Frame-buffer on Demand: Applications of Stateless Client Systems in Web-based Learning

Sheng Feng Li
Laboratory for Communications Engineering,
Department of Engineering, University of Cambridge,
Cambridge CB2 1PZ, UK

Quentin Stafford-Fraser
And
Andy Hopper
AT&T Laboratories Cambridge,
Cambridge CB2 1QA, UK

ABSTRACT

The growth of the Internet and the World Wide Web has changed the way people are educated, and distance learning is amongst the most promising fields of new Web applications built from existing services with supporting infrastructures. In this paper, we introduce a number of new applications built from Stateless Client Systems to assist the learning of computer-based activities. Stateless Client Systems separate the display interface from the application logic in windowing systems. They embed a client/server architecture, where the server executes all the applications and the client simply presents the frame buffers or screen images to the user and accepts user input. We seamlessly introduce a proxy server into the client/server architecture and create tools that enable teachers and students to record, replay, operate and share computer-based work sessions. The concepts and the systems for distance learning can be easily extended to generic collaborative work.

Keywords: Stateless Client System, Virtual Network Computing, World Wide Web, distance learning.

1. INTRODUCTION

In the article titled "As We May Think", Vannevar Bush proposed a hypothetical device called a *memex* [6]. The device would contain a large number of records, all of which cross-linked with one another, so that they may be consulted with exceeding speed and flexibility. Over the years, there were sporadic attempts to implement this concept, with examples including Brown University's Intermedia System [25] and the Knowledge Management System developed by the researchers from Carnegie-Mellon University [3]. Built on the Internet, the World Wide Web and the Hypertext Markup Language (HTML) finally brought to life the memex of Dr. Bush's imagination [19]. The critical element of the memex is the hyperlink, as human brain operates by association.

The growth of the Internet and the World Wide Web has changed the way people are educated, and the Web-based training is warmly welcomed because of the economic and the educational benefits it brings us. It reduces travel cost and provides just-in-time instructions [8]. People who are already familiar with the computer and the Web technology quickly adopted this type of on-the-job training, with which they can demand for help whenever they need and perform their tasks on their PCs or workstations at their own time and pace. Text-based instructions usually cannot provide them with sufficient visual cues of exactly what to do, and this is when multi-media technologies such as the video technology come to assist. However, the technical limitations of video taping computer screens make the screens very difficult to read [11], so we turn to the so-called Stateless Client Systems for help.

Stateless Client Systems separate the display interface from the application logic in windowing systems. They embed a client/server architecture, where the server executes all the applications and the client simply presents the frame buffers or screen images to the user and accepts user input. Since the entire state is preserved in the server, the client becomes stateless. During their operations, changes to the frame buffer are sent from the server to the client in the form of pixel values, and key strokes and mouse clicks are sent from the client to the server in terms of events. With our logging facility, the messages flowing between the client and the server can be recorded and replayed [13]. The resulting movie preserves the platform independent property of a video playback, while at the same time, it achieves a higher quality in displaying sharper screen characters. These recordings can be distributed, edited, annotated and cross-linked with one another, and users can browse through and search for the recorded work sessions in the same way they do with Web pages.

In the field of distance learning, the majority of the systems are still of the presentation types, and the conferencing types are just beginning to be used [9]. To provide the full benefit of Web-based learning, a system must be interactive between teachers and students, and among students themselves. Not only can our recorded work sessions be cross-linked with one another, but also they can be cross-linked with real-time work sessions. If multiple users log into the same real-time work session via the hyperlinks, they can interactively share workspace and applications within that session.

Our applications of Stateless Client Systems should only be regarded as tools that assist IT-related training where students learn about computer-based activities. They don't offer a complete suite that supports distance learning in every aspect.

However, these tools can be integrated into a distance learning environment such as the Virtual Design Studios [21] or the Interactive Remote Instruction system [4], and work in conjunction with other tools such as shared whiteboard, video conferencing, bulletin board, on-line chat and email.

In Section 2, we describe a Stateless Client System, namely the VNC (Virtual Network Computing) [20] developed at the AT&T Laboratories Cambridge. Section 3 illustrates the applications of VNC in Web-based learning, Section 4 gives the performance results of these applications, and Section 5 compares our approach with other related work.

2. VIRTUAL NETWORK COMPUTING

In a client/server architecture, the functionality of the overall system is split between the server and the client, which are connected through appropriate communication channels. Whether the client is "thin" or "thick" is related to the exact division of this functionality. Thin clients require more communication and interaction with the server, but they are simpler than thick clients and can be quickly downloaded from the Internet. There are compelling reasons for retaining a thin-client architecture, as it removes the requirement for specially adapted hardware and software, and preserves the client's full functionality [10]. The VNC system pushes this trend to an extreme by moving the execution of applications completely to the server, leaving the client stateless. When the applications are executed at one site, their frame buffers or screen images can be transferred across the network to be displayed at a remote site. Without changing any windowing system, VNC breaks it into client/server pieces, namely the VNC server, the VNC client and the VNC protocol that connects the previous two (Figure 1). The VNC server executes all the applications

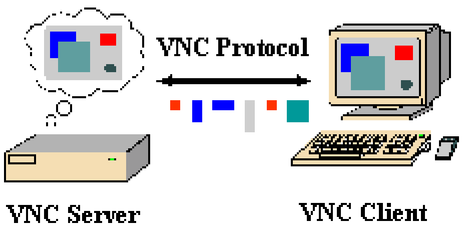

Figure 1. Virtual Network Computing

and generates the frame buffer. The VNC client displays these frame buffers and accepts user input. And the VNC protocol defines the mechanism that transfers the changes to the frame buffer in units of rectangles containing pixel data from the server to the client, and the user events such as keystrokes and mouse clicks from the client to the server. We term the former *frame buffer update* messages and the latter *user event* messages.

3. APPLICATIONS OF VNC

We extend the basic VNC system by seamlessly introducing a proxy server into its client/server architecture (Figure 2). The proxy server can multicast the *frame buffer update* messages from one VNC server to multiple VNC clients, and merges the *user event* messages from multiple clients to the server. While at the same time, it stores these messages to the log file. This extension results in a number of new applications as follows:

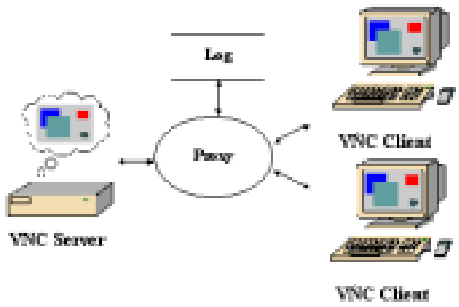

Figure 2. Extending VNC with proxy

Recording and replaying

Our applications of VNC start with recording and replaying work sessions [13]. The recording of a work session is performed by the proxy server shown in Figure 2. This proxy can intercept, store and forward messages between VNC client/server without changing either of them. To facilitate future replaying, the *frame buffer update* messages are time-stamped and saved to a log file. The recorded medium is a stream of rectangles containing pixel data of the screen images. Each rectangle represents a rectangular area of the frame buffer. Together with other rectangles in the same message, they cover the frame buffer area which has been changed since the last update. The replaying of work sessions is performed in a tool called the Reviewer (Figure 3). The Reviewer provides the user with VCR-like control (i.e. fast forward, rewind, play and stop) of the recorded session. Like the VNC client (also known as the Viewer), the Reviewer just displays the *frame buffer update* messages received, except that these messages now come from the log file rather than the VNC server.

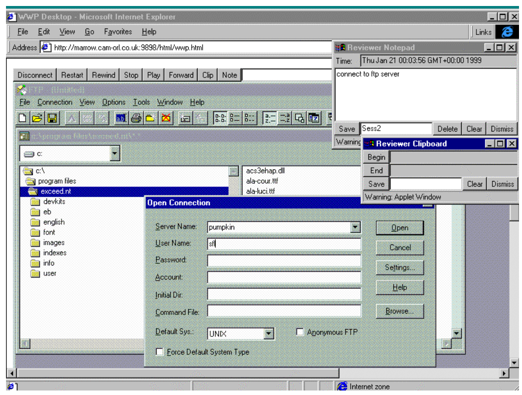

Figure 3. Reviewing Windows/NT session

Editing and annotating

The recorded work sessions can be replicated, edited, annotated, and published on the World Wide Web. The editing and annotating panels are shown at the top-right corner of Figure 3. We provide two ways of editing a recorded session. Users can either cut a long session into shorter ones or combine several short sessions to form a long one. The recorded sessions can contribute to the knowledge resources carried by what Tanaka referred to as the *meme* media [23]. The *meme* media are required to carry any kinds of knowledge resources including documents and tools. Such media can replicate themselves, to recombine themselves, and to be naturally selected by end users who are interested in the publication, the exchange and the reuse of knowledge. Annotation is a primary way of collaboration. Figure 3 shows a push button labelled “Note” at the top of the Reviewer applet. When this button is pushed, the annotating panel will appear, inviting users to take a note about the activity. The annotations are also timestamped and saved, and these timestamps serve as synchronisation points. During the playback, when the timestamp of the annotation matches that of the *frame buffer update* message displayed, the annotation will pop up for a specified period of time. Users can delete or modify previous annotations as well as adding new ones. The annotations, together with the hyperlink that leads to the launching of the corresponding work session, are automatically collected and embedded in the HTML page that publishes the work session.

Browsing and searching

Browsing is a canonical activity for hypermedia users. The recorded work sessions can be grouped into folders on the local host or distributed among hosts world wide, and together they form a repository. The primary way of aggregating these sessions is to embed their hyperlinks in their text annotations. The text annotations form HTML pages that can be cross-linked with other local or remote Web pages containing hyperlinks to more recorded or real-time work sessions. For example, the embedded hyperlink to the recorded session of Task1 (Figure 4) is specified by the URL (Uniform Resource Locator): *http://pumpkin.uk.research.att.com:9800/Task1.html*, where “Task1.html” contains the name, the location and the parameters of the Reviewer applet, and one of the parameters specifies the name of the log file. Once the user clicks “Start

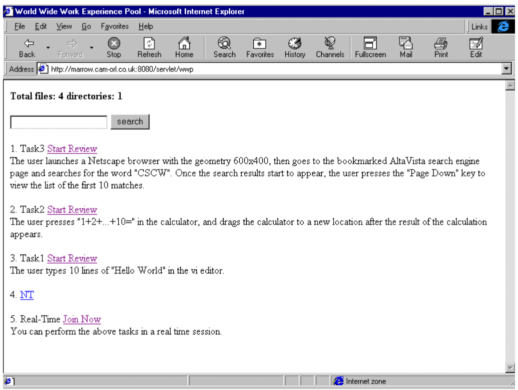

Figure 4. Browsing & searching sessions

Review” (Figure 4), the Reviewer applet will be downloaded followed by the recorded messages from the named log file. She can then control the playback of the recorded session as shown in Figure 3. In a large hypermedia repository with thousands of sessions, users can have trouble locating information solely by browsing, either because the structure of the repository is not well suited to their search, or because they have forgotten where they stored the sessions. Armed with the Cobra search engine [16], we index the annotated sessions and enable the users to search for them by keywords, phrases and more complicated queries.

Operating and sharing

Similarly, the hyperlinks to real-time work sessions can also be embedded in the text annotations. For example, the embedded hyperlink to the real-time session in Figure 4 is specified by the URL: *http://marrow.uk.research.att.com:9600/viewer.html*, where “viewer.html” contains the name, the location and the parameters of the Viewer applet, i.e. the VNC client. Once the user clicks “Join Now” (Figure 4), the Viewer applet will be downloaded and connected to the VNC server running on “marrow”. She can then remotely operate the real-time session. Because of the platform independence of the VNC protocol, the VNC server can run on Windows PCs, Unix workstations and Macintosh desktops, and the VNC client can run on any platform with a Java-enabled Web browser. That is to say, from one host, a user can remotely operate any windowing system and learn about software applications that are not even installed or supported in her local host. The sharing of work sessions is enabled by the same proxy server that performs recording [12] as shown in Figure 2. The proxy also provides floor control and user awareness support. Instead of connecting to the VNC server directly, the downloaded Viewer applet, together with other collaborative Viewers, now connect to the proxy. Multiple users can then share the view and the control of the session simultaneously (Figure 5).

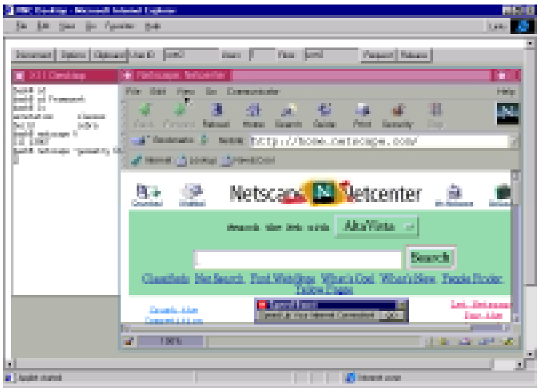

Figure 5. Sharing Unix work session

4. PERFORMANCE

Since the basic VNC system was released in February 1998[1], more than 200,000 users have downloaded and evaluated it on their Unix workstations, Windows PCs and Macintosh desktops respectively. They use VNC for remote viewing and control of heterogeneous windowing systems from their local hosts in presence mobility. The feedback from them has been very encouraging.

We extend the basic VNC system and create a couple of new applications, namely the Reviewer through which users can replay and review previously recorded computer sessions, and the Viewer through which they can operate and share real-time computer sessions. Both the Reviewer and the Viewer applets are very thin: 34k bytes for the Reviewer and 48k bytes for the Viewer in their corresponding Java class files. These applets take 5 to 10 seconds to download from the Web server in our Local Area Network (LAN). Once they are downloaded to the Web client, they’ll take residence in the client’s memory.

[1]VNC is available for downloading at *http://www.uk.research.att.com/vnc/*

We make use of our system to prepare visual aides to go with text-based on-line help files. Our main concern is the storage and the bandwidth required by our tools. Here we present two typical examples of tutorial sessions we have created and measure the storage and bandwidth consumption of these tools.

- Session 1 takes place on a Unix/X-Window workstation, where the teacher demonstrates how to send an email through *exmh* [18]. The email has "exmh test message" as the subject and "Hello World" as the body. The demonstration follows closely to the 6-step instructions given by the on-line help file available from *http://www.beedub.com/exmh/exmh.n.html#SENDING MAIL*. To prepare the training material, the teacher has the VNC server running on a Unix workstation named "pumpkin", the proxy server running on a second Unix workstation named "marrow", and operates the remote X Window session through the Viewer applet in the Web browser on his PC. The scrollable viewing area is 796 pixels in width and 576 pixels in height. The hosts are connected by 100Mbps Ethernet links. Table 1 shows that session 1 takes 34 seconds to complete, requires a data transfer rate of 0.53 Mbytes per second (4.24Mbps) at peak, and results in a log file of 3.10 Mbytes. The statistics are obtained by taking the average of three consecutive trials.

- Session 2 takes place on a Window NT workstation, where the teacher demonstrates how to transfer a file using *ftp* from Hummingbird's Exceed package. The size of the file is 2Kbytes. The demonstration includes "connecting to an FTP server", "listing the contents of a directory", "setting the file transfer type", "dragging and dropping files" and "disconnecting from an FTP server", specified by Hummingbird's on-line help file. To prepare the training material, the teacher has the VNC server running as a service on an NT workstation named "grapefruit", the proxy server running on "marrow", and operates the remote NT session through the Viewer applet in the Web browser on his PC. The scrollable viewing area is 1152 pixels in width and 882 pixels in height. The hosts are connected by 100Mbps Ethernet links. Table 1 shows that session 2 takes 51 seconds to complete, requires a data transfer rate of 0.54Mbytes per second (4.32Mbps) at peak, and results in a log file of 4.20 Mbytes. The statistics are obtained by taking the average of three consecutive trials.

	Session 1	Session 2
Time(S)	34	51
Peak Rate(MB/S)	0.53	0.54
Storage Space(MB)	3.10	4.20

Table 1. Performance statistics

Now we have an idea about the storage and the bandwidth consumption, the question becomes "can we provide sufficient resources to support our applications?" The emerging DVD (Digital Versatile Disc) technology is designed to meet the ever-increasing storage capacity demands of the computer industry [7]. With DVD-R (write once, read many) providing 3.9 GB and DVD-RAM (rewritable) featuring 2.6 GB and 5.2 GB (double sided), these discs can each hold approximately 10 hours of recorded training sessions, and hence have enough capacity to meet the storage requirement of our applications. Similarly, network bandwidth is becoming more plentiful. Members of our laboratory have been using VNC to perform daily tasks for years, in our 100Mbps LAN. Companies such as Acorn have been planing video-on-demand for home, which requires a data transfer rate in the same order as that required by our applications. Therefore we are convinced that, in the near future, when multimedia applications such as video-on-demand gain their popularity, our applications of *frame-buffer-on-demand* will naturally follow.

5. RELATED WORK

Our applications of VNC have added a set of new tools to the distance learning environment. These tools enable teachers and students to record, replay, operate and share computer-based work sessions. Unlike the Classroom 2000 project [2] that supports multiple teaching and learning styles, our tools only focus on the support for the teaching and learning of computer-based activities. However, they can be integrated with other tools such as electronic whiteboard and video conferencing to offer wider support for more styles. In this section, we compare our applications with similar tools already available in distance learning.

The Classroom 2000 project developed at the Georgia Institute of Technology tests the hypothesis that tools to aid in the capture and subsequent access of classroom information will enhance both the teaching and the learning experience [2]. Advancements in video technologies have simplified the capture process to the point where the creation of large collections of video is commonplace. Many tools use video as a powerful medium to record and replay various types of activities including lectures and meetings, with support for annotations and retrieval [24, 2, 17]. However, video recording computer-based activities makes the computer screens very difficult to read during playback [11]. In our applications, we record changes to the frame buffer during a work session without extra devices such as cameras or video capture cards, and the resulting playback is as clear and sharp as the original session. We believe that screen capture of computer-based activities is a better alternative than video capture, although the latter is frequently used in the recording of general activities. Lotus ScreenCam [14] and Microsoft Camcorder share our view by using dynamic screen captures to create movies for training and system demonstration. While both ScreenCam and Camcorder are optimised for Windows PC, they loose the platform independent property of a video playback. Our tools preserve this property, and users can playback Unix sessions on Windows PCs and Windows sessions on Unix workstations.

Event-based recording and replaying [22] proposes a completely different approach to video and screen capture and playback, in such a way that events carry semantics while video and screen images don't. Let's take the *draw line* activity as an example: event-based messages will specify the starting point, the end point and the colour of the line while our *frame buffer update* message specifies the position, the size and the pixel values of the rectangle which contains the line. Now let's take the *draw circle* activity as another example: event-based messages will specify the centre, the radius and the colour of the circle, while our *frame buffer update* message still specifies the position, the size and the pixel values of a rectangle, but here the rectangle contains the circle. Event-based messages have a much smaller size, so they require less bandwidth to transmit and less space to store. On the other hand, these messages require much more processing, for example, X Window events require the display platform to run a program called the X server to process them. It is difficult to undo processed events, and as a result, rewinding a recorded session during playback is particularly difficult to support. However, the ability to rewind is very important to students who may fail to grasp important issues during the first-time review. The second constraint of event-based messages is that they tend to be system specific, i.e. events from different systems require different programs to understand and process them, e.g. the X server which processes X Window events can not be used to process Windows events. Our *frame buffer update* messages are platform independent, and the same Reviewer tool can replay X Window, Windows and Macintosh sessions on any platform with a Java-enabled Web browser.

Our applications differ from other shared window systems in the way that we have integrated both synchronous and asynchronous sharing schemes. We have compared our Reviewer tool with currently available tools that support recording and replaying. This covers the presentation type (i.e. the asynchronous mode) of distance learning. In the following paragraphs, we compare our Viewer tool with other tools that support remote operations and interactions to cover the conferencing type (i.e. the synchronous mode).

The Interactive Remote Instruction (IRI) system [4] creates a distributed virtual classroom within which teachers and students have effective tools for preparing, presenting and assimilating class material and for interacting with one another even when they are geographically dispersed. It contains a Tool Sharing Engine based on the X Teleconferencing and Viewing (XTV) system [1]. The Engine allows sharing of all X Window applications. At any given moment, only one participant can provide input to the applications and the output is distributed to all participants and displayed locally through the X server on their workstations. The X server can be shrunk to a size so small that it can easily be incorporated into web pages and downloaded on demand to Java enabled browsers[2]. This means XTV users can now operate and share X Window applications on the Web. The reliance of XTV on the X protocol is conceptually at the same level as the reliance of our system on the VNC protocol [12]. What makes our system fundamentally different is that X protocol is system specific while the VNC protocol is not tied to any operating system or windowing system. Many collaborative applications make use of system specific events to create shared components [5], and as a result, they loose the platform independence property. VNC and NetMeeting [15] have broken this event-based convention and make the sharing take place at the frame buffer level. NetMeeting is built on the T.128 protocol, formerly known as the T.SHARE application sharing protocol. Although T.128 itself is platform independent like the VNC protocol, NetMeeting only supports the sharing of Windows applications. Our system follows the principle of *download once, share everything*. In a network with sufficient bandwidth, with the same Viewer tool, users can operate and share X Window, Windows and Macintosh sessions on any platform with a Java-enabled Web browser.

[2] In August 1997, GraphOn Corporation announced the availability of GO-Joe, the thin client X server for all Java desktops.

6. CONCLUSION

In this paper, we combined two enabling technologies, namely the Stateless Client System VNC and the WWW (World Wide Web). The VNC generates a new medium, i.e. the stream of rectangles containing pixel data of the frame buffers, and the WWW provides an ideal environment to present such a medium. The platform independent nature of VNC and the ubiquitous nature of WWW together result in a number of applications that can assist distance learning of computer-based activities synchronously and asynchronously, in a demand-led rather than a product-driven style. We compared our Stateless Client System with event-based systems and discussed their advantages and disadvantages. We traded off storage and bandwidth for heterogeneity.

In our future work, we will extend the VNC system one step further to include some event-based features, mainly because events can serve as indexing points into activities [5]. Without giving up platform independence, we'll store user events such as keystrokes and mouse clicks to automatically index a recorded VNC session. Hence, users will be able to not only search for work sessions, but also search within work sessions.

7. ACKOWLEDGEMENT

Sheng Feng Li is a Ph.D. student sponsored by EPSRC and AT&T, Quentin Stafford-Fraser is a research engineer at AT&T, and Andy Hopper is the professor of Communications Engineering and the director of AT&T Laboratories Cambridge. We would like to thank Tim Mills and Tristan Richardson from AT&T for supporting the project, and Antony Rowstron and John Bates from the Laboratory for Communications Engineering for their valuable comments.

9. REFERENCES

[1] Abdel-Wahab, H. and Feit, M. (1991): "XTV: A Framework for Sharing X Window Clients in Remote Synchronous Collaboration", *IEEE Conference on Communication Software: Communication for Distributed Applications & Systems*, North Carolina, April, 1991

[2] Abowd, G., Atkeson, C., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N. and Tani, M. (1996): "Teaching and Learning as Multimedia Authoring: The Classroom 2000 Project", *ACM Multimedia'96*, Boston, MA, November 18-22, 1996.

[3] Akscyn, R., McCracken, D. and Yoder, E. (1988): "KMS: A Distributed Hypermedia System For Managing Knowledge in Organizations", *Communications of the ACM*, Vol. 31 No. 7, July 1988

[4] Al-Shaer, E., Youssef, A., Abdel-Wahab, H., Maly, K. and Overstreet, C. (1997): "Reliability, Scalability and Robustness Issues in IRI", *IEEE Sixth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'97)*, Cambridge, Massachusetts, June 18-20, 1997

[5] Bates, J., Spiteri, M., Halls, D. and Bacon, J. (1998): "Integrating Real-World and Computer-Supported Collaboration in the Presence of Mobility", *IEEE Seventh*

International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'98), IEEE Computer Society Press, June 17-19, 1998.

[6] Bush, V. (1945): "As We May Think", *The Atlantic Monthly*, July 1945

[7] D'Ambrise, R. (1997): "A Closer Look at DVD", Maxell Corporation of America, http://www.cd-info.com/CDIC/Technology/DVD/dvd.html, 1997

[8] Driscoll, M. and Thomson, R. (1997): "The Web as a Learning Environment", *IEEE Sixth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'97)*, Cambridge, Massachusetts, June 18-20, 1997

[9] Fukuda, S. (1997): "Working Group Report on Distance Learning", *IEEE Sixth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'97)*, Cambridge, Massachusetts, June 18-20, 1997

[10] Hild, S. and Robinson, P. (1997): "Mobilizing Applications", *IEEE Personal Communications*, October 1997

[11] Ishii, H. and Miyake, N. (1991): "Toward An Open Shared Workspace: Computer and Video Fusion approach of TeamWorkStation", *Communications of the ACM*, vol. 34, no.12, December 1991

[12] Li, S. and Hopper, A. (1998a): "A Framework to Integrate Synchronous and Asynchronous Collaboration", *IEEE Seventh International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'98)*, IEEE Computer Society Press, June 17-19, 1998.

[13] Li, S. and Hopper, A. (1998b): "What You See Is What I Saw: Applications of Stateless Client Systems in Asynchronous CSCW", *The Fourth International Conference on Computer Science and Informatics (CS&I'98)*, Research Triangle Park, North Carolina, Oct. 23-28, 1998.

[14] Lotus (1997): "Lotus ScreenCam", Lotus Development Corporation, http://www.lotus.com/products/ screencam.nsf, July 1997

[15] Microsoft (1997): NetMeeting 2.0 Reviewers Guide, http:// www.microsoft.com/netmeeting/, June 1997

[16] Mills, T., Moody, K. and Rodden, K. (1997): "Cobra: A new approach to IR system design", Proceedings of RIAO'97, July 1997, also available from http://www.orl.co.uk/~tjm/ papers/index.html

[17] Minneman, S., Harrison, S., Janssen, B., Kurtenbach, G., Moran, T., Smith, I. and Melle, B. (1995): "A Confederation of Tools for Capturing and Accessing Collaborative Activity", *ACM Multimedia'95*, San Francisco, CA, November 1995.

[18] Peek, J. (1995): *MH & xmh: Email for Users & Programmers*, Third Edition, ISBN 1-56592-093-7, O'Reilly & Associates, Inc. 1995

[19] Powers, S. et al. (1998): *Dynamic Web Publishing*, Second Edition, ISBN 0-57521-363-x, Sams.net Publishing, 1998

[20] Richardson, T., Stafford-Fraser, Q., Wood, K. and Hopper, A. (1998): "Virtual Network Computing", *IEEE Internet Computing*, Vol. 2 No. 1, January/February 1998

[21] Simoff, S. and Maher, M. (1997): "Web Mediated Design Courses: Challenges and Realities in Teaching Electronic Collaboration", *IEEE Sixth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'97)*, Cambridge, Massachusetts, June 18-20, 1997

[22] Spiteri, M. and Bates, J. (1998): "An architecture to support storage and retrieval of events", *Proceedings of MIDDLEWARE 1998*, IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing, Lancaster, UK, September, 1998

[23] Tanaka, Y. (1996): "Meme Media and a World-Wide Meme Pool", *ACM Multimedia'96*, Boston MA, USA, November 18-22, 1996

[24] Weber, K. and Poon, A. (1994): "Marquee: A Tool For Real-Time Video Logging", *Proceeding of ACM CHI'94*, pages 58-64, April 1994.

[25] Yankelovich, N., Haan, B., Meyrowitz, N. and Drucker, S. (1988) "Intermedia: The Concept and the Construction of a Seamless Information Environment", *IEEE Computer*, January 1988