

Instrumentation and Controls Division

APPLICATION OF TELEPRESENCE TECHNOLOGIES
TO NUCLEAR MATERIAL SAFEGUARDS

by

M. C. Wright and J. A. Rome

Oak Ridge National Laboratory*
P.O. Box 2008
Oak Ridge, TN 37831-6004
423-574-5604

Paper to be presented at the
Sixth International Conference on Facility Operations – Safeguards Interface
Jackson Hole, Wyoming
September 20-24, 1999

*Research sponsored by the U.S. Department of Energy and performed at Oak Ridge National Laboratory, managed by Lockheed Martin Energy Research Corp., for the U.S. Department of Energy under contract DE-AC05-96OR22464.

APPLICATION OF TELEPRESENCE TECHNOLOGIES TO NUCLEAR MATERIAL SAFEGUARDS

Michael C. Wright
Oak Ridge National Laboratory
PO Box 2008; MS 6004
Oak Ridge, Tennessee 37831
(423)574-5604

James Rome
Oak Ridge National Laboratory
PO Box 2008; MS 6298
Oak Ridge, Tennessee 37831
(423)574-1306

ABSTRACT

Implementation of remote monitoring systems has become a priority area for the International Atomic Energy Agency and other international inspection regimes. For the past three years, DOE2000 has been the United States Department of Energy's (DOE's) initiative to develop innovative applications to exploit the capabilities of broadband networks and media integration. The aim is to enhance scientific collaboration by merging computing and communications technologies. These Internet-based telepresence technologies could be easily extended to provide remote monitoring and control for confidence building and transparency systems at nuclear facilities around the world. One of the original DOE2000 projects, the Materials Microcharacterization Collaboratory is an interactive virtual laboratory, linking seven DOE user facilities located across the US. At these facilities, external collaborators have access to scientists, data, and instrumentation, all of which are available to varying degrees using the Internet. Remote operation of the instruments varies between passive (observational) to active (direct control), in many cases requiring no software at the remote site beyond a Web browser. Live video streams are continuously available on the Web so that participants can see what is happening at a particular location. An X.509 certificate system provides strong authentication. The hardware and software are commercially available and are easily adaptable to safeguards applications.

I. INTRODUCTION

For the past three years, the DOE2000¹ program has been the U.S. Department of Energy's program to change fundamentally the way scientists work together and how they address the major challenges of scientific

computation. The DOE2000 program has three main goals: improved ability to solve DOE's complex scientific problems, increased R&D productivity and efficiency, and enhanced access to DOE resources by R&D partners. One of the strategies to meet these goals is the construction of national collaboratories. Collaboratories provide integration on the Internet of unique or expensive DOE research facilities together with the expertise for remote collaboration, experimentation, sample preparation, software development, modeling, and measurement. In addition, collaboratories benefit researchers by providing tools for video conferencing, shared data viewing, and collaborative analysis.

The DOE2000 research program consists of seven technology R&D projects and two cooperative pilot projects, jointly funded by DOE2000 and a scientific program area. While the projects are all independent entities, they are expected to interact to produce or specify tools of general interest. The cooperative pilot projects have been expected to demonstrate the utility of collaboratories in the scientific environment and to serve to motivate other DOE communities to become involved. The goal of the Diesel Combustion Collaboratory² (DCC) is to improve on the already successful collaborative effort of the Diesel Combustion Research CRADA partners by implementing, evaluating, and testing a set of collaborative tools. Participants are SNL, LBNL, LLNL, LANL, and the University of Wisconsin. Industrial partners are Cummins Engine Co., Caterpillar Inc. and Detroit Diesel.

The Materials Microcharacterization Collaboratory (MMC) pilot project^{3,4} unites the five DOE-funded electron beam microcharacterization facilities located at ANL, LBNL, ORNL(2) and the University of Illinois at Urbana-Champaign. Also included in the MMC project

is a microcharacterization facility at the National Institute of Standards and Technology. To ensure that technology benefits can be applied to venues other than electron-beam microcharacterization, the MMC also includes neutron and x-ray beam lines at ORNL and BNL. Current industrial partners are Gatan Inc., R. J. Lee Group, EMiSPEC Systems Inc., Philips Electronic Instruments, Hitachi Instruments, Inc., Japan Electron Optics Laboratories-USA, SUN Microsystems, and Graham Technology Solutions.

The two pilot projects are complementary. The DCC is focusing on tools for shared analysis and archiving of data from experiments and models among a collection of institutions that have been working together for many years. In contrast, the MMC project is focusing on tools to make its experimental facilities and expertise available to its own members but mostly to the national materials science research community.

The MMC is interested in real-time interactive control of complex remote instrumentation and in real-time interaction among collaboratory participants. Remote monitoring instrumentation used in the nuclear materials safeguards community has generally operated in a much less interactive mode. Remote autonomous instruments collect data over some time interval, store the data locally, then transmit the accumulated data at a later time. Expertise gained through the DOE2000 program will provide the materials safeguards community with additional options for remote monitoring.

II. REMOTE INSTRUMENT OPERATION

A. Three Approaches

The MMC partners, before the formation of the Collaboratory, were performing a significant amount of research and development into remote control of scientific instruments. We are continuing to refine our independent approaches while migrating to a common one. Tools for interactive instrument control fall into three broad categories: Web-based, client-server, and remote computer control. At ANL work has focused on Web-based instrument control using CGI and perl scripts. LBNL has focused on a distributed computing middleware model using Java, C++, and CORBA. At ORNL most remote control has been done using commercial remote computer control software, specifically Timbuktu Pro. Each of the three approaches has advantages for different applications. Web-based tools show great promise for wide distribution of basic instrument operation, but not all instrument functionality can be easily provided in the confines of a browser. CORBA-based distributed computing is efficient and

scalable but requires a complete redesign of an instrument's data acquisition system if it is applied to an existing instrument. Commercial remote control software can easily control an existing instrument's computer rather than the instrument itself, but remote control software uses proprietary protocols, and there are concerns about security and network efficiency.

B. Web-based Remote Operation

The World Wide Web is experiencing explosive growth in both its distribution and its capabilities. While originally used for the display of multimedia information, the Web has now become a mechanism for interactive applications. The advantages of a web-based solution are numerous. Web communication is platform independent, browsers are ubiquitous and everyone knows how to use them, web traffic is usually allowed to pass through firewalls, and strong security can be implemented. A number of web-based solutions have been successfully implemented within the MMC. Figure 1 shows one of these, remote control of a state-of-the-art scanning electron microscope. Only a web browser is required at the remote site. At the local site, the web server communicates through CGI scripts to a "telepresence" server that handles communication with instrument, user authentication, session control, and security.

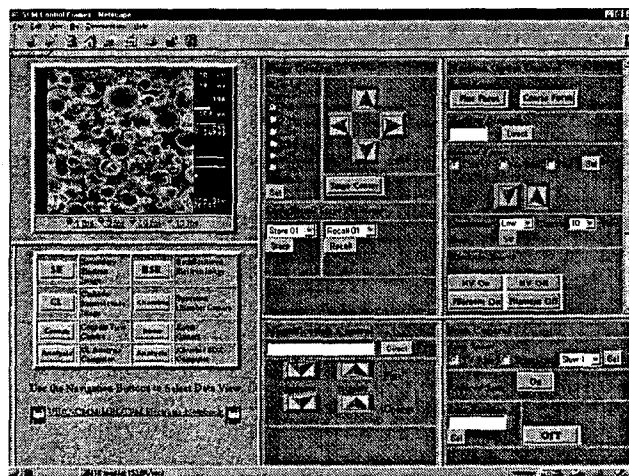


Figure 1. Screen shot showing web-based control of a scanning electron microscope.

C. Client-Server Remote Operation

When writing a new data acquisition system, or when sufficient access is provided to an existing one, a client-server approach has many advantages. It can be a very efficient user of network bandwidth and strong security can be readily implemented.

We have begun development of a collaborative framework for distributed instrument control⁵. Requirements are specified in terms of functionality, scalability, interactivity, and safety and security. To meet these requirements, we have introduced three types of services in the architecture: Instrument Services (IS), Exchange Services (ES), and Computational Services (CS). These services may reside on any host in a distributed system. The IS provide an abstraction for manipulating different types of instruments; the ES provide common services that are required between different resources; and the CS provide analytical capabilities for data analysis and simulation. These services are brought together through CORBA and its enabling services, e.g., Notification Services, Time Services, Naming Services, and Security Services.

The instrument server sends information to the client describing the operational parameters of the instrument. These are set, get, can do, and metadata about things such as the range and step size of each setting. The client dynamically generates a graphical user interface to allow the remote user to control the instrument. The server runs under UNIX, while the client is written in Java to allow operation on any platform.

D. Computer Remote Control

Essentially all modern scientific instrumentation is controlled by a computer and increasingly, these computers are commodity PCs running a version of Windows. In many cases, the controlling software for the instrument is "closed" with no documented ability to externally script or to control the application. Remote control software bypasses this limitation by controlling the remote computer rather than the remote instrument. The screen, keyboard, and mouse of the local computer are duplicated at the remote site. This software is commercially available at very low cost. Most of our work has been done using Timbuktu Pro⁶ because it supports Windows 95/98, Windows NT, as well as Macintoshes. A computer running any of these operating systems can control a computer running that same operating system or any of the others as well.

Performance on high speed (>10 Mb/s) local networks is excellent with only a slight sluggishness in the perceived responsiveness of the computer. Performance on a wide area network will, of course, depend on the network performance but is acceptable even at ISDN speeds. One reason that performance is better than might be expected is that Timbuktu Pro does not simply send bitmap images of the local screen; when possible, it sends the system commands that cause elements of the screen to be redrawn. These commands can require much less bandwidth to transmit than the

resulting bitmaps. Another commercial program, pcAnywhere³², provides capabilities similar to Timbuktu, for Windows only, but has the added feature of a Java applet version of its client⁷. The remote user can simply connect to the local computer via a Web browser and the required client is downloaded automatically.

A limitation of these products, for some of our purposes, is that they are proprietary and do not support Unix. We would like, for example, to be able to be able to use alternative compression schemes for screen images or to be able to encrypt the communication between the clients and servers. Electron microscope images are noisy and do not compress well using traditional techniques. Hooks for these changes are available with VNC from AT&T Laboratories Cambridge^{8,9}. VNC is distributed under the terms of the GNU General Public License, meaning the program and source code are freely distributable. Servers exist for X Windows, Win 32, and Macintosh (beta). A Java viewer is available, which will run in any Java-capable browser. VNC can be used with SSH to provide substantially increased security. VNC is slower than the PC-only products because it does not take advantage of the system screen-drawing commands mentioned above.

III. VIDEO EVERYWHERE

Tools for interactive audio/video communication fall into three broad categories: one-to-one, one-to-many, and many-to-many. One-to-many Web broadcasting (ex, RealVideo) is easier than the others since several seconds of data buffering can be introduced to smooth over momentary lapses in network performance. One-to-one interaction via commercial desktop video conferencing tools can be done today and the advent of the H.323 standard has enabled much more interoperability between these programs. Multiway conferencing is the most difficult to achieve. The nonverbal interaction, for example, to determine who speaks next in a group situation requires high fidelity and low latency connections.

Video conferencing is important for collaboration. Reasonably high quality audio/video conferencing has been available for some time with the use of hardware compression and dialup ISDN lines (ex, PictureTel). However, we are interested in software-only Internet-based solutions so that they can be widely deployed at low cost. Frame grabbers and video cameras are now available for as little as one hundred dollars and in some cases are already built into the computer. Modern personal computers now have the computing power to compress and decompress video streams on the fly. The leading applications for one-to-one video conferencing

are CU-SeeMe¹⁰ and Microsoft NetMeeting. New programs with similar capabilities are entering the market but most only support Wintel PCs. Image quality varies with a number of factors but is typically on the order of 160×120 pixels at 5 frames per second. Even this low resolution is useful in many applications.

Multiway video conferencing usually requires a reflector such as White Pine's MeetingPoint¹⁰. The reflector accepts audio/video feeds from each participant and sends them out to all the others. Multiple reflectors can be connected so that, for example, if three participants at each of two sites are communicating, only one copy of each stream is transmitted over the wide area network. Each reflector replicates the stream for the participants at its site. A newer tool, very similar to CU-SeeMe, called iVisit¹¹ allows direct peer to peer connections without the need for a reflector. However, this means that each participant must transmit his video stream to every other participant rather than a single stream to the reflector.

An alternative to a reflector is the use of IP multicast using the Mbone tools (vic & vat). These tools work well on Unix, marginally on Windows, and are essentially nonexistent on the Macintosh. While in principle more efficient than a reflector in use of the network, IP multicast is not supported by all routers. IP multicast is most useful when a large number of people need to observe the same broadcast.

Perhaps surprisingly, network video works much better than network audio since dropped frames on video are annoying but gaps in the audio stream are intolerable. In addition, network latencies or buffering usually cause a few seconds delay in the audio making interactive conversation awkward at best. For now, in the MMC we are using telephone conference calls combined with video via CU-SeeMe for our weekly project meetings. Commercial market forces should lead to advances in desktop video conferencing in the reasonably near future. Internet video conferencing that is truly as transparent as a telephone call will probably have to wait until quality of service features are implemented on the Internet.

For one-to-many transmissions, we have made extensive use of streaming JPEG video in a web browser. For most applications, we have found this technique to be more useful for daily interaction than traditional video conferencing. These video streams perform remarkably well and require no special software on the receiving end. Using drivers from Graham Technology Solutions¹² running on Sun workstation we can transmit 320×240 pixel video at 10 frames per second to a Netscape browser running on a desktop computer. Using frames,

multiple video images can be display on a single web page.

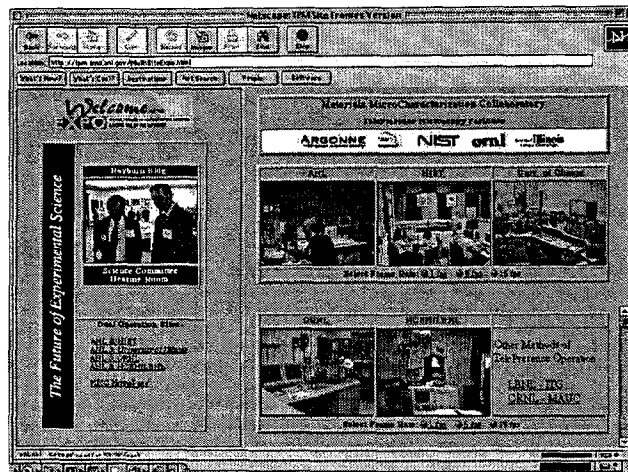


Figure 2. Screen shot showing six live streaming JPEG video streams on a single web page.

Figure 2 shows an example screen shot from a recent MMC demonstration. The five video images on the right show live feeds from the five main MMC microscopy sites around the country. The image on the left is a live image from the site of the demonstration itself. That image was generated using a program called Webcast¹³, running on a PC laptop. Webcast broadcast clients send their video streams to a special web server that makes them available on the web. The server can accept up to 26 streams and broadcast them to up to 256 recipients. Webcast is hard coded to limit the load on the network so that images are limited to about 200×150 pixels at 5 frames per second.

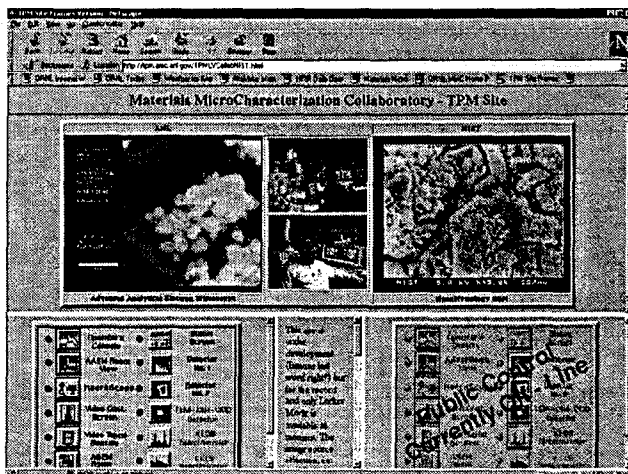


Figure 3. Screen shot showing the use of streaming JPEG images as a collaboration tool.

These video feeds can also serve as a powerful collaboration tool. Figure 3 shows an example where two microscopists can work together to examine the same type of sample using electron microscopes with different capabilities. The smaller images in the center show the operators of the microscopes and the larger images show the live images coming from the microscopes themselves. While audio can also be transmitted to these web pages, we have found it much more satisfactory to simply use the telephone.

Another example of the usefulness of streaming, in Figure 4, shows how these images can be used to create a sense of community among the collaborative participants. The page shows static images of three microscope labs and three staff offices. Any or all of the images can be converted into live video by simply clicking on the image. This is the online equivalent of walking down the hall and entering an office or a lab to see what is happening there. The occupants of the rooms know if anyone is watching and also can see the IP address of the viewers. In the MMC project we have about 50 video feeds available at all times in labs and offices around the country. The desire is to simulate being "down the hall while around the world". A sub-thousand dollar PC, a \$100 camera, the Webcast client, and a network connection are all that are required to place a location live on the Internet.



Figure 4. Screen shot showing the use of streaming JPEG images to create a "virtual laboratory".

In cases where more resolution is desirable but frame rate is not an issue other products can be used to transmit high-resolution still images over the web. A freeware utility, pjWebCam, runs as a simple web server that will transmit static images each time it is accessed. This image can be transmitted at the full resolution of the video camera, generally 640×480 pixels. The program

can also be configured to grab pictures at a specified interval and upload them to an FTP site.

We like the video driver software provided with the Winnov Videum¹⁴ cards. These low-cost cards have inputs for composite video, S-video and a Winnov digital video camera. The card can switch between the inputs so that three camera can be simultaneously connected without the need for an external switcher. The hardware and driver software can simultaneously provide video to more than one Windows application. For example, a camera can be used for a live video image and, at the same time, that camera or one of the other connected cameras can be used to send higher resolution still images on demand. A utility is provided that allows the settings of the hardware and driver (video source, brightness, contrast,...) to be controlled by a remote computer over the network. In addition a remote electronic pan, tilt, and zoom capability is provided. Since the images transmitted over the network generally are at significantly less resolution than is available from the camera, this remote steering is quite useful.

IV. SECURITY ARCHITECTURE

Security is vital on the web. The MMC is putting valuable and complicated facilities online to the whole world. It is a challenge to implement a security architecture that works for all of our instruments and remote control approaches. The most mature and broadest security technology is based on the use of x.509 v3 certificates that link an identity to a public key. These certificates implement the public key infrastructure (PKI) that can be used for many different purposes:

- Secure authentication of clients and servers
- Encryption of Web traffic (SSL)
- Secure e-mail with digital signatures and encryption (S/MIME)
- Object signing of Java programs and Active-X controls
- Authorization certificates

Securing Web traffic is the easiest and most cost-effective way to introduce secure remote access. To implement the secure sockets layer (SSL) encryption requires a Web server that supports it (Netscape, IIS, Apache Stronghold) and a certificate for the server.

A. Certificate authorities

In principle you can purchase server certificates from well-known public certificate authorities (CAs) such as Verisign, Thwate, or RSA. However, they provide public guarantees as to the identity of the certificate holder that may not be needed for a

collaboratory. You may also not be listed in Dunn and Bradstreet or be willing to pay \$350 per server. So, It is best to issue your own certificates. It is possible to "roll your own" certificates using the SSLEAHY toolkit, but we do not recommend it. To make it easy to administer the certificates, a user-friendly graphical user interface and a significant infrastructure are required to implement certificates. It is well worth the cost of a certificate management system from a vendor such as Netscape. The Netscape system provides a secure Web-based access for users and administrators to apply for, issue, and to retrieve certificates. Pricing for these products is usually "by certificate," and costs run from \$7 to \$50 per certificate.

With your own CA, you can issue certificates that can be used for secure e-mail and object signing. In addition, you can issue short-term certificates that expire in a few weeks to allow temporary access without having to create and maintain certificate revocation lists (CRLs).

B. Scalability

Scalability is a big issue for the public key infrastructure (PKI). The original vision of a global CA hierarchy, with one root CA issuing a few CA certificates, and each of these CAs issuing some more, and so forth failed to materialize. This failure was due to the lack of trust across domains. People realized that it was not possible to really identify a person without some out-of-band method à la PGP (pretty good privacy). It is hard to know whether a given John Smith is the one you went to high school with unless you call on the telephone and ask a few questions. Fortunately, collaboratories are at just the right size where trust is not a big issue. They are large enough to be able to afford the costs of proper security implementation, and they are small enough so that usually someone knows someone who knows every person in the collaboratory. And it is unlikely that scientists become criminals overnight so that CRLs are not needed.

C. Secure servers

Although Microsoft's Internet Information Server (IIS) is free, we cannot recommend it for high-security applications. It has had numerous security problems, and even worse has a basic design flaw. IIS shares the same security context as Internet Explorer (IE). Secure Web servers must accept the CA that issues their server certificate as "trusted." This is done using IE. The problem with this approach is that you probably want to set up your server to require client certificates and to accept only those that are issued by your CA. If you do not "untrust" all other CAs, anyone who presents a valid certificate from the other CA will be allowed into your

system. However, once you perform this operation, it is impossible to verify the authenticity of downloaded security patches for IIS, because you cannot verify their signatures.

D. Client certificates

Your CA can issue client certificates. We have found it convenient to embed some role-based access information into these certificates as shown in Figure 5. The MMC has four classes of user:

- Guest — can observe
- Researcher — can operate most features of the equipment remotely and access the data
- Operator — can access all equipment features and all data
- Administrator — can change security implementation and manages users

When a user presents his certificate to an MMC facility, we can quickly change the Web pages to only present the allowed features and access points.

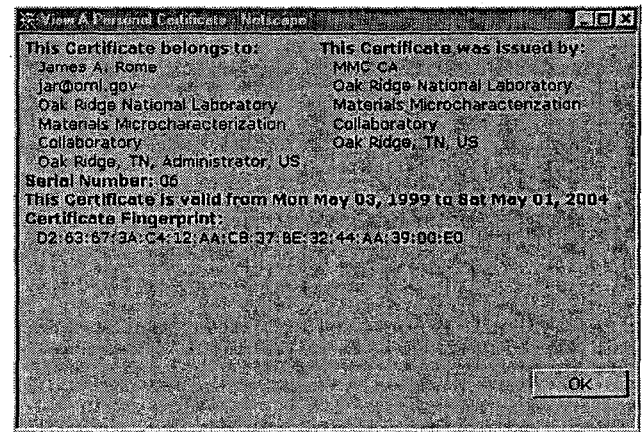


Figure 5. MMC client certificate showing the role of the user (Administrator).

An important issue is whether the user's security context (his certificate) can be moved from computer to computer and outside the Web browser. Both Netscape and IE support the export of the user's certificate and his private key into a password-protected PKCS#12-formatted file. Therefore, a user can obtain his certificate from Netscape, export it, and import it into IE or Netscape on the same machine or on other machines. However, although PKI is supported in both Java 1.2 and in CORBA, neither of them imports PKCS#12 files. This is a big problem because it means that it is difficult to use one security context (i.e., the same certificate) for a user in standalone applications. As a result, we will probably implement the security controls using a Web-based interface that can launch a Java applet.

E. Strong authorization

The standard Unix-like file access permissions no longer suffice to implement a realistic security policy. Permissions may be a function of time, require certain user bona fides, or have to implement stakeholder rights. The latter is an important issue for government work. We all must obey certain government rules and regulations, and sometimes we must prove that these are actually being enforced. For example, for medical experiments, an agency might have to certify that a proper protocol was filed and was approved by the agency before access is allowed. The user might have to prove that he has passed x-ray safety training, computer security training, and so forth.

As part of the DOE2000 program, the Lawrence Berkeley National Laboratory has developed the Akenti¹⁵ system to implement these concepts in cyberspace. Stakeholders independently make assertions about resource use. Trusted third parties certify user attributes required for the use conditions, and authenticated users that possess the required attributes easily gain access. All of these operations are performed by implementing extensions in the x.509 protocol. A different approach to authorization certificates is being developed by the SPKI (simple public key infrastructure) IETF working group¹⁶. The tenet of SPKI is that ultimately, only the owner of a resource can grant access to that resource, and issues a certificate to grant this privilege.

V. SUMMARY

The MMC is interested in remote scientific collaboration and not just remote instrument control. While centered around electron microscopy, the MMC includes neutron and X-ray beamlines and other microcharacterization instrumentation. They are included to ensure that the tools and techniques we develop will not be useful only for microscopy but for the general scientific community. We are continuing to refine our independent approaches while migrating to a common Web-centric approach. We are now performing materials science research with Internet-based remote experiment control and experimenter interaction.

We believe that these telepresence technologies can be readily adapted to provide the nuclear materials safeguards community with valuable new options for remote monitoring of facility operations. In addition, these technologies can improve personal interactions within this global community. The rapidly improving performance, reliability and security of the Internet will lead to wide spread use of telepresence technologies in the near future.

ACKNOWLEDGEMENTS

The MMC project is a true collaboration of many people. The authors want to recognize the important contributions of the following individuals: Edgar Voelkl, Larry Allard, Ted Nolan, Ed Kenik and Cam Hubbard of Oak Ridge National Laboratory; Nestor Zaluzec of Argonne National Laboratory; Michael O'Keefe, Bahram Parvin, and John Taylor of Lawrence Berkeley National Laboratory; Jim Mabon of the University of Illinois, Urbana-Champaign; and Michael Postek of the National Institute of Standards and Technology.

Funding for the MMC is provided by the U.S. Department of Energy, U.S. Department of Commerce, and the industrial partners mentioned above. ORNL is managed by Lockheed Martin Energy Research under contract number DE-AC05-96OR22464.

REFERENCES

1. DOE2000 Program URL = <http://www.mcs.anl.gov/DOE2000>.
2. The Diesel Combustion Collaboratory URL = <http://www-collab.ca.sandia.gov/Diesel/ui/>.
3. The Materials Microcharacterization Collaboratory URL = <http://tpm.amc.anl.gov/MMC>.
4. James Rome, "Science From a Distance", *The World & I*, October 1998, pp 182-189.
5. B. Parvin, J. Taylor, G. Cong, "DeepView: A Collaborative Framework for Distributed Microscopy", IEEE Conference on High Performance Computing and Networking, Orlando, FL, Nov 9-13, 1998.
6. Timbuktu Pro URL = <http://www.netopia.com/software/tb2>.
7. pcAnywhere32 URL = <http://www.symantec.com/pcanywhere>.
8. Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood & Andy Hopper, "Virtual Network Computing", *IEEE Internet Computing*, 2, 1, 33-38 (1998).
9. VNC URL = <http://www.uk.research.att.com/vnc/>.
10. CU-SeeMe and MeetingPoint URL = <http://www.wpine.com>.
11. iVisit URL = <http://www.ivisit.com>.
12. Graham Technology Solutions URL = <http://www.graham.com>.
13. Webcast URL = <http://www.gcomm.com>.
14. Winnov URL = <http://www.winnov.com>.
15. Akenti URL = <http://www-itg.lbl.gov/security/Akenti>.
16. SPKI URL = <http://www.clark.net/pub/cme/html/spki.html>.