Collaborative Environments in Java

Krishnaprasad Thirunarayan
Department of Computer Science and Engineering
Wright State University, Dayton, OH-45435.
Email: tkprasad@cs.wright.edu
Phone: (937)-775-5109
Fax: (937)-775-5133

Abstract— **This article surveys Collaborative Technologies with an eye towards available software tools and environments implemented in Java. The article cites example application areas for collaboration, discusses technical problems to be solved, and then presents some of the existing Java-based tools. It also sketches various dimensions of collaboration in developing large hardware systems.**

Keywords— **Collaborative Technologies, Java.**

I. Introduction

COLLABORATIVE TECHNOLOGIES are broadly defined as technologies that enable collaboration among individuals who may be geographically distributed but are engaged in a common task and share information, possibly working together interactively. Early on, collaborators relied on published materials, audio/videotapes, faxes, and U.S. mail to overcome the distance barrier. Subsequently, electronic collaborative tools enabled collaboration that is both faster and cheaper.

Early electronic collaborative tools were *asynchronous* as exemplified by E-mail, FTP, Listservs and Newsgroups. With substantial improvements in computer technology and networking *synchronous* communication became efficient and affordable. This led to tools for supporting real-time chats, electronic whiteboards, teleconferencing, video conferencing, group decision support systems, etc. The multi-site team communication enabled by these tools helped spontaneous creation of information to share.

Real-time interaction among collaborators obviously requires that they all be on-line at the same time. This may not be feasible or convenient when distributed collaborative teams are in different time zones or find it hard to get a mutually agreeable meeting time. This necessitates the merger of asynchronous and synchronous tools to create and share information.

II. Applications

Collaborative technologies (abbreviated CT in what follows), (or equivalently, computer-supported cooperative work) have varied practical applications. Here we provide a non-exhaustive list of sample applications.

A. Distance Learning

Using CT, a student can take a class from a remote location without actually being on college campus. From student's perspective this adds to convenience and enables access to resources that may not have been available otherwise.

B. Business Applications

CT enables individuals to work on the same project from different sites, or communicate with and assist remote customers. It has potential to improve availability of trained personnel to attend to emergencies, especially to understand, diagnose, and fix non-trivial customer problems quickly. The reduced transportation can lead to savings in money and time.

C. Group Experiments

CTs provide uniform development environment that can help groups to cooperatively design, develop, implement, and analyze systems and products. In particular, it enables building applications for multiple users to experiment with simulations and interactively visualize the shared set of numerical data and computations. This also serves as foundation for multi-point control of complex systems.

D. Computer-Aided Design

As systems get larger and more complex it will be necessary involve teams of experts, who may be geographically separated, for the design, implementation, integration, and testing of the various components. CTs can aid in this process as outlined below.

III. Collaboration in Hardware Systems Development

There are different dimensions to collaboration in the context of Hardware Systems Development.

A. Component-level Collaboration during Execution

Large systems-on-a-chip of the future will be complex and require integration of components of diverse nature such as electronic or mechanical, analog or digital, etc, produced by a variety of differing technologies. To be most effective, these on-chip components must "collaborate" during execution.

B. Module Sharing during Development

To minimize duplication of work and to facilitate sharing of designs and implementation, it is important to have design databases that contain reusable designs and implementations that can be customized to meet the needs of a

specific application. A platform-independent tool, such as a generic synthesis tool, can be built using the available web technology and tools (such as using Java, Tcl/TK plug-ins, and the Web browsers) to maintain repositories of design templates, and enable distributed users to incrementally download and customize relevant parts. Another aspect is to enable users to access to large and diverse set of remote CAD tools using web browsers.

C. Team Collaboration during Design and Maintenance

The design, implementation, analysis, test, and evaluation of large systems in general will require high-level collaboration by teams of people and specialized equipment. In addition, once the systems are deployed, diverse activities such as performance monitoring, fault diagnosis and rectification, upgrade, etc may benefit from collaborative tools that will overcome the distance barrier to enable the appropriate personnel be available for the various tasks. In particular, the collaborative tools and environments discussed below will aid in such activities.

IV. Technical Problems and Potential Remedy

We sketch some of the problems to be tackled when implementing a general tool for collaboration.

A. Portability

The Internet is a network of heterogeneous computers. In order to collaborate over the Internet we need tools that can be integrated with each other and be able to *run* on almost all platforms. The Java programming language provides a solution to this portability requirement. In particular, Java provides a platform-independent software environment in the following two ways:

- *Hardware Independence*: Java compilers translate Java source code into Java byte codes, which is architecture-neutral. This code is then executed by a platform-specific Java interpreter.
- *Operating System Independence*: Java provides APIs for common OS services such as file and network I/O, graphical user interfaces, concurrent programming, etc. These facilities are supported by the underlying Java run-time.

B. Synchronous and Asynchronous Communication

If the collaborators are widely scattered geographically, then both synchronous and asynchronous capabilities must be provided by the tool for effectiveness. The former capability is needed for a real-time dialog among the participants while the latter capability will enable collaborators in a different time zone to contribute their mite. Multiple participants may wish to create and modify existing information and then view them simultaneously.

Java environment provides support for concurrent programming and class libraries for networking, remote method invocation, object serialization, and multi-media presentation.

C. Security

In general, collaboration and Web server security impose conflicting requirements — one pushing for openness and the other for restrictions. The collaborative tools should support access control mechanisms in general and secure access to remote sites in particular.

Java language and its implementation promote safety and security.

D. Other Requirements

A collaborative tool should also support infrastructure for document management (such as replication, versioning and revision control capability on files), task management (to coordinate, manage, and track different collaborators and make them aware of each other), group calendaring and scheduling, etc. Data Management issues and tradeoffs in the context of CSCW are discussed in [7].

V. Sample Java-based collaborative tools

We now summarize some well-known collaborative environments written in Java.

A. Habanero

The following description of NCSA Habanero has been extracted and adapted from the on-line documentation [4].

Habanero is a collaborative framework and environment containing a set of applications. Through Habanero one can create or convert existing Java applications and applets into collaborative applications that enable interaction with other people on the Internet. In general, it enables construction of groupware systems and supports the transformation of single-user software tools into multi-user collaborative tools.

The NCSA Habanero framework provides state and event synchronization for multiple copies of a software tool. It works by replicating applications across clients and then sharing all state changes in those clients. The environment includes a server that hosts sessions and a client that interacts with sessions using a variety applications called Hablets. The server is a process that runs in the background and links Habanero clients during a collaborative session. A session is defined and distinguished by its session name, machine name or IP number and the listening port number. This information is defined by the first client that creates a session. The Habanero client starts in session definition mode : it can either create session or join an on-going session.

Typically, Java applets are single-user applications that are embedded in HTML pages on the Web and run in a Web browser. The most straightforward "habanerization" of an applet would allow distributed multiple users to interact with the same application. For example, in the context of a Whiteboard application, all the users can draw on the whiteboard and see what the other users have drawn as if they were all viewing the same whiteboard. Another example is the voter tool. A vote can be arbitrarily defined by a member of a collaborative session. The initiator

of a vote defines the question, the answer mode, and the vote choices. The initiator also determines where the vote results are displayed. Once the vote is defined, the initiator submits it to the group and each collaborator receives a "vote" window prompting them for a reply. After all ballots are returned or the timer expires, the Arbitrator returns the results.

B. DistView

DistView is a framework for building collaborative applications that supports both synchronous and asynchronous collaboration over the Web [3]. The DistView framework is currently being used as an experimental testbed for wide-area scientific collaboratory work that enables dispersed community of scientists to perform real-time experiments at remote facilities. In particular, it has been used in the context of space physics and aeronomy.

C. VisAd

VisAD is a Java class library for interactive and collaborative visualization and analysis of numerical data [10] [11]. It combines a flexible data model and distributed objects (via Java RMI) to support sharing of data, visualizations and user interfaces between different data sources, different computers and different scientific disciplines. Other examples of Java-based visualization tools are WebWinds from JPL, Horizon from NCSA, etc.

D. Tango

TANGO Interactive is a Java-based Web collaboratory system that has been used for such application domains as education, command and control, health care, and computer steering [9]. Complete solutions have been built for distance learning and training, remote consulting and user support, and programmable 3D virtual environments. For developers, TANGO Interactive also provides a framework to build collaboratory applications.

E. PPP

The PPP system is an example of a design environment for low-power digital circuits that demonstrates the feasibility of Web-based EDA tool integration [6].

VI. Other examples of collaborative tools

A. Communicator

Netscape Communicator consists of an integrated suite of tools that helps users communicate, access, and share information on the Internet [2]. In particular, Netscape Collabra software allows users to collaborate with co-workers by setting up and managing discussion groups. Netscape Calendar software provides facilities for scheduling meetings, managing to-do lists, etc.

B. Netmeeting

Microsoft's NetMeeting supports "face to face" conversations and collaboration with friends and co-workers over the Internet. It supports audio and video conferencing, data and program sharing, text chat, whiteboard, and file transfer, for all Windows users.

C. ClassWise

This product supports distance education in real-time by delivering live or recorded course with real-time audio, allowing slide presentations and annotations, text chat (as a backup for network conjestion), etc [1]. It allows interactive or self-paced learning over the Internet, and supports recording synchronous sessions for later playback.

D. TopClass

TopClass is a Web-based training tool that enables creation of course materials and tests for distance learning, and management of interactions between instructor and students [8].

VII. Conclusions

In this paper, we identified exisiting Collaborative Tools and Environments implemented in Java that can potentially help groups to cooperatively and interactively design, implement, integrate, analyze, monitor, maintain, and diagnose systems.

References

[1] ClassWise http://www.magideas.com/
[2] Netscape Communicator Professional. http://home.netscape.com/communicator/v4.0/professional.html
[3] The DistView Collaboratory Toolkit http://www.eecs.umich.edu/distview/
[4] NCSA Habanero http://havefun.ncsa.uiuc.edu/habanero/
[5] Microsoft Netmeeting. http://www.microsoft.com/windows/netmeeting/
[6] de Micheli g., et al, PPP: A Web-Based CAD Environment http://akebono.stanford.edu/users/nanni/research/www/www_ppp/ppp.html
[7] Prakash, H.S. Shim, and J.H. Lee, Issues and Tradeoffs in CSCW Systems, IEEE Transactions on Data and Knowledge Engineering (to appear).
[8] TopClass http://www.wbtsystems.com/
[9] Tango http://www.npac.syr.edu/projects/tango/index.html/
[10] Visualization and Analysis of Data http://www.ssec.wisc.edu/ billh/visad.html
[11] W. Hibbard, VisAD: Connecting people to computations and people to people Computer Graphics 32, No. 3, 1998, 10-12.