LBNL-40021 Rev.

The Reality of Collaboratories

D. A. Agarwal, S. R. Sachs, and W. E. Johnston [1]

Ernest Orlando Lawrence Berkeley National Laboratory
Berkeley, CA 94720 USA

Abstract

Advances in data acquisition and dissemination technologies, together with several other computer science technologies, provide an opportunity to define new environments for remote access to scientific instruments and collaboration between researchers at remote sites. These environments, called "collaboratories"[1], are intended to provide complete location-independent collaborative access to instruments, data acquisition and analysis resources, as well as to collaborating researchers.

Over the past two years we have built a prototype collaboratory at the Spectro-Microscopy Facility of the Advanced Light Source (ALS) Beamline 7.0 at Lawrence Berkeley National Laboratory. The scientists who use the Beamline are based at different cities around the world. Before the introduction of the collaboratory, experiments at the Beamline required that multiple researchers travel to Berkeley, CA. With the collaboratory, scientists can now discuss experiments with other scientists around the world and at the Beamline, "walk" around the Beamline using remotely controlled cameras, access data as it is taken, monitor instrument parameters, and do limited control of the experiment from their offices.

This paper presents the design of the prototype Spectro-Microscopy Collaboratory, and the lessons learned in providing remote access to the facility. It also presents the software architecture and components that are being developed to provide a toolkit for bringing future experimental equipment on-line.

Keywords: collaboratory; remote access; spectro-microscopy; telepresence; electronic notebooks; experiments on-line

[1] This work is supported by the Director, Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences Divi sion, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098 with the University of California.

Submitted to Computer Physics Communications, 31 October 1997

1 Introduction

The U. S. Department of Energy has many user facilities which are used by researchers all over the world; one such facility is the Advanced Light Source (ALS), one of the brightest sources of soft X-ray beams. Beamline 7.0 was the first operational user beamline, the Spectro-Microscopy Facility. This facility is based on an electron undulator light source coupled to a high resolution spherical grating monochromator. Both the undulator and monochromator are state-of-the-art machines. This undulator Beamline provides the soft X-ray photons for a suite of experimental analytical equipment that makes up the Spectro-Microscopy Facility. These instruments make particular use of the high brightness of the undulator Beamline.

The principal users of the facility have their home institutions about 2200 miles from it. The collaboratory testbed is providing them with remote access to the ALS, allowing experiment control and monitoring from their offices. Internet Protocol (IP) network-based facilities allow researchers at different locations to collaborate on experiments as if they were together in the same laboratory. The expected value of these geographically distributed environments includes substantially increased effectiveness in doing science, and an enabling capability for analytical and high-value production use by industry.

The Distributed, Collaboratory Experiment Environments (DCEE) Program sponsored by the U. S. Department of Energy (DOE) consisted of four projects to build prototype remote experiment and collaborative environments. These prototypes are providing remote access to expensive, hard-to-duplicate, DOE scientific facilities, including electron microscopes, high resolution NMR and spectro-microscopy instruments, and a tokamak fusion reactor.

The DOE 2000 Program is providing the next stage in the development of collaboratory capabilities and consists of both research and development and pilot projects. These projects are providing essential collaboratory components identified by the original DCEE program. The pilot projects are implementing collaboratory technology in the areas of diesel combustion research and materials micro-characterization.

Other collaboratory development efforts include: a research database called the Worm Community System [12], and efforts to build remote access to instruments and collaboration capabilities in the electron microscopy community [11,14] and upper atmospheric research community [3] to name just a few. An excellent discussion of collaboratories in general may be found in [7].

The work described here represents some of the research and development of the mechanisms and infrastructure required to make collaboratories a reality. Some of these mechanisms have already been developed. Several of the

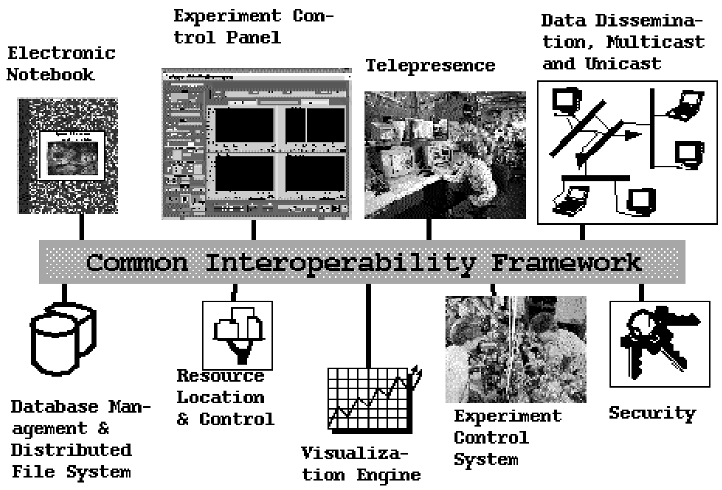

Fig. 1. The components of the collaboratory environment.

mechanisms need further research and development. The discussions here will focus on the Spectro-Microscopy Collaboratory DCEE project which we are continuing to develop and enhance.

2 Discussion

The traditional tools utilized by experimental science researchers include paper notebooks for recording experiment data, turn-key experiment control systems, and security accomplished through isolation, obscurity, or lock and key. In building a collaboratory environment it is not enough to simply provide access to the instruments, access to the scientific environment must also be provided. There are many software components required to make collaboratories a reality. Figure 1 illustrates these components. The tools required to build a collaboratory include electronic notebooks, on-line instruments, telepresence facilities, access control mechanisms, data dissemination mechanisms, resource location and coordination facilities, a unifying user interface and distributed file systems.

The work of making collaboratories a reality in scientific environments includes the research and development of the mechanisms and infrastructure that support them. In this paper we will describe the current state of each component, our experiences, and current development efforts.

2.1 Telepresence capabilities

The first step in building the collaboratory was to provide remote scientists a way to talk to each other and to scientists at the Beamline. We initially provided Internet multicast-based videoconferencing[9]. These tools provide workstation-based videoconferencing capabilities with a camera, microphone, and speakers at each site.

However, a remote researcher needs the ability to "walk" around the experiment site and interact with people. We will refer to this capability as telepresence. Telepresence facilities include videoconferencing capabilities, remotely controllable pan/tilt/zoom/focus cameras, echo-cancellation hardware for audio, and tools for rendezvous and launch of sessions. The multicast-based videoconferencing tools allow scalability to hundreds of simultaneous participants. The remotely controlled cameras are mounted in the ceiling and are attached to a video switch that is also remotely controlled. The echo-cancellation hardware allows hands-free natural conversations between the participants.

To avoid the need for sites that are transmitting audio and/or video to administer their network media feeds directly, or even to transmit continuously, a conference control tool has been developed to allow remote lauch and control of the videoconferencing sessions. Remote researchers are now able to keep in touch with each other and keep track of what is going on at the Beamline. When no remote researchers need the session, the camera feeds can be turned down or off. The conference controller can also be used as a training tool since it allows a remote site to determine what videoconferencing tools are running at a site, as well as the current settings.

The telepresence capabilities have provided researchers an even greater sense of connection to the project, and to the other participants than was originally expected. The conference controller provides complete local control over what actions can be taken by the remote users and over the list of authorized remote users. The combination of the conference controller and cameras that are mounted out of view has reduced the sociological concerns to negligible levels. This allows us to maintain continuous availability of the telepresence components since the users no longer disable the system as they did with workstation-based cameras.

2.2 Data dissemination

A critical element needed so that interaction between and within the collaboratory components easy to build is a common data dissemination layer. Each collaboratory component may use multiple communication mechanisms:

unreliable or reliable, unicast or multicast. For example, the experiment control and monitoring component in the Spectro-Microscopy Collaboratory uses three different communication channels. One is a reliable unicast connection from the site in control of the experiment. Another is the unreliable multicast of experiment parameter read-outs. The last channel is the reliable multicast of experiment data and control coordination data. The videoconferencing programs use unreliable multicast.

Current operating systems provide IP based unreliable and reliable unicast and unreliable multicast through the sockets interface. Although there are several reliable multicast research projects, there are only a few implementations. The interfaces to each communication protocol, and the services provided by them, differ significantly and require a relatively high level of expertise for their correct use. One of our goals in building the data dissemination layer is to provide a single consistent application programming interface (API) to all of the required communication protocols. This API has as its primary interface calls for *open, send, receive, poll and close.* This same set of calls works regardless of the target protocol.

Early in the development of the Spectro-Microscopy Collaboratory we defined a prototype data dissemination library which supported unreliable and reliable unicast and reliable multicast. The unicast protocols were UDP and TCP. The reliable multicast protocol used was Totem[10]. This library was used in development of all the components of the collaboratory, and significantly reduced the development time and training required by the developers. Also, since the application's interface to all three protocols was the same, component developers were able to use unicast mechanisms during development and testing and then switch to using multicast as the program matured. This allowed development of programs that used reliable multicast with only a minimal understanding of the reliable multicast protocol.

An additional benefit realized by having a single common communication library was the ability to upgrade communication capabilities simply by creating a newer version of the data dissemination library component. Further development of the data dissemination component is continuing as a DOE 2000 research and development project. This work will also provide definitions of the interface accessible from Java and from the Common Object Request Broker Architecture (CORBA).

2.3 Policy-based access control in widely distributed environments

Widely distributed systems and collaborative environments that involve multi-user instruments at national facilities, widely distributed supercomputers and large-scale storage systems, data sharing in restricted collaborations, and network-based multimedia channels, give rise to a range of requirements for distributed access control. Among other things, for example, administration of such resources as network quality-of-service will need to be handled by an automated authorization infrastructure so that management of both resource availability and allocation, as well as subsequent enforcement of use-conditions, can be completed automatically and without recourse to a central or single authority.

In all of these scenarios, the resource (data, instrument, computational and storage capacity, communication channel) has multiple stakeholders (for scientific experiments, typically the intellectual principals and policy makers), and each stakeholder will impose use-conditions on the resource. All of the use-conditions must be met simultaneously in order to satisfy the requirements for access.

Further, it is common that scientific collaborations are diffuse, with the principals and stakeholders being geographically distributed and multi-organizational. Therefore the access control mechanism must accommodate these circumstances by providing distributed management of policy-based access control for all resources, authentication, integrity, confidentiality, etc. of resource related information, and mechanisms supporting the internal integrity of distributed systems. The computer systems based resource access control mechanisms should be able to collect all of the relevant assertions (stakeholder use-conditions and corresponding attributes) and make an unambiguous access decision.

Authorization based distributed security is an approach that addresses these general goals. Authorization and attribute certificates are digitally signed documents can assert document validity without physical presence of the signer, or the physical possession of holographically signed documents. The result is that the digitally signed documents that provide the assertions of the principals, stakeholders, attribute authorities, etc., may be generated, represented, used, and verified independent of time or location.

Other parts of the approach are implemented through the use of "authorities" that provide delegation mechanisms and assured information as digitally signed documents: identity authorities connect human entities and system to digital signatures, stakeholder authorities provide use-conditions, attribute authorities attest to user characteristics, etc. Additional components include reliable mechanisms for generating, distributing, and verifying the digitally signed documents, mechanisms that match use-conditions and attributes, and resource access control mechanisms that use the resulting credentials to enforce policy for the specific resource.

The architecture of such an approach is described in[6]. The prototype provides for objects / data / resource owners and other stakeholders to remotely exercise control over access to the resource. Also, legitimate users (those that satisfy the use-conditions of the resource stakeholders) are enabled to obtain easy access, and unqualified / un-authorized users are strongly denied access. The implementation provides a policy engine that implements both flat and hierarchical, multiple use-condition policy models. It uses X.509 identity certificates[4,13], and also ad-hoc attribute and use-condition certificates that are distributed by Web and LDAP servers. The policy engine has been interfaced to the Apache Web server in order to provide its directory based access control (whereby all Web objects as provided with this access control mechanism), and it has been interfaces to an implementation of the IETF's SPKM GSS (a secure messaging API for distributed application development[8]. The initial application of the prototype is for a Web based object archive system[5].

2.4 Instrument control and monitoring

The experiment control and monitoring software is traditionally a closed turnkey system designed to have a single user logged in directly on the experiment console. Since these systems were not designed with remote operation in mind they are often difficult to rewrite or retro-fit. Our approach to these systems has been to modify the control program to provide external access to system state, data, and control variables. We have accomplished this by a combination of techniques, including writing these variables to files and adding support for remote communication connections.

Initially we reconfigured the existing experiment control software to run simultaneously at remote sites and at the Beamline, using status update messages to maintain synchronization. However, it soon became apparent that the original interface developed for control of the Beamline experiments was too complicated and difficult for a remote user to use. The interface assumed that the user was intimately familiar with the instrument and the control interface, which was only true of researchers at the Beamline. Also, the control program allowed operations involving maneuvers associated with sample preparation and positioning that should only be performed by researchers physically at the Beamline. Additionally, it was difficult to provide appropriate feedback regarding inappropriate actions to the remote user.

The experiment control software for the Beamline has now been redesigned to use a client/server architecture. The control program at the Beamline runs a server. Each remote site and even the Beamline site runs a client program that provides the user interface. This client program was designed to be simple, and it contains only those controls and displays that are needed by the user. In the new design we added controls and displays only as deemed necessary by the researchers. This prevented arbitrary complexity in the interface.

Maintenance of the experiment control software was a key consideration in its design. The Beamline equipment and instrumentation is not static, equipment is added or removed regularly. The experiment control and interface programs often need to be updated to reflect these changes. The Beamline scientists must be able to easily update the control and interface programs themselves. This means that the programs must be developed in a programming language the researchers know. Also, the programs must be presented as building blocks rather than a complete control system. The data dissemination component provides one means of achieving this goal by providing an easy to use communication layer that allows the researchers to build the control software without needing to understand the communication mechanisms required.

2.5 Electronic notebook

Electronic notebooks are essential components of a collaboratory infrastructure, as they allow researchers to record all significant information about their collaborative experiments, including ideas, discussions, experiment configuration, experimental data and results, and conclusions from analyzed data. The information, recorded as notebook objects, may be text, images, drawings, plots, data files, voice, video, or any digital data format. The remote participants of a collaboratory view the notebook objects either synchronously—as the objects are created in the notebook by one of the participants, or asynchronously with respect to object creation. Synchronous use of an electronic notebook gives researchers the ability to complement their teleconferences about a collaborative experiment, by allowing them to simultaneously investigate experimental data through the same set of images, plots, data, or text files. Asynchronous use of an electronic notebook is also very valuable, as each researcher may, independently of others, retrieve and view notebook elements, and contribute to the shared experimental record.

Electronic notebooks provide scientists with capabilities not available in paper notebooks, such as automatic recording of data acquisition, voice and video annotations, easy and efficient search of the entire notebook for keywords, dates, results, etc., and the synchronous viewing and authoring of notebook entries. Thus, electronic notebooks provide a medium for organizing, archiving, retrieving, viewing, and communicating all the information developed during a collaboration, whether created by software collecting data from instruments, or in real-time conferences with remote colleagues, or even when working alone. An electronic notebook used in a collaboratory environment captures and documents group understanding. It is a valuable tool to orient new participants of a project, to discuss on-going work, and to document results and conclusions.

Current and past commercial electronic notebook products either do not offer the capabilities needed by collaborating researchers, or do not have the flexibility and extensibility needed by the diverse scientific environment and notebook styles, different computer platforms, and the variety of data types used. Newly emerging technologies can be use to remove some of the barriers that have traditionally limited the development of electronic notebooks suitable to scientific collaborations. Java and the Internet provide an ubiquitous and cross platform development environment. The Common Object Request Broker Architecture (CORBA) provides a flexible, object oriented infra-structure to connect a variety of user interfaces, to notebook services. Other technologies, such as pen input, desktop audio and video recording, and application specific data type support by data base management systems can also be used to improve user interfaces and notebook services.

The DOE 2000 electronic notebook project, itself a collaboration among three DOE laboratories, precisely leverages on these technologies. The electronic notebook developed for the Advanced Light Source (ALS) Collaboratory at LBNL (www-itg.lbl.gov/Notebook/Project.html) is part of this project. The notebook consists of an *user interface*, a *notebook engine*, a *database management engine*, a *data acquisition engine*, and a *visualization engine*.

The *user interface* implements a notebook style required by physicists of the ALS as they remotely participate and conduct experiments. The *notebook engine* communicates with the database management engine in order to input or retrieve objects of the notebook, providing full support for browsing, searching, and visualization of the stored data. Using multicast communications, it communicates with each user interface, in order to update information that has been entered or modified by other users. The database management engine is an extended set of the Object Protocol Model and associated tools [2]. It offers uniform access to multiple databases systems, and responds to notebook engine requests for query, browse, search, or data entry of notebook objects. The data acquisition engine collects data from the experiments via a data acquisition software, and informs the notebook engine about the address of the collected data. Upon receiving this information, the notebook engine creates an electronic notebook entry, which is immediately available to all collaborators for browsing, annotations, plotting, analysis, and searching. The visualization engine has facilities for generating plots and pictures from data stored in the notebook. Currently it consists of a built-in plotting function. In the future, it will include functions supporting data analysis, tracking and steering capabilities, as well as visual representations of search result.

2.6 Common software infrastructure

The underlying infrastructure of the collaboratory environment is a common framework that supports the interoperability of the various components. This framework provides a single integrated environment that is easily extensible and scalable. Each software component can be built as a plug-in module that interoperates with other modules through a common interface, common security, and common communication mechanisms. New software tools are plugged into the framework and register with the resource manager. Other tools contact the resource manager and request use of resources, as needed. A directory service provides resource location and a rendezvous information. Two components of this infrastructure, the data dissemination and security, were described earlier in this paper.

Another essential infrastructure component is a shared file system. Data and result files need to be available to all collaborators. The electronic notebook provides one method of sharing these files, but there are situations during an experiment when pasting these files into the notebook is inappropriate. Our initial approach to providing a secure means of sharing files between sites was to use a commercially available shared file system. The product allowed Kerberos authenticated access to the individual files and directories could be further protected through the use of access control lists. Unfortunately, the product was designed to provide enterprise-wide solutions and did not scale down to very small installations. As an interim solution we are currently using one of the PC, Windows based products that allows secure remote access to the UNIX NFS file system.

The collaboratory environment is composed of many different software and hardware components. One of the challenges in building collaboratories is to provide a single integrated view of the environment for the users. The conference control tool mentioned in the telepresence section integrates the videoconferencing interfaces but this is only a small part of the work that needs to be accomplished in this area. The design and development of a unifying interface that provides a common desktop for all of the collaboratory tools, resource management and an interface to a collaboratory directory service is another one of the DOE 2000 projects under development.

3 Conclusion

A collaboratory prototype at Beamline 7.0 is now largely complete. The remote researchers are able to monitor experiments, use remote control, “walk” around the ALS with remote control cameras, and interact with the researchers at the ALS. Reliable multicast capabilities allow multiple remote researchers to collaborate on the experiment. An electronic notebook was recently added to the environment.

When the common security architecture is in place, the user will be presented with a single login to the collaboratory environment and all credentials for the use of the various tools will be obtained based on that single login. Similarly, when the common communication mechanisms are deployed, they will allow components like the electronic notebook to talk to the experiment control software, conference controller and the videoconferencing software.

There are still many components under development, but the collaboratory environment has already provided significant benefits to the researchers at the ALS, in terms of their ability to interact with remote sites during an experiment. By deploying the collaboratory components to the Beamline researchers at an early stage in the development process we are able to evaluate and refine the design during the development process.

Deployment of the collaboratory environment has reinforced the importance of the primary design goals of ease of use and ease of maintenance. Additional concerns addressed in the design of the collaboratory software include scalability to large numbers of remote users, maintenance of remote sites, hardware platform and operating system portability, software licensing restrictions and cost of commercial software (especially in small research group environments).

For further information about the Spectro-Microscopy Collaboratory, the DCEE program, and the DOE 2000 program see http://www-itg.lbl.gov/BL7Collab, http://www-itg.lbl.gov/DCEE, and http://www.mcs.anl.gov/DOE2000 respectively.

References

[1] Cerf, V. G., "National Collaboratories: Applying Information Technologies for Scientific Research," National Academy Press: Washington, D. C., 1993.

[2] Chen, I. A. and Markowitz, V. M., "The Object-Protocol Model (Version 4)," *Technical Report LBNL-32738 (revised)*, (1996).

[3] Clauer, C.R., et al, 1993 "UARC: A Prototype Upper Atmospheric Research Collaboratory," *EOS Transactions American Geophysical Union*, 267 (74)

[4] Ford, W., Computer Communications Security: Principles, Standards, Protocols, and Techniques, Prentice-Hall, Englewood Cliffs, New Jersey, 07632, 1995.

[5] Johnston, W., Jin G., Larsen, C., Lee, J., Hoo, G., Thompson, M., Tierney, B., Terdiman, J., "Real-Time Generation and Cataloguing of Large Data-Objects

in Widely Distributed Environments," *To be published in International Journal of Digital Libraries - Special Issue on "Digital Libraries in Medicine"*. Available at http://www-itg.lbl.gov/WALDO.

[6] Johnston, W. and Larsen, C., "Security Architectures for Large-Scale Remote Collaboratory Environments: A Use-Condition Centered Approach to Authenticated Global Capabilities," Lawrence Berkeley National Laboratory Technical Report, LBNL-38850. (draft at http://www-itg.lbl.gov/security/publications.html)

[7] Kouzes, R. T., Myers, J. D. and Wulf, W. A., "Collaboratories: Doing Science on the Internet," *IEEE Computer* , Vol. 29, No. 8, August 1996.

[8] Linn, J., "Generic Security Service Application Program Interface", Sep 1993. Available at http://ds.internic.net/rfc/rfc1508.txt. Also see more recent and related drafts at the IETF Common Authentication Technology home page (http://www.ietf.cnri.reston.va.us/html.charters/cat-charter.html) and at http://www.ietf.cnri.reston.va.us/ids.by.wg/cat.html.

[9] Macedonia, M. R., Brutzman, D. P., "MBone Provides Audio and Video Across the Internet," *IEEE Computer*, Vol.27-4, April 1994, pp. 30-36.

[10] Moser, L. E., Melliar-Smith, P. M., Agarwal, D. A., Budhia, R. K., and Lingley-Papadopoulos, C. A., "Totem: A Fault-Tolerant Multicast Group Communication System," *Communications of the ACM*, April 1996, 54-63.

[11] Parvin, B., Taylor, J., Callahan, D. E., Johnston, W., Dahmen, U., "Visual Servoing for Online Facilities," *IEEE Computer*, July 1997.

[12] Schatz, B. R., "Building an Electronic Community System," *Journal of Management & Information Systems*, 1992.

[13] Schneier, B, Applied Cryptography, Second Edition, John Wiley & Sons, 1996

[14] Young, S. J., Fan, G. Y., Hessler, D., Lamont, S., Elvins, T. T., Hadida, M., Hanyzewski, G., Durkin, J. W., Hubbard,P., Kindalman, G., Wong, E., Greenberg, D. Karin, S. and Ellisman, M. H., "Implementing a Collaboratory for Digital Microscopy," *Int. J. Supercomputing Applications*, (1996), 10, 170-181.