Java Object-Sharing in HABANERO

A new framework for collaborative tool development uses any platform that supports Java.

The National Center for Supercomputing Applications (NCSA) at the University of Illinois hosts University and corporate supercomputer research teams distributed around the world. Typically, the various research teams use a wide variety of workstation hardware and operating system software. It is common for teams to evolve domain-specific tools, or to purchase proprietary tools, to assist in special-purpose calculations, visualizations, and data analysis.

With routine Internet use in much of the world, it has become possible to consider designing tools that support collaboration. Whereas network communications links previously required semi-custom solutions for each user community, Internet/Web-based communications are capable of reaching a much larger group of people.

Habanero™ was written to facilitate the use of real-time multi-user software tools in education and the sciences. Unlike earlier projects (such as NCSA Telnet, NCSA Collage, NCSA Mosaic), where different source code was used for each supported hardware platform, the Habanero framework supports multiple hardware platforms by virtue of implementation in the Java programming language from Sun Microsystems. Implementation in Java necessitates, with minor exceptions, production of only a single source code. Beyond the obvious economic advantages of a single source code base, Java allows passing instantiated objects between hardware platforms.

Shared software tools quickly become important as distributed, virtual, ephemeral work teams are used to pool resources. Current collaboration practice frequently involves travel, face-to-face meetings, and the use of printed material and photographs or drawings in an attempt to communicate underlying concepts, design rationale, and the range of design options considered. In complex problem domains such as CAD models, numerical visualization displays, and mathematical models, this very basic style of communications generally does not provide sufficient expressive power. Further, when attempts are made to actually conduct work in such ways, the resultant amalgam of fax transmissions, overnight shipments, and telephone calls significantly burdens the process. If the principal design and analysis tools in use by the scientists and engineers could themselves be shared between colleagues, a team member could very

Annie Chabert, Ed Grossman, Larry Jackson, Stephen Pietrowicz, and Chris Seguin

quickly show the particular result under discussion to the rest of the team. An example of multi-tool synchronous collaboration in a medical conferencing setting is shown in Figure 1. Here, the collaborators are utilizing chat, whiteboard, and MRI gateway tools in order to best communicate various forms of information.

There are obvious advantages, in both salary expenses and time-to-market considerations, in using a single source program that can be run on the entire range of hardware platforms. Portable compiled byte code reduces the administrative burden of supporting multiple hardware platforms. However, platform-native program code is used to gain access to hardware devices in the absence of suitable Java Applications Programming Interfaces (APIs), to improve processing speed in certain instances, and to capitalize on existing software investments. Some platform-specific code is contained within two tools that are shipped with Habanero, but the framework itself is entirely written in Java.

The *Wrapped Object* Concept

We designed the Habanero framework with the goal of making it as simple as possible for a tool developer to make a sharable application, either by altering an existing single-user application or by developing a new application from scratch. This communications code is the only code we require the application developer to implement. Habanero creates a surrogate for each main application object; they communicate via the Habanero API. We only require the application developer to implement this communication code.

The *wrapped* interface specifies four methods that allow an application to display itself in a window, transfer program state to and from other clients, and to handle events which come from other clients. Developers may opt to utilize additional features for further control over multi-user behavior. An interface is one of two mechanisms for inheritance provided by the Java language. An application programmer needs to write, or extend, the four methods the Habanero wrapped interface specifies in order to acquire multi-user capability.

When designing a collaborative application, it is important to specify the granularity at which actions should be shared. The granularity could range from highly abstract concepts to low-level events (for example, sharing every mouse interrupt or graphics drawing occurrence). While Habanero allows a developer to create sharable actions of any degree of abstraction, our default granularity is a semantic event, such as the result of pressing a button or choosing a menu item. The Java Abstract Windowing Toolkit (AWT) has a similar notion that it calls an action event. The Habanero default strategy is to intercept each AWT action event, include it in a Habanero object, and execute the result at each Habanero client. Habanero makes it easy to intercept other AWT events (such as mouse moves) or to define arbitrary new actions. Application designers choose exactly which actions should be shared.

Further, each application may choose its own rules of behavior, or floor control. By default we use free for all, wherein each user sees the same actions in the same order, and no action is ever rejected. Habanero also provides arbitrators that enforce locks, or turn-taking. Developers can extend the existing arbitrators to perform application-specific functions. One interesting use of arbitrators is to enforce the rules of a game. We include a checkers game to illustrate this point.

Habanero uses its surrogate for the wrapped application object to determine which tool handles a newly received collaborative action. This works well when each tool has only one object handling events. More complex applications will generally have a number of windows, each with its own event-handling path, and Habanero provides additional facilities for them. If the windows are based on the same underlying data, they can register as views of the surrogate. Each view is notified when a collaborative action is sent to the surrogate. Windows that operate on different data get surrogates that are children of the application's surrogate, allowing Habanero to send collaborative actions directly to them.

Additional interfaces provide routines that are called by Habanero when an application is started or stopped. One example tool uses this device to acquire its initial parameters from one user, and only subsequently becomes visible to other participants.

Sharing Information between Applications

A brief overview of how the Habanero framework shares information between clients is appropriate. We discuss an executable data object, an action, the connections, and the arbitrators.

The medium of exchange in Habanero is an action. The executable data structure can store anything from mouse clicks and key presses to images and user-defined objects. In many applications, the client's choices are transparently sent to the arbitrator. Additionally, the framework supports transfer of larger changes. First, a programmer can create a new action, a facility currently only used in the framework. Sec-

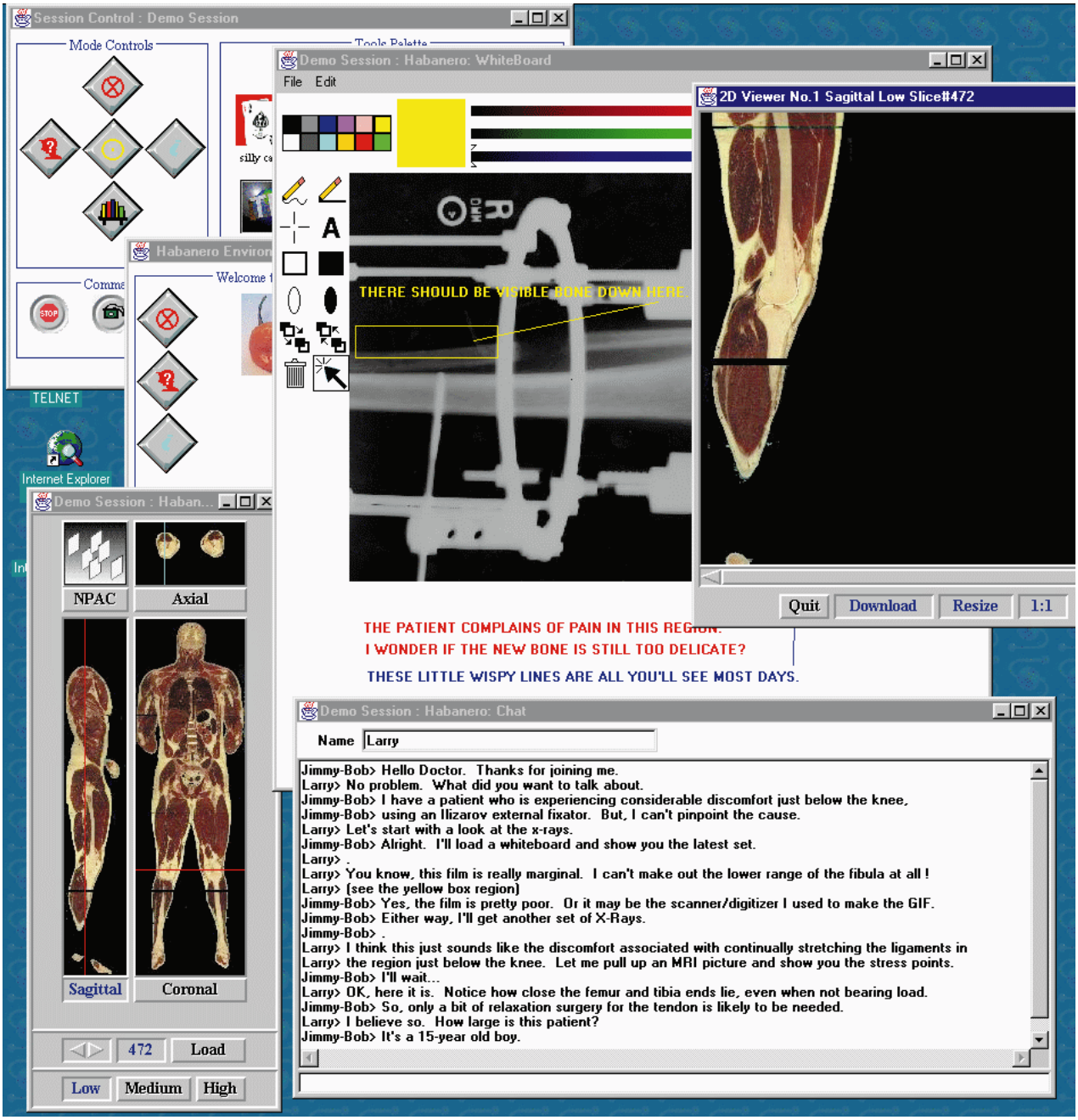

Figure 1. Medical teleconference utilizing X-ray annotation, MRI database gateway, and other tools.

ond, the programmer can create an object that can be written to, and restored from, a stream. The object is delivered to a specific method on all clients.

Arbitrators that only process small actions often share a pipe. Arbitrators processing larger actions may request a pipe, allowing others to continue processing smaller actions. At present, arbitrators make a TCP/IP socket connection to each client. This obviously will not scale well, hence investigation of multicasting to large numbers of clients.

Although we talk about actions being sent to an arbitrator and then distributed to all clients, it's shorthand for a more complex procedure. First, an application requests permission to perform a class of actions. Next, the application sends the specific action to be shared. The arbitrator examines the action to determine if it is legal. This allows a central check against programmers who would rewrite application code to perform illegal actions. In addition, it allows the programmer to separate the action's legality from the action's effects. Illegal actions are discarded, and legal actions forwarded to all clients. Within the client, actions are executed in the order of examination by the arbitrator.

Habanero Tools

The Habanero download includes many tools, to give users a glimpse of the types of collaborative work the system can support, and to help new developers understand use of the framework. Additionally, tutorial materials concerning changing applets to shared applications are available on our Web site. These tools illustrate how the framework solves different multi-user collaborative software development problems of different categories. Some tools are used primarily to communicate, others coordinate, and others are domain specific production tools.

The Existing 1.0 Final Tool Set: Two programs that include source code are the Drawing Tool and Card Demo. Both of these were originally demos distributed with the Java JDK from Sun Microsystems. Slightly modified versions are distributed with Habanero to show how to change existing applications into shared tools.

Users in a session can communicate through a text-based chat utility. This chat utility does not utilize the Internet Relay Chat protocol, but shares an array of character strings.

An audio chat tool is also available. Native code libraries, called from a Java control layer, perform GSM audio compression. At this time, only Sparc machines running Solaris can use the audio tool.

A shared white board allows image files to be retrieved from a user's local disk, or from the Web, or from clipboard paste from another tool, then annotated using our mark-up tools. Source code is provided for instructional purposes.

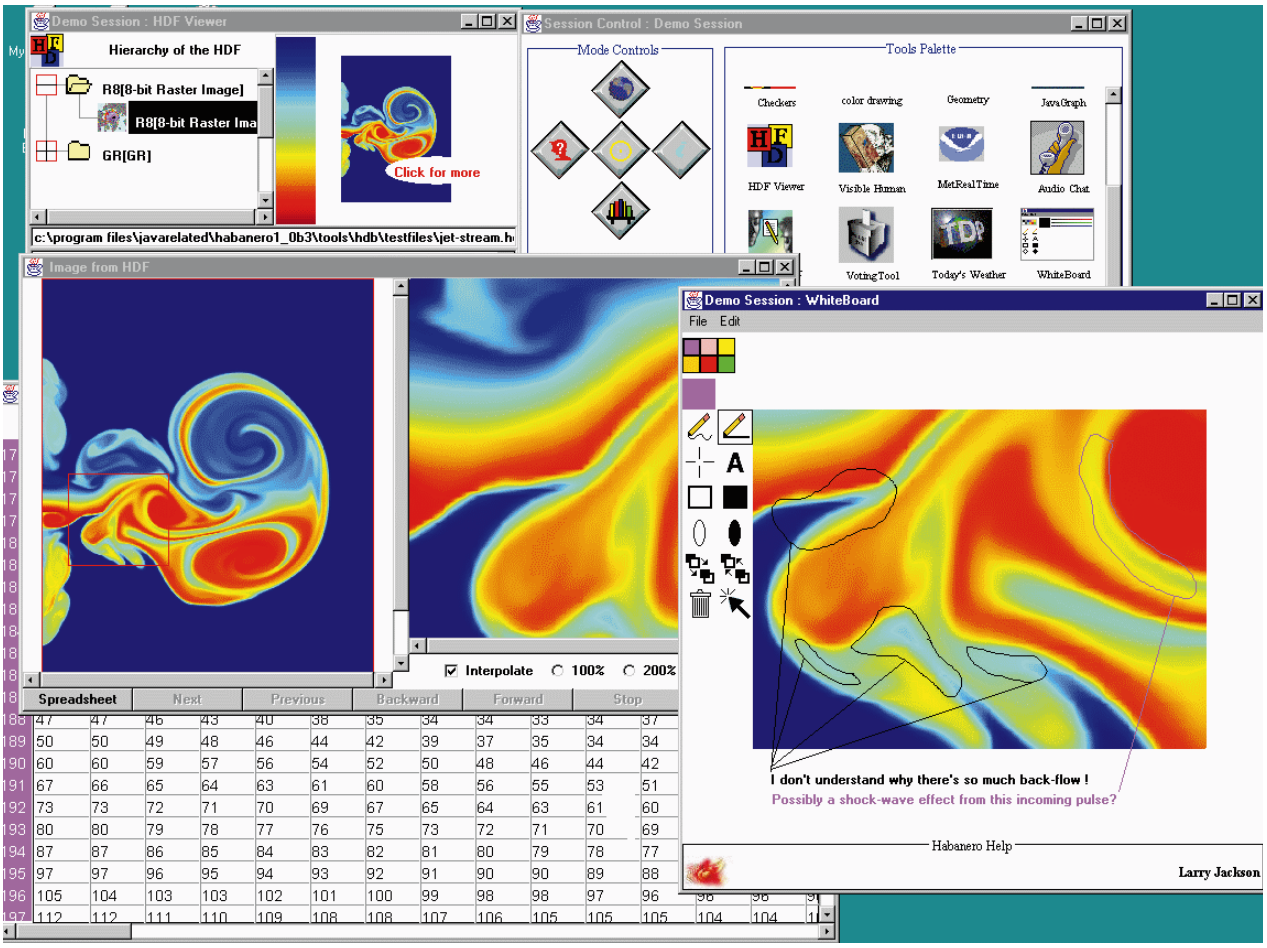

Figure 2. Hierarchical Data Format dataset visualizer browsing gas jet flow results.

The Weather Visualizer was originally an applet written by the University of Illinois Department of Atmospheric Sciences. Current weather satellite imagery is loaded from the Web, with a series of overlays. Participants use the tool to bring up different weather photographs, overlays, and statistics. We added the multi-user capability, and a shared pointer, to facilitate distance education applications.

Michael Chang and Paul Coddington of Syracuse University created the Visible Human applet, which we modified for multiple users. This application

front-ends a database of MRI-scanned cross sections of a human male from the National Institutes of Health. Selection and viewing of cross-sections are now shared, and our clipboard allows export of images to the Habanero whiteboard. This application was initially ported to Habanero in four hours.

To demonstrate the flexibility of the arbitration mechanism, an arbitrator was created that understands, and enforces, the rules of Checkers. The source code illustrates the use of our locks and keys mechanism.

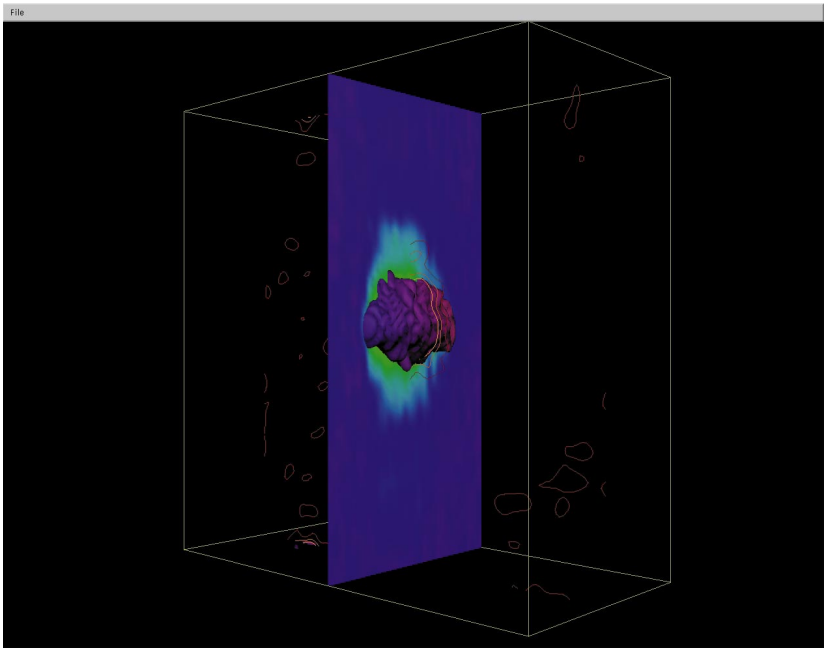

Figure 3. Visualization of the star IRC+10216, expelling two different hydrogen cyanide molecule isotopes

David E. Joyce at Clark University has created an applet in connection with an instructional project based on Euclid's Elements. His Geometry applet has been modified as a demonstration vehicle in distance education.

Eric Gilling created a Java engineering calculator while a student at Urbana High School. After his graduation, he modified it for multiple users.

We provide a collaborative text editor for ASCII files. Changes/annotations by differing individuals are indicated in different colors. Although far less feature-rich than commercial word processors, this basic tool hints at applications in development of contracts, specifications, press releases, and other documents where the wording must be agreed to by multiple parties.

We've also created a utility to allow NCSA Mosaic Web browsers to be controlled collaboratively. By sending messages to Mosaic's TCP/IP-based Common Client Interface (CCI), the WWWSession utility can control the actions of the other Mosaic browsers in the session, whether the browser is running the Unix or Windows-95/NT version of Mosaic. The tool can be extended to control other browsers via their proprietary command dialects.

Collaboration sessions sometimes require that participants reach some sort of agreement. We have included a voting tool that allows public or anonymous voting on a question. The results are displayed after the vote has been completed, or when the time allowed expires. The arbitrator delivers the tally, complete with pie-chart graphics, as per the dissemination instructions of the owner when the tool was activated. Although the tool is configured initially by one user, the arbitrator controls tool execution to preclude tampering with the results.

The former NCSA Relativity group, now at the Einstein Institute in Potsdam, Germany, created a 2D graphing utility to collaborate with colleagues in remote locations. It graphs 2D data and allows zooming and annotation.

A Virtual Wind Tunnel was created by David Oh and others at MIT and modified for multiple users. Synchronizing the displays of a variety of hardware types necessitated coordinating between machines in order to compensate for different processor speeds. Centrally broadcast time ticks, with acknowledgment by each client, are used to ensure all participants are viewing the same display at the same moment. Coordination of the pause control was added so users always see the same frame. This is the first of the Habanero tools where it was necessary to perform this level of client cross-checking to ensure consistency.

The NCSA Collaborative HDF Viewer facilitates shared analysis of Hierarchical Data Format files. HDF is a self-describing data format and access libraries used by NASA and the Japanese Space Agency for image archival. A sample session using this tool is shown in Figure 2. Further work is underway to segment, and preposition very large data files to minimize network delays when working with files approaching a terabyte in size. The user interface and display code is written in Java, but the large HDF libraries are written in C.

Current Work

NCSA and Habanero users continue to develop and adapt tools. The following are representative developments.

NCSA has modified an applet from the NOAA

Pacific Marine Environmental Laboratory to allow NOAA researchers to view real time Tropical Atmosphere-Ocean (TAO) array meteorological data.

NCSA Astro3D, shown in Figure 3, is a collaborative 3D astronomy tool that displays radio astronomy telescope data. It is being created with guidance from Raymond L. Plante of the Astronomy Digital Image Library project [6]. The tool uses the Java 3D API and allows users with high-end equipment, such as an ImmersaDesk or the CAVE virtual environment, to collaborate with users with low-end workstations. Participants in the collaboration can steer the camera for all users in the virtual space, create and share 3D pointers, annotate regions, and save their work in VRML.

The Nethematic system created by Paul Grimm from Fraunhofer Institutes, Darmstadt, Germany, allows mathematicians to create a network of cooperating compute servers to distribute a computation over multiple machines. Using Java RMI, the client program parses the formula and invokes the calculation facilities of each servers. Functions are calculated once, but the results are displayed to multiple colleagues using Habanero. This application demonstrates a solution of the general need by Habanero to avoid redundant calculations. Further, the task management and load-sharing provisions demonstrated have applicability to the much larger computational tasks generally allotted to supercomputers. Figure 4 shows the system in operation between two clients.

We are prototyping clients using the CORBA IDL to interface with the large amount of legacy code that exists. Our first application is a tic-tac-toe game that uses CORBA to communicate. A collaborative effort is now underway with the Nuclear Regulatory Commission and the Civil Engineering Department at UIUC to share a large plotting program via Habanero. The original code is Fortran, with graphics using the X window system.

Researchers from the VOSAIC/Video Datagram Protocol group of the Computer Science Department at UIUC have developed a Java movie viewer, wrapping native-code libraries for MPEG decompression. A pure Java version has also been developed. Collaboration features such as synchronized pause and exporting images to the whiteboard will also be added. The VOSAIC/VDP adaptive protocol will also be incorporated, to make best use of bandwidth that is time-varying.

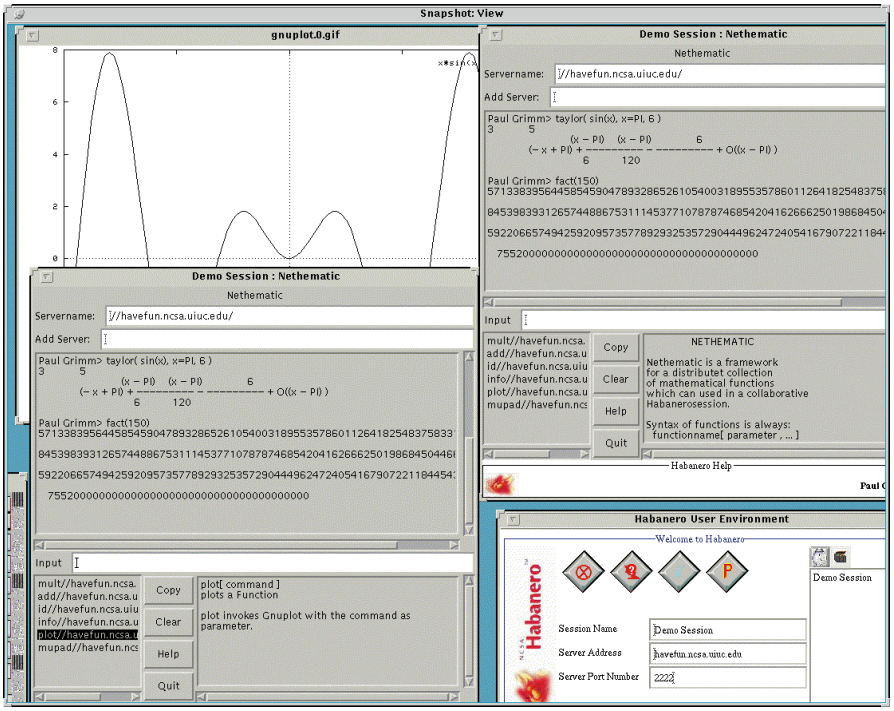

Figure 4. Nethematic mathematical calculations using multiple computer hosts and Habanero dissemination of results.

Related Work

Several toolkits support building collaborative applications. Commercial products such as Lotus Notes now contain extensive collections of asynchronous collaboration support tools. Some systems, such as MMConf [1], Distview [5] and Rendezvous [4], support real-time collaboration by providing generic facilities for floor control, network routing and session management. Others, such as wOrlds [7], Team Rooms [6], and CBE [3] offer APIs to build collaborative spaces (respectively called locales, team rooms, rooms). Because they manage both synchronous and asynchronous collaborations, they place more emphasis on session management. Consequently data, collaborative spaces, and group and user identities must be persistent.

Current versions of the Habanero are closer to the former set of toolkits, although built with Internet technology and using Java to avoid platform dependence. Habanero is currently being extended, as part of the DARPA-funded project Integration of Synchronous and Asynchronous Collaboration (ISAAC), to provide support for asynchronous work as well. ISAAC will extend Habanero's session model to include asyn-

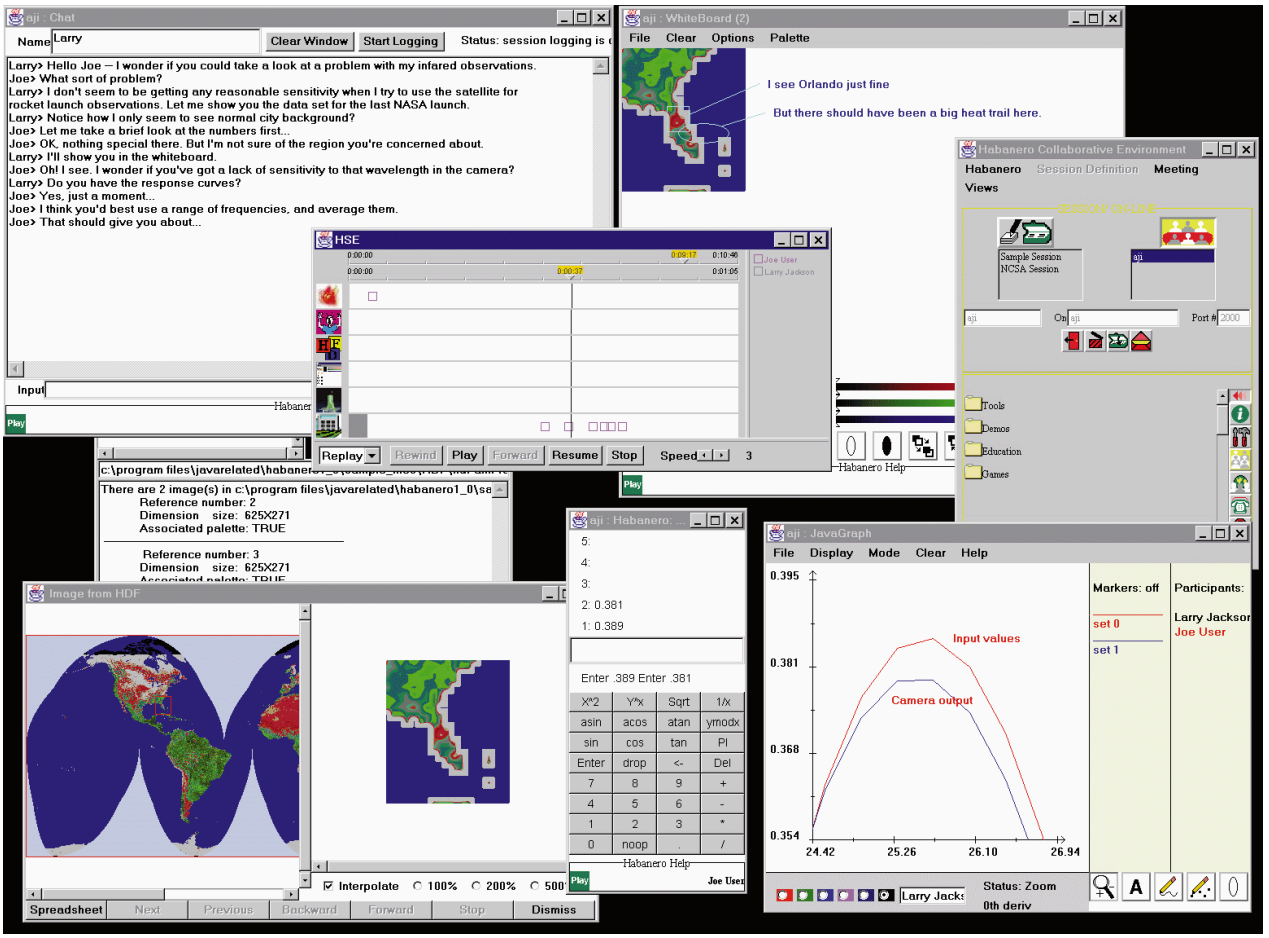

Figure 5. Fast replay of a multiple-tool collaboration sessions

chronous participants. Additionally, Habanero supports record and replay (Figure 5), which allows a recording of a synchronous session to be made available asynchronously to interested parties.

A family of tools support real-time collaboration via low-level sharing of screen images and mouse or keyboard events. Of these, a current example is NetMeeting from Microsoft. Such systems typically display a great deal of hardware and operating system dependence. Although highly expedient in providing a remote viewing and operability capability for a very limited numbers of colleagues with the supported brand of workstation, support is not now provided for multiplatform work, for sharing of programmer defined abstractions, or for elective temporary suppression of the sharing process. The Habanero design philosophy, in contrast, promotes tool-specific variability in sharing decisions, and the use of abstraction by the tool designer.

Promondia, like Habanero, uses Java for building groupware applications. However, Promondia is Web browser-oriented. Promondia is an open system that supports collaborative applets. The advantage of such a system is that new or updated applets can be downloaded dynamically from the server without any local administration. Conversely, Habanero is application-oriented, providing direct access to data stored on local machines as well as on a server. Two planned features, dynamic download of new collaborative applications and a modified environment that would be accessible as an applet through a Web page, will address Habanero's failings in this area. Nonetheless, given the current security restrictions on applets in Web browsers, we feel a standalone application that allows reading from and writing to local disk and arbitrary socket connections is ultimately more useful.

Conclusions

The Habanero framework has proven to be extremely capable, and usable in a wide variety of applications. In particular, it provides a strong capability to host adaptive or negotiated protocols wherein the various machines need to exchange configuration and control information, in addition to the tool-specific information.

Collaborating using homogeneous functionality, but heterogeneous presentation, is another promising area of investigation we are pursuing. At NCSA, the results

of real-time computationally intense modeling and visualization work at supercomputer centers could be deployed to users having access only to more limited processors or Internet bandwidth. Further, using a subset of the display functionality of the high-end version of a tool need not invalidate the notion of using that tool in collaboration with hardware of less capability.

Adding synchronous multiplatform collaboration to existing tools is readily possible. This allows immersive communication within the tools of the problem domain, and we look forward to the use of more expressive tools in complex domains. Delays in convening distributed teams are also avoidable through mechanisms such as Habanero, as extremely ad hoc meetings are supportable. As these new facilities add to the communications options people already have, they are desirable. As they constitute a functionally new capability, without parallel on isolated workstations, their exact value is uncertain at this time.

It is possible in this phase of Internet's development to consider specific designs of tools encompassing real-time collaboration functions. Existing tools were generally not designed with this capability in mind. So, we are at a turning point in tool design philosophy. Whereas formerly, of necessity, people worked with computers by themselves and communicated separately from this work, the tools themselves can now incorporate domain-specific collaboration facilities. We believe this is an important capability, especially in computational science and education.

However, based on our discussions with user communities, many people are not yet accustomed to the idea of incorporating remote collaboration in their work. Although it is possible to consider the impact of such a change on the practice of work sociologically, only a very small percentage of the population has begun to consider such changes on a personal level. We believe it will be desirable, albeit different, and as such will require some reinvention of the work process. C

References

1. Crowley, T., Milazzo, P., Baker, E., Forsdick, H. and Tomlinson, R. MMConf: An infrastructure for building shared multimedia applications. In *Proceedings of CSCW'90* (1990), 329–342.
2. Grossman, E. and Kothari, J. Neighborhoods: A protocol for facilitating synchronous collaboration. In *Proceedings of the 5th International WWW Conference* (1996), Posters Notes, 111–118.
3. Lee, J., Prakash, A., Jaeger, T., and Wu, G. Supporting multi-user, multi-applet workspaces in CBE. In *Proceedings of CSCW'96* (1996), 344–353.
4. Patterson, J., Hill, R. and Rohall, S. Rendezvous: An architecture for synchronous multi-user applications. In *Proceedings of CSCW'90.* (1990), 317–328.
5. Prakash, A., and Shim, H. DistView: Support for building efficient collaborative applications using replicated objects. In *Proceedings of CSCW'94* (1994), 153–164.
6. Roseman, M., and Greenberg, S. TeamRooms: Network places for collaboration. In *Proceedings of CSCW'96* (1996), 325–328.
7. Tolone, W., Kaplan, S., and Jitzpatrick, G. Specifying dynamic support for collaborative work within wOrlds. In *Proceedings of COCS'95* (1995), 55–65.

URLs

Framework for Integrated Synchronous and Asynchronous Collaboration Home Page; www.ncsa.uiuc.edu/SDG/Projects/ISAAC.
NCSA Astronomy Digital Image Library Home Page; imagelib.ncsa.uiuc.edu/imagelib.
NCSA Habanero, (1996). NCSA Habanero Home Page; www.ncsa.uiuc.edu/SDG/Software/Habanero/.
Promondia Home Page, (1996); www4.informatik.uni-erlangen.de/Projects/como/
Sun Microsystems, (1994). The Java language: //An overview; java.sun.com/doc/Overviews/java/java-overview-1.html

Annie Chabert (chabert@ncsa.uiuc.edu) is a research programmer focused on human computer interaction, collaboration, and information technologies in the Habanero project at the Software Development Division, National Center for Supercomputing Applications, at the University of Illinois, Urbana-Champaign.
Ed Grossman (egrossman@ncsa.uiuc.edu) is a Senior Research Programmer on the Habanero project. He is the lead architect of the Habanero framework, and is interested in distributed applications and application frameworks.
Larry S. Jackson (jackson@ncsa.uiuc.edu) is Principal Investigator for the ISAAC project and technical program manager for the Scientific Collaboration System group at the Software Development Division of the National Center for Supercomputing Applications at the University of Illinois, Urbana-Champaign. He is the former technical program manager for the Mosaic project.
Stephen R. Pietrowicz (srp@ncsa.uiuc.edu) is manager of the High Performance Java Visualization group at the National Center for Supercomputing Applications at the University of Illinois, Urbana-Champaign. He is currently responsible for collaborative Web-browser development.
Chris Seguin (seguin@uiuc.edu) is a Ph.D. candidate in the Computer Science department at the University of Illinois at Urbana-Champaign.

This research is supported in part by DARPA grant N66001- 96-C-8511. "Framework for Integrated Synchronous and Asynchronous Collaborative Technologies."

This research is funded in part by the State of Illinois, the University of Illinois, the JavaSoft division of Sun Microsystems, Inc., and EarthWeb, Inc

This material is based upon work supported by the National Science Foundation under Grant No's. NSF ASC 93-15256, NSF ASC 89-02829, NSF ASC 89-02829 REU, and N6601-95-C-8511. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

This project is staffed in part by researchers from INRIA (Institut National de Recherche en Informatique & Automatique), France, from ITI (Information Technology Institute), Singapore, and from the Fraunhofer Institutes, Germany.

Habanero is a registered trademark of the National Center for Supercomputing Applications at the University of Illinois.