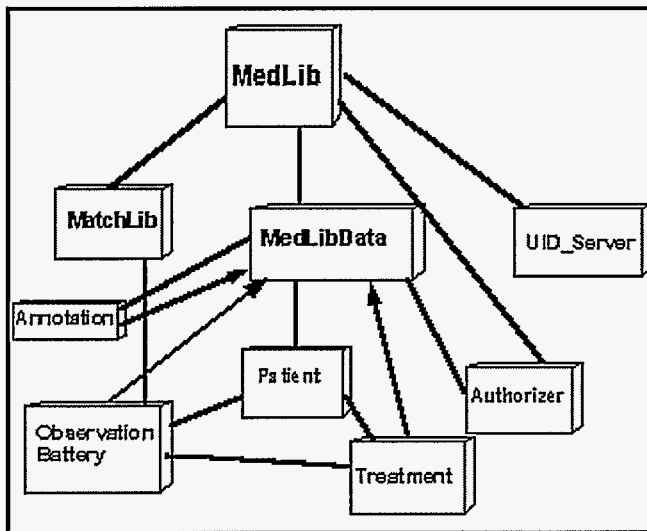


Figure 2: TeleMed Objects

A patient's treatment record appears by double-clicking on the patient's name in the interface in [Figure 1](#). The user interface manifestation of the Observation Battery object in [Figure 2](#) is called a Graphical Patient Record (GPR) and is shown in [Figure 3](#).

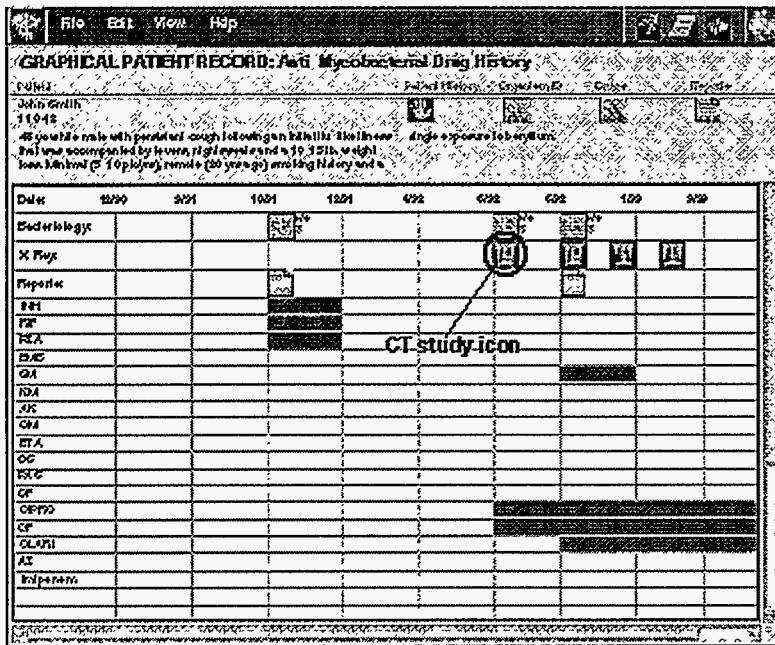


Figure 3 TeleMed Graphical Patient Record

The GPR is an excellent example of a TeleMed media-rich document and distributed object technology. The GPR is a virtual document, a patient record that is empty until it is dynamically populated by requests for distributed objects. The MedLib object in Figure 2 contains the information necessary for "harvesting" this data from all appropriate sites. Thus, laboratory reports may be retrieved from the National Institutes of Health near Washington, DC while radiographic data may reside at the NJC in Denver. So, for example, when all patient data are retrieved, icons representing laboratory tests, radiographic studies, drug treatments, etc. are drawn on the GPR template. Each of these icons is mouse-sensitive and, when clicked, call up additional user interfaces and related data.

Before looking at these interfaces it will be helpful to know more about what goes on at the distributed objects level. In Figure 4 we show the relationships between the client process (TeleMed GUI) and the two controlling objects, MedLib and MatchLib. Any of these three entities can reside at any location. In fact, the TeleMed GUI can communicate with any number of MedLib objects, which, in turn, can call upon the services of any number of MatchLib objects. Suppose, now, the user clicks on a CT study icon in the GPR in Figure 3. This causes a request to be sent to the current MedLib to retrieve that patient's CT study from the corresponding persistent object store. That transaction causes the user interface shown in Figure 5 to appear.

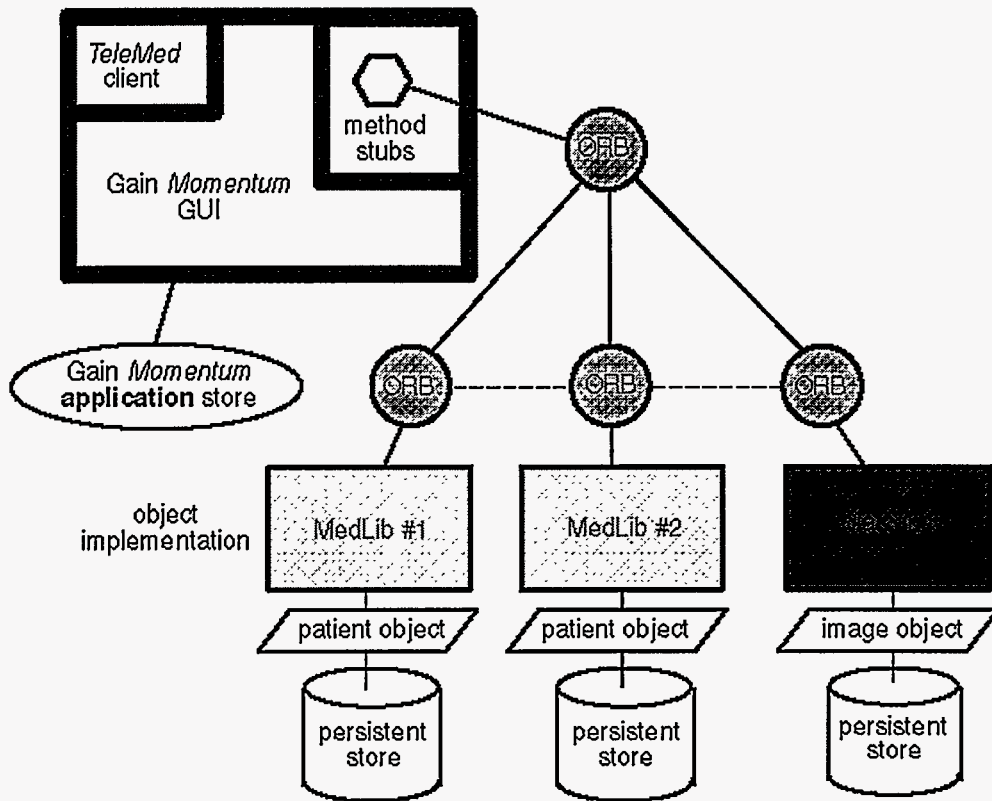


Figure 4: *TeleMed* Object Architecture

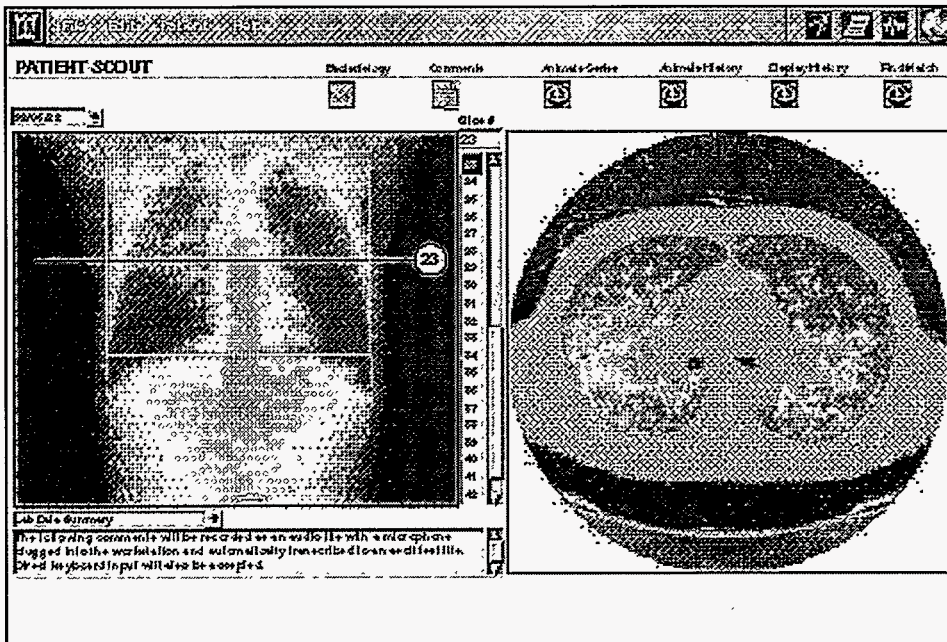


Figure 5 *TeleMed* CT Study Interface

The image on the left of [Figure 5](#) is a scout, so named because it was originally used by the CT technician as a guide in determining where to produce full transverse slice images of the patient. In this interface the

scout is similarly used, but now as a guide for the physician in selecting slices to view from the database. To do this, the horizontal cursor is dragged up or down to the desired location and released. Here, slice number 23 was selected and is shown on the right.

If you have a Java-enabled browser, you can get a idea of what the interface behaves like with a prototype Java applet accessible from <http://www.acl.lanl.gov/Java> which implements a portion of this interface. We anticipate that much of our future development will be with Java rather than with Gain Momentum in order to reduce distribution and maintenance costs and to ensure cross-platform capability at minimal cost. This development will be particularly enabled with Java ORB technology that is now appearing in the marketplace. With Java and ORBs we can maintain the same capability we deliver with Gain Momentum today including security, multimedia data handling and annotation capability resting on a full transaction processing distributed object database system.

Any object in the system can be annotated with sound or text so that the physician can note some important feature in an image and call the attention of a consulting physician in a natural manner. These annotations are stored in the object database with links to the actual data. In addition, the annotations can be viewed as a group so that individual annotations can be examined including seeing them in their full context. This is very important for referrals in which the referred to physician wants to examine the entire patient record without having to discuss details with the attending physician.

The system maintains full data integrity in the multiple databases so that any new information added by a team of physicians can be viewed by all the others in near real time.

Data Mining

We now describe one of TeleMed's most powerful features. This feature allows a user to perform a "query by example" search of an image database. Many technologies are represented in this feature - massively parallel computation servers, image analysis agents, and distributed object computing. To be specific, the MatchLib object shown in Figure 4 encapsulates the image analysis agent as a member function. For best performance MatchObj will typically reside on a massively parallel computer because the matching algorithm is inherently parallel. The signature database, which contains representative features of each image, usually resides on the same machine as MatchLib. Finally, the user invokes this entire matching operation simply by clicking the "Find Match" button in the upper right of Figure 5. The selected slice is used as the query image. The result of a matching operation is shown in Figure 6.

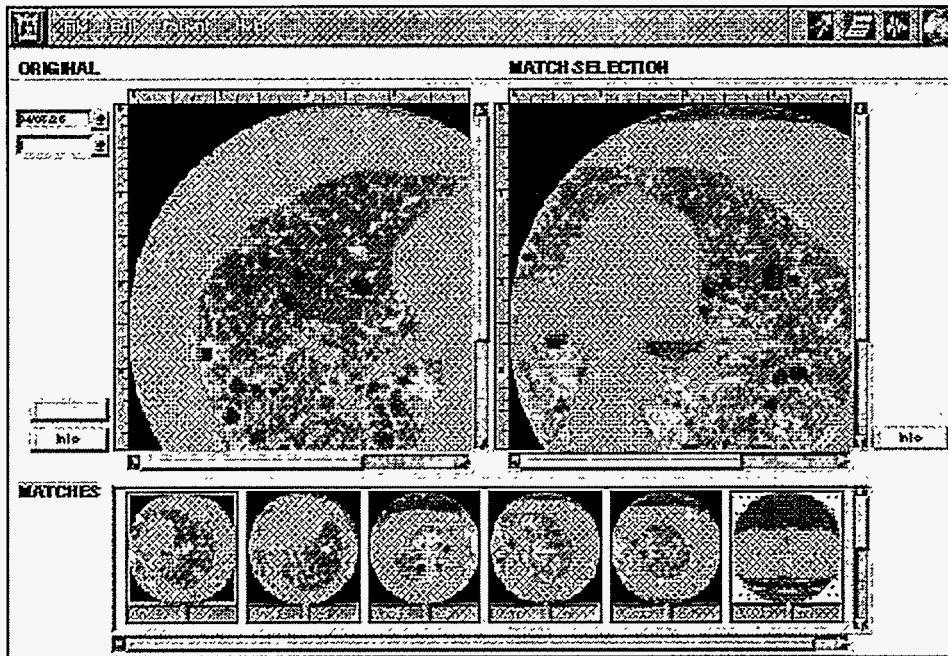


Figure 6 *TeleMed* Image Matching Interface

In Figure 6 the upper left image is the same one the user specified as the example query image. The result of the match is summarized by the thumbnails in the lower scrolling window. Clicking in a selected thumbnail causes its full-size representation to appear in the upper right comparative inspection window.

Security Infrastructure

Protecting patient confidentiality is the primary security concern for any medical data. Also important are protections against unauthorized additions, deletions and modifications of patient data. The security infrastructure layer of *TeleMed* is intended to provide the security services necessary to allow access control for patient data. This includes two fundamental services: an authentication framework and secure remote method calls. Any particular access control policy is implemented at a higher level, using an application-specific authorization object and access control lists.

Security is introduced in the system through filters and transformers in the method stubs in Figure 4. The basic interactions for secured objects are depicted in figure 7. Circles represent processes or objects, arrows represent communication, and dashed boxes represent address-space boundaries.

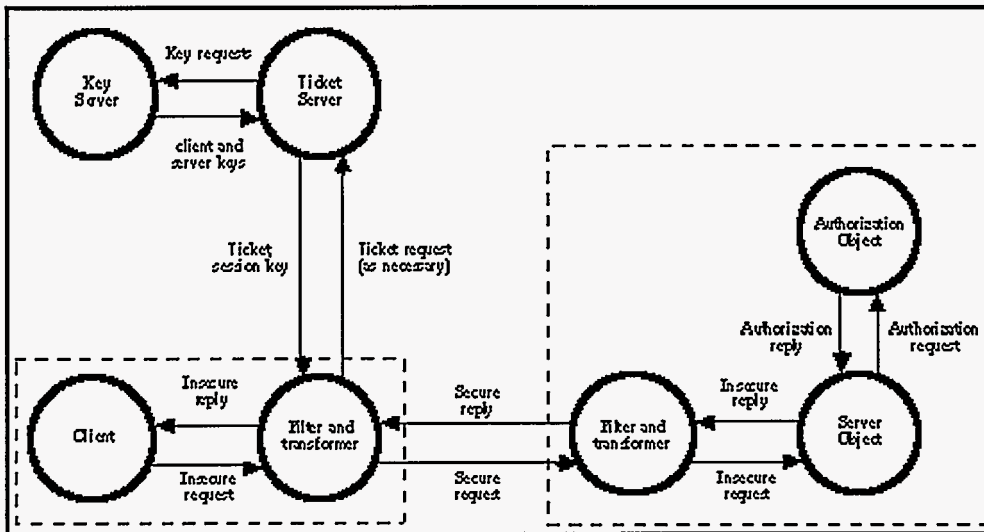


Figure 7: Secured object interactions

A key and ticket server pair provide fundamental authentication services for both human users and CORBA server objects, and also provide a mechanism for secure session key exchange. The authentication and key exchange protocol is similar to Kerberos. Users register RSA public/private key pairs with the key server, while DES secret keys are registered for CORBA objects.

Encrypted RSA private keys are stored in the key server to allow convenient remote access by users. By retrieving private keys via CORBA, users need not have their keys stored on a local file system. Private keys are encrypted with a DES secret key (constructed from a user-selected pass-phrase) before submission to the key server, so the private keys are never disclosed to a system administrator.

A secure interaction with a server object is initiated when a client makes a remote method call. The ORB³ marshals the call into an ORB request. An `outRequestPreMarshal` filter processes the request, by initializing several global variables with values identifying the client and providing a session key, an encrypted ticket and a forwarding token for the server. If necessary (such as the first time a client interacts with a particular server), the filter will request a session key and ticket from the ticket server.

When a ticket is requested for a particular client-server pair, the ticket server retrieves the public key for the client and the secret key for the server. The ticket server then generates a session key, and constructs the required ticket. The ticket consists of a digest of the client user name and the session key. The ticket server returns a copy of the session key encrypted with the client's public key and a copy of the ticket encrypted with the server's secret key, along with a signature to authenticate the ticket and session key. The filter authenticates the returned values, deciphers the session key using the client's private key, and stores both the session key and the encrypted ticket for future remote method calls.

After the filter completes its processing, the ORB passes the request (as an array of unsigned char) to a client-side request transformer. The transformer encrypts the request, adds the client's user name, encrypted ticket and forwarding token to the request, and passes the data back to the ORB, which forwards the request to the remote server.

When the server-side transformer receives the encrypted request, it extracts the client's user name, the encrypted ticket and the forwarding token from the request. It then uses the server's secret key to decipher the encrypted ticket and verifies the identity of the client. Finally, it uses the session key from the ticket to decipher the encrypted remainder of the request, and passes the deciphered request back to Orbix. The transformer stores the session key for use in encrypting any method reply.

The ORB uses the deciphered request to dispatch the appropriate method call to the local server object. The server object processes the method call, consulting a local authorization object to ensure that the requested access is authorized. When the method call returns, the ORB passes the reply request to an `outReplyPreMarshal` filter. That filter determines whether the method is returning non-sensitive image data,

and sets a global variable indicating whether or not the transformer should encrypt the reply. For performance reasons, image data that does not contain identifying information is returned unencrypted.

After the filter completes its processing, the ORB passes the request to the server-side transformer. The transformer encrypts the data, depending upon the value of the variable set by the `outReplyPreMarshal` filter. It also adds the value of the variable to the request (so that the client can determine whether or not the incoming reply is encrypted), and passes the data back to the ORB.

When the reply returns to the client side, the ORB passes the request to the client-side transformer. That transformer checks whether the reply is encrypted, and deciphers it, if necessary. It then passes the deciphered request back to the ORB, which returns the reply normally.

Like Kerberos, the authentication mechanism is implicit. Users authenticate their identity by virtue of their ability to successfully decipher the encrypted session key. Servers authenticate their identity by virtue of their ability to successfully decipher the encrypted ticket. The ticket server authenticates its identity by signing the session key and ticket it produces (using an RSA private key).

It is worth noting that the client of Figure 7 may be a CORBA server acting on behalf of another client of its own. For example, in the *TeleMed* system, when a user requests a merged patient record, a *MedLib* server must contact other *MedLib* servers to gather references to the distributed patient data. In such a case, the object must be able to establish a connection with exactly the privileges appropriate to its own client. That is, in some cases, a server object must be able to forward the credentials of its client to other servers.

To support credentials forwarding, the ticket server provides a forwarding token whenever a session is initiated. A particular transformer may or may not forward that token to its server, depending upon whether or not credential forwarding is to be allowed. The forwarding token may be used by the server to request a new ticket (for another server) on the client's behalf. A ticket retrieved in such a manner identifies both the effective user (the original client) and all intermediate users (server objects), so that each server can be programmed to limit or even prohibit the use of credential forwarding by its clients.

We plan on evolving this design to be in compliance with the CORBA security model that has been recently adopted by the OMG.

Summary

In the *TeleMed* application, we have developed a powerful environment for navigating through patient data both by browsing and by providing specialized search tools to show relationships between data that are not always obvious to the physician or the end user. This enables the presentation of complex information in a controlled manner so that one can obtain the information of importance without information overload. The paradigm used here is very general, only requiring small modifications to be used on a wide range of disciplines. It is extensible in at least two ways. First, the ability to display virtual visual "metadata" whose spatial relationships conveys higher level relationships is a general capability with applicability far beyond medicine but applying broadly in science and engineering applications and beyond. Secondly, the idea of a "patient record" is directly applicable in engineering systems where one tracks a complex system over its entire life history from design and assembly to testing and evaluation at a later data after aging or potential damage. We see an ever increasing need for proper organization and archiving of important information related to experimental as well as computational systems. *TeleMed* is the beginning of a new type of application in this area of information handling. More documentation on *TeleMed*, including the detailed information on the Object Database Adapter technology we have implemented is available at <http://www.acl.lanl.gov/sunrise/Medical/overview.html>.

Reference

¹ David Forslund and David Kilman, "The Virtual Patient Record: A Key to Distributed Healthcare and Telemedicine", <http://www.acl.lanl.gov/telemedicine/virtual.html>.

² Alan Dickman, "Two-Tier Versus Three-Tier", *Information Week*, Nov 13, 1995.

³ The ORB we have been using is Orbix from Iona Technologies, see <http://www.iona.ie>.

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.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.