

HICSS34  
Minitrack on Developing and Deploying Collaborative Problem Solving Environments  
Maui Hawaii  
January 3-6, 2001

## **Supporting Scientific Analysis within Collaborative Problem Solving Environments**

**Velvin R. Watson**  
Senior Staff Scientist  
NASA Ames Research Center  
Moffett Field, CA 94035-1000  
[vwatson@mail.arc.nasa.gov](mailto:vwatson@mail.arc.nasa.gov)

### **ABSTRACT**

Collaborative problem solving environments for scientists should contain the analysis tools the scientists require in addition to the remote collaboration tools used for general communication. Unfortunately, most scientific analysis tools have been designed for a "stand-alone mode" and cannot be easily modified to work well in a collaborative environment. This paper addresses the questions, "What features are desired in a scientific analysis tool contained within a collaborative environment?", "What are the tool design criteria needed to provide these features?", and "What support is required from the architecture to support these design criteria?."

First, the features of scientific analysis tools that are important for effective analysis in collaborative environments are listed. Next, several design criteria for developing analysis tools that will provide these features are presented. Then requirements for the architecture to support these design criteria are listed.

Some proposed architectures for collaborative problem solving environments are reviewed and their capabilities to support the specified design criteria are discussed. A deficiency in the most popular

architecture for remote application sharing, the ITU T.120 architecture, prevents it from supporting highly interactive, dynamic, high resolution graphics.

To illustrate that the specified design criteria can provide a highly effective analysis tool within a collaborative problem solving environment, a scientific analysis tool that contains the specified design criteria has been integrated into a collaborative environment and tested for effectiveness. The tests were conducted in collaborations between remote sites in the US and between remote sites on different continents. The tests showed that the tool (a tool for the visual analysis of computer simulations of physics) was highly effective for both synchronous and asynchronous collaborative analyses. The important features provided by the tool (and made possible by the specified design criteria) are:

1. The tool provides highly interactive, dynamic, high resolution, 3D graphics.
2. All remote scientists can view the same dynamic, high resolution, 3D scenes of the analysis as the analysis is being conducted.
3. The responsiveness of the tool is nearly identical to the responsiveness of the tool in a stand-alone mode.
4. The scientists can transfer control of the analysis between themselves.
5. Any analysis session or segment of an analysis session, whether done individually or collaboratively, can be recorded and posted on the Web for other scientists or students to download and play in either a collaborative or individual mode.
6. The scientist or student who downloaded the session can, individually or collaboratively, modify or extend the session with his/her own "what if" analysis of the data and post his/her version of the analysis back onto the Web.

7. The peak network bandwidth used in the collaborative sessions is only 1 K bit/second even though the scientists at all sites are viewing high resolution (1280x1024 pixels), dynamic, 3D scenes of the analysis.

The links between the specified design criteria and these performance features are presented.

## **Keywords**

Collaboration, virtual environments, visual analysis, scientific visualization, remote visualization, problem solving environment (PSE).

## **INTRODUCTION**

Considerable research is being conducted on collaborative problem solving environments (CPSEs) because of the major benefits expected. (A list of current CPSE research projects with hyperlinks to the relevant web pages is given in the reference section.) In the past, major attention has been given to video, audio, whiteboards, chat rooms, and document sharing. However, for a scientist, the tools that he/she commonly uses for analysis are often more important than any of these. Therefore, CPSEs for scientists should also incorporate the analysis tools commonly used by the scientists.

Today, scientists are beginning to expect highly interactive, high resolution, dynamic, 3D graphical representations of their data or simulations because of the major improvements in computer graphics (even on the personal computers) and because of the high performance 3D graphics used in computer games. For many scientific applications, high performance graphics is not only expected, it is critical for effective visual analysis of complex data. Therefore, it is desirable for future scientific analysis tools to provide highly interactive, high resolution, dynamic, 3D graphics.

Some very effective tools have been developed for scientific analysis in a single user mode. Examples are Tecplot<sup>TM</sup>, IDL<sup>TM</sup>, Ensight<sup>TM</sup>, and FieldView<sup>TM</sup>. Unfortunately, most of the tools currently used for scientific analysis cannot be easily modified to work well in CPSEs. In the future, to take advantage of the major benefits of collaboration using CPSEs, it will be important to design scientific analysis tools so they can be used effectively in CPSEs.

This paper lists the features desirable in a collaborative analysis tool. Then the design criteria for creating analysis tools that can provide these features are specified. Next, some proposed architectures for collaborative CPSEs are reviewed and their capabilities to support the specified design criteria are discussed. Finally, an example to illustrate that the specified design criteria will provide the desired features in a collaborative scientific analysis tool is presented. The example consists of the integration into a CPSE of a commonly used scientific analysis tool that was created with these design criteria. This system has been tested between sites in the US and between sites on different continents. The results of these tests are presented and the performance is correlated with the design criteria.

## **FEATURES DESIRED IN COLLABORATIVE SCIENTIFIC ANALYSIS TOOLS**

CPSEs for scientists should provide for both synchronous and asynchronous collaborative analysis.

Important features for effective synchronous collaboration during scientific analysis are:

1. A user-computer interface with highly interactive, high resolution, dynamic, 3D graphics.
2. All scientists should be able to see the same view of the analysis simultaneously. Having individually controllable views is also desirable, but an ability to synchronize views is very important.
3. Control of the analysis should be transferable between scientists

4. The system should have nearly the same system responsiveness as if the tools were being operated in the stand-alone mode.
5. The system should provide the same quality of views of the analysis as if the tools were being operated in the stand-alone mode.
6. The scientists should be able to conduct the collaborative session using a network bandwidth commonly available to colleagues.

Additional features important for asynchronous collaboration during scientific analysis are:

1. Scientists should be able to record segments of an analysis session or all of an analysis session and post these for others to replay either collaboratively or individually.
2. Scientists should be able to easily edit and concatenate these session recordings.
3. Remote scientists should be able to easily replay these analysis sessions.
4. Remote scientists should, during replay of these analysis sessions, be able to modify or extend the analyses with their own “what if” analyses and post these sessions for others to replay.

Most scientific analysis tools available today cannot be easily modified to provide these features.

## **DESIGN CRITERIA FOR ACHIEVING THESE DESIRED FEATURES**

Some design criteria for providing the above features are:

1. Utilize the high performance graphics becoming available (even on PCs) to represent the data or simulations with highly interactive, high resolution, dynamic, 3D scenes.
2. Provide a capability for recording a journal file of any segment of an analysis session.
3. Provide a capability for controlling the analysis tool with a journal file. (i.e., a capability to replay an analysis session or any segment of a session.)

4. Provide a capability to easily edit and concatenate the journal files.
5. Provide a capability to condense journal files to contain only the commands needed for effective playback.

## **REQUIREMENTS OF CPSE ARCHITECTURES TO SUPPORT THESE DESIGN CRITERIA**

In order for an architecture to support the specified design criteria, it will need to provide the following:

1. Low latency communication between the remote sites. For synchronous collaboration with highly interactive computer-user interfaces, the interfaces for all sites must keep current with the interactions of the controlling site with nearly no perceptible delay. For example, when interactively manipulating 3D scenes, the scene position must not lag the controlling actions by a perceptible delay or controlling the scene becomes awkward. In addition, all sites should be viewing the same scene position simultaneously. Therefore, the events controlling the interfaces must be passed to all remote sites with low delay. It is unlikely that web based architectures that pass all communications using browsers through http servers (for example, with CGI) will be able to pass events between sites with sufficiently small delays to provide a suitably responsive interactive interface. On the other hand, the tests with RemoteFAST [20] (described in the sections below) show that an architecture that provides dedicated event handling ports between the remote sites does provide a highly responsive interactive interface.
2. Use of intelligent, compact communication between remote sites. Architectures that process high resolution scenes into pixels at one site and then send pixels to all other sites cannot currently provide the dynamic scenes that are desirable (and sometimes even

required) for scientific analyses. (There will be situations where sending pixels may be the best approach, but the designer should be aware that selecting this approach eliminates the possibility of providing high resolution, dynamic graphics with the network bandwidths that are commonly available between scientists.) The basic problem is that scientists currently receive greater than  $10^9$  bits per second of visual information from their workstation screens (24 color bits per pixel,  $10^6$  pixels per frame, and 60 frames per second) whereas the bandwidth that is commonly available between scientists is orders of magnitude less. Furthermore, scientists are reluctant to permit too much data compression for fear of creating visual artifacts that may misrepresent their data. (Even the low resolution scenes used in desktop video have not been widely accepted because of the low quality of the video scenes over networks that are typically available to scientists.)

Network bandwidths are increasing rapidly, and one may argue that networks will soon provide the bandwidths to send pixels over the network fast enough to equal the information bandwidth between the workstation and the user. However, it is likely that the information bandwidth between the workstation and the user will also increase. Workstation processing power has typically increased more rapidly than bandwidth, and it is likely that this power will be used provide larger and more sophisticated displays (such as autostereoscopic displays) and other types of sensory information. (We are not close to exceeding the information processing bandwidth of the human visual system. The human eye has one hundred times more receptors than pixels on the current workstation displays. Therefore, even if we knew how to efficiently map pixel information to receptors, we could increase the bandwidth by one hundred without exceeding the information processing bandwidth of the human visual system.)

Architectures that send window drawing commands (such as the ITU T.120 architecture [27]) also do not provide adequate performance for remote applications that require dynamic, high resolution graphics. To illustrate this, the reader can try to use Microsoft NetMeeting to share any application that uses dynamic, high resolution graphics, such as a VRML browser.

As illustrated in the TEST RESULTS section below, architectures that send application specific data and events for controlling applications that run at all sites (such as RemoteFAST [20]) can provide the high resolution dynamic scenes required for scientific visual analysis if the client computers at each site can render the scenes fast enough (see the next requirement). Fortunately, even the PCs are gaining the ability to render high resolution 3D scenes rapidly.

3. Sufficiently powerful computers at each remote site to support sophisticated client software. The discussion of the previous requirement pointed out the need for adequate computer power at each user site to provide rapid rendering of high resolution 3D scenes. In addition, future computer-user interfaces will call for voice recognition, user tracking and awareness, support for haptic devices, and other features which will become available in the future. These features typically require substantial computing power and flexibility --- much more than is provided by the typical "thin client" computers. (For scientific research, it is still highly cost effective to invest in improving the computer-user interface. Therefore, designing computer-user interfaces for "thin clients" or other "lowest common denominator clients" is not wise.)



4. Event recording and playback (if the scientific analysis applications do not inherently provide this capability). For applications that do not provide event recording and playback, it may be possible to wrap the application so that event recording and playback is done by the wrapper.

## **COMMENTS ON SOME PROPOSED ARCHITECTURES FOR CPSEs**

The most popular architecture for remote application sharing, ITU T.120 [27], is used in NetMeeting and other commercial CPSEs. This architecture is based on running the application on one computer and sending window drawing commands to all other sites. As discussed in the section above, this architecture does not provide adequate support for remote collaboration requiring high resolution, dynamic scenes. (However, there are applications that don't require high resolution, dynamic scenes, and the ITU T.120 architecture may be appropriate for these applications.)

Architectures such as WebFlow [9] (which is the basis for the DOD's portal to high performance computing, the Gateway [8], and the commercial package, Tango [22]) are based on running the applications at all sites with dedicated communication ports for passing events between the applications. The test of RemoteFAST [20], described below, illustrates that this method does provide the high resolution dynamic scenes required for scientific visualization. (WebFlow is based on Java technology.)

Many of the CPSE systems are based on utilizing the Web browsers and CGI with http servers for all communications. This method is unlikely to provide the low latency required for collaboration with applications that use highly interactive computer-user interfaces.

The DOE is developing many building blocks that can be used for CPSEs. For example, some of the basic communications needs for CPSEs, such as reliable multicast, are being developed in the

Collaboratory Interoperability Framework Project (CIF) [2]. The Common Component Architecture Project (CCA) [1], does not yet have a complete implementation, but this project will likely provide components that are especially suited for scientific research.

## **INTEGRATION OF A TOOL WITH THE SPECIFIED DESIGN CRITERIA INTO A CPSE**

A tool commonly used for visual analysis of computer simulations of physics, FAST [20], was integrated into a CPSE to illustrate how the design criteria above can provide the desired features listed above.

(FAST is an acronym for Flow Analysis Software Toolkit.) How FAST achieved the design criteria is discussed first followed by a discussion of how FAST was integrated into a CPSE by taking advantage of these design criteria.

### **How FAST achieved the design criteria**

FAST was designed for high performance, high resolution, dynamic 3D visual analysis of computer simulations of complex physics, and it has been a very popular tool for analysis in computational fluid dynamics. To achieve high performance, FAST launches parallel tasks that are all controlled by a central hub.

The highly interactive, dynamic, high resolution, 3D graphical user interface is achieved by utilizing efficient event handling within the parallel tasks, and by using computers with high performance 3D scene rendering. FAST was designed to run on SGI workstations and currently runs only on workstations. (Fortunately, even PC's can now be obtained with high performance 3D scene rendering, so this design criteria no longer requires very expensive workstations.).

The capability for recording journal (script) files of analysis sessions is achieved by having each parallel task report events to the controlling central hub and having the hub record the events. The event handler

in each parallel task does not directly cause actions within the task. Instead, the handler sends an ASCII text (command script) representation of the action to the hub. The hub in turn records this script in a journal (script) file and then sends the command script back to the parallel task for execution.

The journal file playback (analysis session playback) capability is achieved by having the hub read the ASCII script from a journal file and send the command scripts to the appropriate parallel tasks. The tasks do not know whether the command scripts sent to it from the hub are the result of an event from within the task or from the hub's reading of a journal file during an analysis session replay. An advantage of this is that the scientist can play segments of a previous analysis, then continue the analysis with his/her own "what if" analyses, and then continue with other segments of previous analyses.

The capability to easily edit and concatenate the journal files is achieved by making the script commands ASCII text. Therefore, the files can be modified with a word processor.

The capability to condense the journal file is achieved with a special utility program. The only cause for large journal files in this system is the rapid recording of mouse movements that change the scene viewing position. These mouse movements are recorded very rapidly to provide rapid and accurate response recording. The utility program condenses the large number of very small transformations into fewer transformations that will provide equivalent smooth looking transformations on playback.

### **Integration of FAST into a CPSE for synchronous collaboration**

To create a synchronous collaborative visualization tool, which we named remoteFAST [20], FAST was combined with a program to handle sockets. To start the synchronous session, FAST is launched with the data to be analyzed at all sites. In addition, the program to handle sockets at each site is launched to create a daemon dedicated to efficient passing of events between the sites. During the session, the controlling remoteFAST site simply detects the script commands as they are being recorded into the

journal file and sends the same script commands over the network to all controlled remoteFAST sites. At the controlled sites, FAST simply reads the incoming script commands (as though they were being read from recorded journal files on the local disk) and passes them onto the FAST hub.

This technique provides many advantages. It is simple. The bandwidth between sites need not be large (because only script commands are sent between sites). And, the system response experienced by the users is nearly the same as the response in stand-alone mode. The system response is very good because the dedicated socket daemons provide a nearly unnoticeable delay in sending the script commands over the network, because intelligent information is sent between sites (rather than pixels), and because the 3D scene rendering is performed by the local computer. Therefore, all remote scientists appear to be seeing the same high resolution, dynamic, 3D scenes simultaneously. (See the TEST RESULTS section for details.)

RemoteFAST is normally used along with a desktop video tool if the network bandwidth permits, or along with a normal phone conference if the network bandwidth doesn't permit the video.

These remote collaboration sessions can be recorded and posted onto the Web for other scientists to playback and modify at their convenience. See the next section for details.

### **Integration of FAST into a CPSE for asynchronous collaboration**

To create an asynchronous collaborative visualization tool, which we named FASTexpeditions [20], FAST was wrapped with a C Shell script to permit use with the World Wide Web. The data to be analyzed and the analysis sessions (journal files) are made available from Web pages. Selecting the data from a Web page causes the data to be downloaded to the local computer, FAST to be automatically launched on the local computer, and a script executed to set up the initial state of the analysis. Subsequent selections of the segments of the analysis from the Web page cause the journal files for those

segments to execute. (For most of the investigations that we have posted to the Web, all of the analysis segment journal files are packaged and downloaded along with the initial data because doing this permits playing of any analysis segment without returning to the remote Web server for the journal files. In this case, the URL used on the Web page refers to the downloaded journal files on the local disk, so the Web browser gets these immediately from the local computer disk rather than waiting for the remote Web server to respond and deliver them.)

Sound files can be included in the journal files for an audio description of the analysis as it occurs.

To provide safety from people who might post malicious journal files, the C Shell wrapper scans each journal file and removes unsafe commands.

To facilitate the ease of collaboratively discussing the posted analyses with remote colleagues, the Web pages containing the FASTexpeditions also contain selections for automatically initiating a synchronous collaboration using remoteFAST (described above).

A utility was created to automatically generate a FASTexpedition Web page with URLs pointing to the data from the computer simulation and the journal files of the analysis segments.

## **TEST RESULTS**

RemoteFAST and FASTexpeditions have been tested in collaborative sessions between sites within the US (primarily NASA Ames Research Center in California and the EPA in North Carolina), between sites in Perth Australia and the US (NASA Ames Research Center), between sites in Monte Carlo and the US (EPA), and between sites in France and the US (EPA). Figure 1 shows the computer screen during a session.

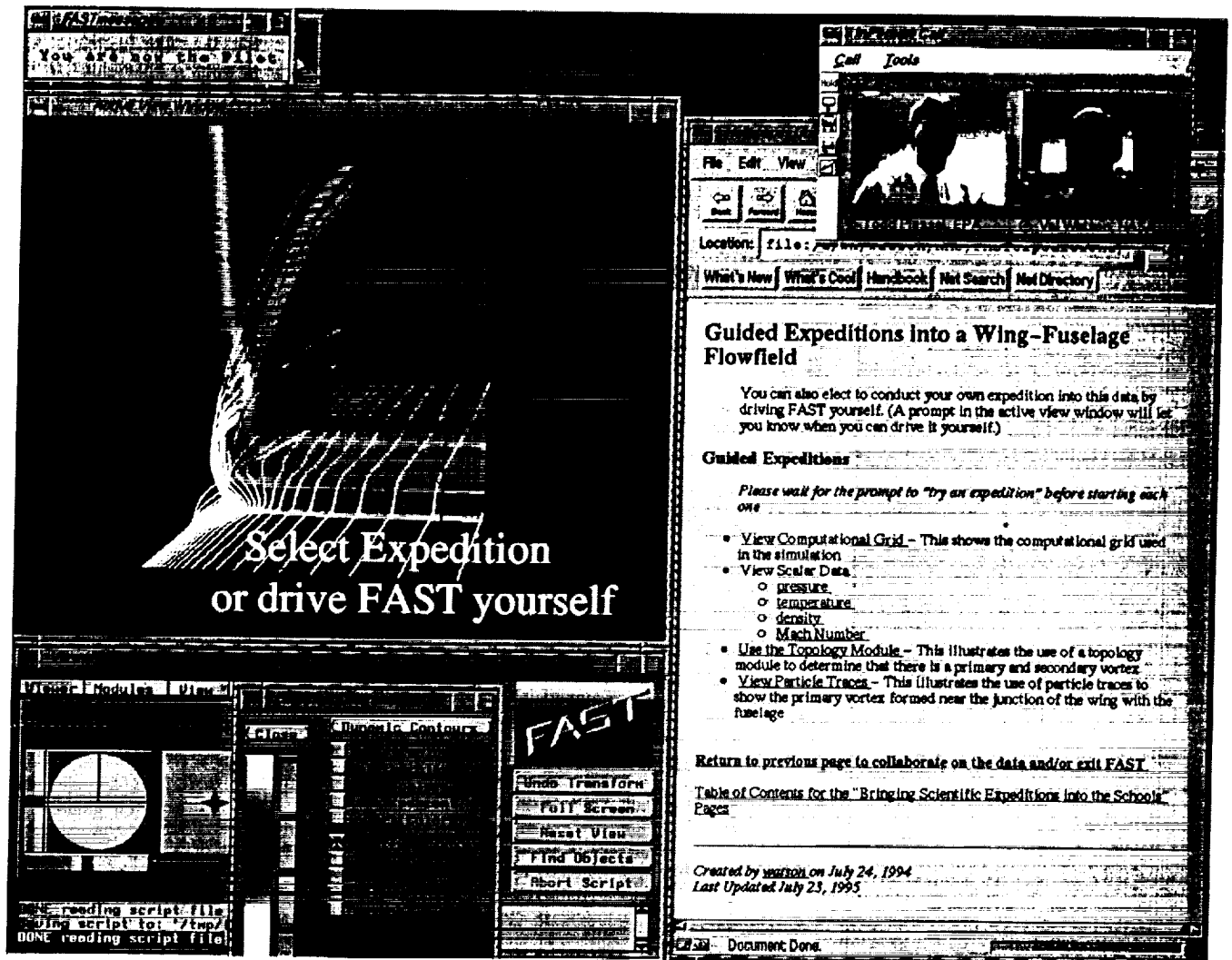


Figure 1. Illustration of a FASTexpedition and RemoteFAST session.

RemoteFAST and FASTexpeditions have proven to be highly effective for synchronous and asynchronous collaboration. The effectiveness of the collaboration was nearly as good as being together in the same office and looking at the same workstation while using FAST for the analysis or for a playback of an analysis. All of the features listed in the "FEATURES DESIRED IN COLLABORATIVE ANALYSIS TOOLS" section above were achieved.

For synchronous collaboration, the response of the visual analysis tool was nearly the same as in stand-alone mode. All sites were able to view the same high resolution (1280x1024), dynamic (greater than 10 frames per second), 3D scenes simultaneously. (Individual sites could independently control their own scene viewing position, but the viewing position could also be resynchronized with the controlling site's viewing position.) Control of the analysis was easily transferred between sites. The bandwidth utilized between sites during a remote collaboration session was measured to peak at less than 1 K bit/second. Note that this low bandwidth utilization and high display performance is achieved by sending script commands over the network and by having the local computer create and render the scenes. This performance cannot now (and will not in the near future) be achieved by sending pixels over the network. Even systems that send scene graphs (such as VRML files) over the network do not match this performance.

For asynchronous collaboration, the analyses posted on the Web were easily downloaded and played. After the initial data download, the playback performance was identical to the performance of playback from journal files on the local disk.

Stereo glasses were often used to obtain stereoscopic scenes in both synchronous and asynchronous modes.

The major advantages of FASTexpeditions (wherein data and journal files of analysis sessions are posted on the Web) over VRML or movie files posted on the Web are:

1. The 3D display performance is superior.
2. Viewers download the actual data and can perform their own "what if" analysis on the data.
3. Viewers can modify the analyses they download and post their own analyses back on the Web.

4. Viewers can collaboratively review and modify the posted analyses with remote colleagues, and these analyses can be posted back onto the Web.

RemoteFAST and FASTexpeditions were used in conjunction with InPerson™, SGI's desktop video conference tool, whenever the network bandwidth was high enough (i.e., between France and the US and between sites within the US), and ordinary phones were used instead of InPerson™ when the network bandwidths would not support satisfactory desktop video (i.e., between Monte Carlo and the US and between Australia and the US).

The scenario used most often to demonstrate the features of FASTexpeditions and remoteFAST follows:

1. A scientist goes to a Web site where FASTexpeditions (data from computer simulations of physics and journal files of various analyses of the data) are posted.
2. The scientist selects one of the FASTexpeditions and views several of the analyses of the data.
3. The scientist then extends the authors analysis with his/her own "what if" analysis.
4. The scientist then contacts the author of the posted data with a phone or InPerson™ and asks the author about one of the features seen in an analysis.
5. The author and the scientist then both initiate a remote collaboration (by making selections on the Web page to automatically start remoteFAST).
6. The author and the scientist then use remoteFAST collaboratively to investigate the feature.

Typically, the desktop video was only used at the beginning of the collaborative session when establishing initial contact. When the interest shifted from the initial "hello" to the analysis of the data, the primary focus was shifted to the 3D scenes of the visual analysis process and to the audio.



## **SUMMARY**

CPSEs have the potential for a major impact in scientific research. To achieve this potential, it is important to include the scientist's favorite analysis tools within his CPSE. However, most current scientific analysis tools cannot be easily modified to work well in a collaborative environment. Therefore, when designing future scientific analysis tools, it is important to design them to work well in a collaborative environment. The features desired in a collaborative analysis tool were listed, and the design criteria for the analysis tool to achieve the desired features were provided.

The features required in CPSE architectures to support the specified design criteria were presented. A review of some proposed CPSE architectures indicated that some do not support the specified design criteria. In particular, the popular ITU standard, T.120, does not support highly interactive, dynamic, high resolution, 3D graphical interfaces.

To demonstrate that the design criteria would provide the desired features, a popular analysis tool that contains the design criteria was incorporated in a CPSE and tested. The tests showed that the tool was very effective for both synchronous and asynchronous collaboration and it provided all of the desired features listed.

## **ACKNOWLEDGMENTS**

FAST was created by the team listed in reference #19. Jean Clucas, John West, and Tim Sandstrom were especially helpful in creating FASTexpeditions and remoteFAST. Tim Sandstrom also provided valuable support for users of these programs. Todd Plessel, with the EPA in Research Triangle Park in North Carolina, installed these programs in the EPA and helped extensively with testing and with many demonstrations.

## REFERENCES

### Current Research on CPSEs by Organization

#### Department of Energy

##### 1. Common Component Architecture (CCA)

(CCA will provide components especially valuable for scientific research)

Contact - Rob Armstrong - Sandia

Website - <http://z.ca.sandia.gov/~cca-forum/>

##### 2. Collaboratory Interoperability Framework Project (CIF)

(CIF will provide fundamental communications needs to build on)

Contact - Deb Agarwal - LBL

Website - <http://www-itg.lbl.gov/CIF/>

##### 3. Advanced Visualization Communications Toolkit

(This will provide network aware components to optimize visualization)

Contact - Deb Agrawal - LBL

Website - <http://www-itg.lbl.gov/~deba/NGI/AdvVizComm.html>

##### 4. Corridor One Project

(High performance visualization over very high bandwidth networks)

Contact - Rick Stevens - ANL

Website - <http://www-fp.mcs.anl.gov/fl/research/Proposals/co.htm>

##### 5. Toolkit for Collaboratory Development

(Includes Core2000 – a collaboratory research environment based on Habanero from NCSA)

Contact – Jim Meyers – PNL

Website - <http://www.emsl.pnl.gov:2080/docs/collab/>

#### 6. DOE2000 Electronic Notebook Project

(Electronic notebook projects at PNL, LBL, and ORNL)

Main Website - <http://www.epm.ornl.gov/enote/>

Contacts

Jim Myers - PNL

Sonia Sachs- LBL

Al Geist - ORNL9. DOE2000 Collaboratory Research

(Basic collaboratory research projects)

Website - <http://www-unix.mcs.anl.gov/DOE2000/collabs.html>

#### 7. DOE2000 Collaboratory Pilot Projects

The Diesel Combustion Collaboratory

Website - <http://www-collab.ca.sandia.gov/Diesel/ui/>

The Materials MicroCharacterization Collaboratory

Website - <http://tpm.amc.anl.gov/MMC/>

Environmental Molecular Sciences Collaboratory

Website - <http://www.emsl.pnl.gov:2080/docs/collab/>

Fusion Collaboratory

Website - <http://www.fusionscience.org/collab/REE/>

### Department of Defense

#### 8. The Gateway

(A portal to high performance computing. This uses a three tier architecture built from WebFlow and Globus)

Contact - Marlon Pierce - WPAFB

Website - <http://www.osc.edu/~kent/theGateway>

## NSF and Universities

### 9. WebFlow

(A visual programming paradigm for Web/Java based coarse grained distributed computing. This is based on Java technology.)

Contacts

Tomasz Haupt - Syracuse University

Wojtek Furmanski - Syracuse University

Website - <http://www.npac.syr.edu/users/haupt/WebFlow/>

### 10. SCIRun

(This provides computational steering)

Contact - Christopher R. Johnson - University of Utah

Website - <http://www.cs.utah.edu/sci/publications/scitools96/>

### 11. NCSA Collaboration Systems

(Habanero provides state and event synchronization for multiple copies of a software tool. It utilizes Java.)

Contact - Polly Baker - NCSA

Website - <http://havefun.ncsa.uiuc.edu/>

### 12. PUNCH

(An Architecture for Web-Enabled Wide-Area Network-Computing)

Contact - Prof Jose Fortes - Purdue Univ

Website - <http://punch.ecn.purdue.edu/>

### 13. DISCIPLE

(Distributed System for Collaborative Information Processing and Learning)

Contact - Dr. Ivan Marsic - Rutgers Univ

Website - <http://www.caip.rutgers.edu/multimedia/groupware/>

#### 14. Space Physics and Aeronomy Research Collaboratory

(an environment for collaborative research in space physics and aeronomy)

Contact - Gary Olson - Univ of Michigan

Website - <http://intel.si.umich.edu/sparc/>

#### 15. Stanford Interactive Workspaces

(For exploring possibilities for people to work together in technology-rich spaces)

Contact - Terry Winograd - Stanford

Website - <http://graphics.stanford.edu/projects/iwork>

#### 16. High Performance and Real Time Corba

(Research on improving the throughput performance and reducing latency of Corba)

Contact - Doug Schmidt - UC Irvine (Wash Univ.)

Website - <http://www.cs.wustl.edu/~schmidt/corba-research-performance.html>

### NASA

#### 17. Intelligent Synthesis Environment (ISE) and Collaborative Engineering Environment (CEE)

(NASA's project to create collaborative analysis and design environments)

Contact - John Malone for ISE – NASA Lewis Research Center

Website - <http://ise.nasa.gov>

Contact - Ed Chow for CEE – NASA Jet Propulsion Laboratory

Website - <http://ce-server.jpl.nasa.gov/>

#### 18. Science Desk

(A project to create collaborative research environments with AI support)

Contact - Rich Keller – NASA Ames Research Center

Website - <http://sciencedesk.arc.nasa.gov>

#### 19. Mars Web Pages

(A website for the collaborative selection of Mars landing sites)

Contact - Glenn Deardorff – NASA Ames Research Center

Website – <http://marsoweb.nas.nasa.gov/landingsites/>

#### 20. FASTexpeditions

(Tools for asynchronous and synchronous collaborative scientific visualization)

Contact – Val Watson – NASA Ames Research Center

Website – <http://www.nas.nasa.gov/Software/FAST/FASTexpeditions>

### Industry

#### 21. Intelligent Human-Computer Interaction

(An environment for collaboration based on rooms. Awareness and privacy issues are addressed.)

Contact - Samuel Bayer - Mitre Corp

Website - <http://www.mitre.org/resources/centers/it/g063/hci-index.html>

### Commercial CPSE Systems

#### 22. Tango Interactive

(Based on WebFlow from Syracuse University)

Contact - Marek Podgorny - WebWisdom

Website - <http://www.webwisdom.com/tangointeractive/>

## **CPSE Organizations**

### **23. Computingportals**

Home Page - <http://www.computingportals.org/>

Survey of CPSE projects - <http://www.computingportals.org/projects>

### **24. Computer Supported Cooperative Work (CSCW)**

Website - <http://www.acm.org/sigchi/cscw2000/index.html>

## **Reports on CPSEs**

### **25. Report on Collaborative Virtual Environments 1998**

University of Manchester, UK, 17-19th June 1998

Elizabeth Churchill and David Snowden

<http://www.fxpal.xerox.com/ConferencesWorkshops/cve/Report.htm>

### **26. Workshop on CPSEs for Scientific Research**

San Diego, CA., 29 June – 1 July, 1999

<http://www.emsl.pnl.gov:2080/docs/cpse/workshop/index.html>

## **CPSE Standards**

### **27. International Telecommunication Union's Proposed Standards**

Complete listing of proposed standards - <http://www.itu.int/publications/telecom.htm>

Proposed standard for application sharing – <http://www.itu.int/itu-t/rec/itu-t/rec/t/t120.html>

Proposed standard for audiovisual and multimedia systems - <http://www.itu.int/itu-t/rec/itu-t/rec/h/h323.html>