Transparent Computer Shared Cooperative Workspace (T-CSCW) Architectural Specification

John C. Checco

Abstract:
The purpose of this paper is to define the architecural specifications for creating the Transparent CSCW model as a consumer based product.

Rev #	Date	Description
0.10	01-10-95	Initial Draft
0.20	01-25-95	Research References
0.30	02-01-95	Multi-Point Architectures
0.40	03-03-95	Details about Connectivity
0.50	04-10-95	Prototype Overview, Logic Flow, Acceptance Requirements, and I/O Control Commands
1.01	05-30-95	Proposed Possible Camera Video as Grayscale
2.01	04-20-96	Replaced dependency upon ISDN with suggestions for other technologies (HDSL).
3.00	08-13-96	Split document into Overview, Architecure, Functional, and Prototype specifications.

Table of Contents

Transparent Computer Shared Cooperative Workspace (T-CSCW) Architectural Specification

John C. Checco
John_Checco@compuserve.com

1 Introduction

The decision of which communication model to implement needs to be reconsidered in parallel with the varying types of collaboration. In essence, the control projection must be clearly aligned with the content projection mechanism. For example, if a collaborative session uses computers to promote social interaction, the content is both visual and aural. Ideally, the control and context should also be visual/aural. The importance of video then far outweighs the necessity of a digitally-shared workspace. On the contrary, if the content being projected is an object of computer-based application, control projection should include the use of that computer application as part of the collaboration. With any method of collaboration, the paradigm of same-time different-place must be applied to the shared workspace.

2 Current Architectures

Systems such as Intel's ProShare®, AT&T Vistium®, and PictureTel's PCS-50® use ISDN for transporting video talking heads and simulating application sharing. In all these examples, application sharing consists of binary data that allows a client to "see" the host application on their computer desktop. But in essence, remote users are only seeing a bitmapped representation of the host application. The host's computer generates these bitmaps and the client's computer displays this bitmap. Some packages allow the bandwidth ratio to change on demand; however, this usually results in loss of video frames, application refresh, or audio depending on the specific application priorities.

Traditional groupware consists of a video link for a talking head, and a specialized application for sharing documents or graphics. Additionally, a second video source can be used for displaying documents or objects. Newer systems break the barrier of traditional groupware by allowing multiple users to collaborate using any software application. Most implementations also provide the ability to exchange data files between users. The use of ISDN provides greater bandwidth which can be used for speedier and more complex transmissions than either POTS or direct LAN connections. Also, ISDN is widely available in most regions of the country.

Current implementations view the presence of video "talking heads" as important as the shared workspace from a bandwidth perspective (up to 80% can be used for video transmission). The

real value of video is the detail conveyed through gestures and gaze. Smaller video images may drastically decrease the sense of telepresence [Prussog, Muhlbach and Bocker 94]. Yet these implementations present the video images in a space too small for any gesture interpretation. In addition, the desktop real-estate must still be split between the user image and shared workspace. This occlusion may prove to be more distracting than no video window at all.

Because of the limit of transmitting data over POTS lines, optimizations were made with these systems to capture the shared application window into bitmaps at specified intervals. These full window bitmaps were then compared to the last instance of the window, and a differential bitmap is generated (much smaller than a full window bitmap). This data is sent to the client workstation, where it is combined with the client's last full window bitmap, and then displayed on the client desktop. With this existing architecture, it is still the host and client computers doing the bulk of the preparation. The new bottleneck exists in the computers' abilities to process the bitmapped *picture* of the application "instantaneously". The methods that were optimized to make use of slow line speeds now become the bottleneck under ISDN.

Finally, the processing overhead associated with current implementations restricts desktops from using groupware to share resource intensive applications. How effective is the communication between a remote computer which refreshes its screen through bitmaps (approximately 0.5 fps) and a host computer running a CAD, 3D rendering, or video editing package? Would these packages even be able to run, given the resources needed to run the groupware itself?

3 Proposed Architecture

An effective solution must not only provide telecommuters with a seamless sense of place but also a seamless integration of a shared desktop. The proposition that is being constructed here is two-fold: use the bulk of the bandwidth and the speed of independent hardware-based video transmission to enhance the sense of a shared workspace, and replace the talking heads model with a full-screen overlayed image of the remote person to promote the effective use of gestures [Heath and Luff 91] and possibly gaze awareness [Ishii, Kobayashi and Grudin 92].

Note: The transport mechanism throughout this proposal focuses on HDSL; however, the CSCW box should have an internal architecture that eases the redesign for other broadband transmission protocols.

3.1 Video Transport

In comparison to conventional solutions, this solution should be compatible with other hardware for video-conferencing. It should allow the use of H.320 and H.261 standards for transporting audio and video. However, if both ends of the connection use this hardware, custom compression should be used to achieve compression rates of at least 25 frames per second of 640x480 resolution 64-level grayscale video.

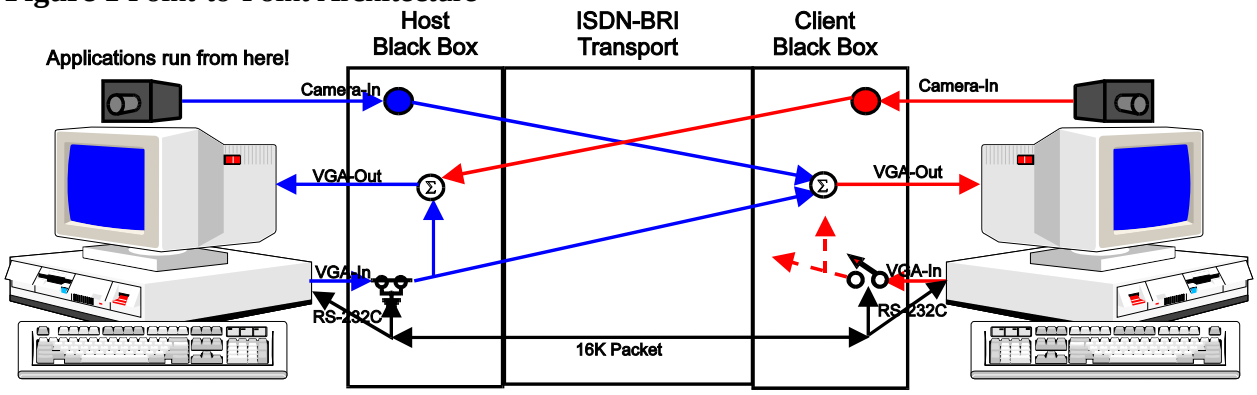
3.2 Desktop Sharing (Transport)

It is the method by which desktop sharing is accomplished which defines the unique content and control projections that make this proposed solution novel compared to other current solutions. Conventional methods use *statically defined* channels to send and receive audio, video and computer input. Most solutions use the video-conferencing standards (such as H.320 for video) to transport these channels. *However, the proposed solution requires an additional channel that may act as either a video input stream or a video output stream to transport the host desktop image to be shared.* Since this additional channel is transporting non-standard (high resolution) video, custom compression must achieve at least 5 frames per second of a maximum 1280x1024 resolution 256-level color video.

3.3 Point-to-Point Architectures

Focusing on a single point-to-point session, there are two stages of communication. The first allows social interaction by only transporting each user's video to the remote user. This model does not allow any application sharing at all. Each user would see the other superimposed upon their own desktop (so they may converse without disturbing any work a user may be doing). The second stage is known as a desktop sharing mode, whereby one user must be designated as the host -- providing the desktop to be shared -- and the other serves as a client. From that point on, each user would be looking at a single desktop with the other user's video superimposed on it. For the desktop itself, there would be no visual differences. Both host and client would be able to view and manipulate the desktop in the same manner.

Figure 1 Point-to-Point Architecture



A simple diagram of a possible architecture is shown in **Figure 1**. The simplicity of this implementation is apparent because the architecture is not computer intensive. Therefore the

only platform dependent piece is the *control projection* software managing the computer input from both users.

3.3.1 Hardware Requirements:

Each workstation would be equipped with the equivalent of a single SVGA-based video mixer. The connection of the two workstations is provided by an HDSL interface that would allow the mixed video signal as well as a small bandwidth of binary data (generated from the *control projection* software) to be sent and received. With current compression technologies and bandwidth requirements, sending 2-way camera video and desktop (as 1-way video) would require more than a single ISDN-BRI line at 112K (56K+56K). Thus, using HDSL as a transport medium allows a more realistic sense of the desktop and overlayed image. This black box would use an OEM HDSL product such as those modems available from PairGain, BrookTree, or AT&T Paradyne as its communication transport device.

In client mode, this hardware would take the NTSC camera signal and pipe it throught the remote connection to the host. It would also receive both an NTSC camera signal as well as an SVGA signal from the host, mix it according to the client user's preferences, and display it directly to the client's SVGA monitor.

In host mode, this hardware would take as input the NTSC camera signal sent from the client site, mix it with the its own SVGA signal according to the host user's preferences, and display it directly to the host's SVGA monitor.

3.3.2 Software Requirements:

A process daemon would be present to allow the use communicate with the CSCW black box device. Ideally, it would be able to manipulate the device setup, download a saved device setup, retrieve and save a device setup.

In desktop sharing mode, it would negotiate computer input traffic as well as video signal switching between the host and client. Unlike other groupware applications, no processing of desktops or bitmaps would occur.

3.3.3 Advantages:

By transporting the desktop as externally manipulated video, the computers' overhead for capturing the desktop as a bitmap, doing a binary differential comparison, transporting the differential bitmap as data, and having the receiving computer decode and display this differential bitmap is no longer incurred. This leaves more timeslices, memory, and processing power to the user's tunning applications. A screen refresh rate of 10 fps would be an *improvement of over 50 times* the refresh rate of current software-only solutions.

By allowing up to a full-screen interpolated overlay of the connected person, gesture interpretation is enhanced, gaze awareness may be communicated, and the shared workspace is no longer occluded. Research has demonstrated that this method of video overlay onto a workspace does not interfere with the perception of workspace [Ishii, Kobayashi and Grudin 92]. It is also known the larger video images may drastically improve the sense of telepresence [Prussog, Muhlbach and Bocker 94] and may also improve the value of gesture [Heath and Luff 91], which may, in turn, help users direct attention appropriately.

It is important to note that the size, position and fade level of each image (workspace and video) should be controlled by each user. This allows a user to explicitly set the level of overlay that causes the least distraction. Each user may then find their own level of comfort with this display system.

An entire host desktop is displayed at all locations without the requirement that all locations contain the applications nor the data being shared. Although this is true for most implementations, current bitmap implementations do not allow the transfer of desktop video outside of the machines internal video memory. Applications that use external video overlays (such as video editing systems) *could not be shared under the current architectures*.

In addition to offloading screen sharing from the confines of the desktop CPU, providing a small software-based input translator for each platform, collaborations can take place over heterogenous desktops. Imagine the power of being able to collaborate on a peer's workspace without having the same platform as that peer! This is possible with current implementations, but has yet to be demonstrated. Also the sociological implications of heterogenous collaboration could be explored.

3.3.4 Disadvantages:

Many of the video compression components needed to implement such a system are not available. Video mixers and digitizers are designed to handle broadcast standards such as NTSC and PAL, not custom resolutions. Also H.320 compression only specifies resolutions of 160x120 at 30 fps (or 320x240 at 15 fps), so to build in both H.320 compatibility and custom compression capabilties creates a more complex system internally and externally (with device handshaking).

This solution does not yet consider the limitations of sharing a host desktop that has a higher resolution than the client's SVGA monitor can physically handle. From a technical level, sending a signal higher than the monitor can handle will permanently damage the monitor. The CSCW box can be made to hold different monitor IDs (and their respective capabilities) in an EEPROM, or installation software can write a specific monitor's capabilities to the EEPROM. But not all monitors send an ID, and having the

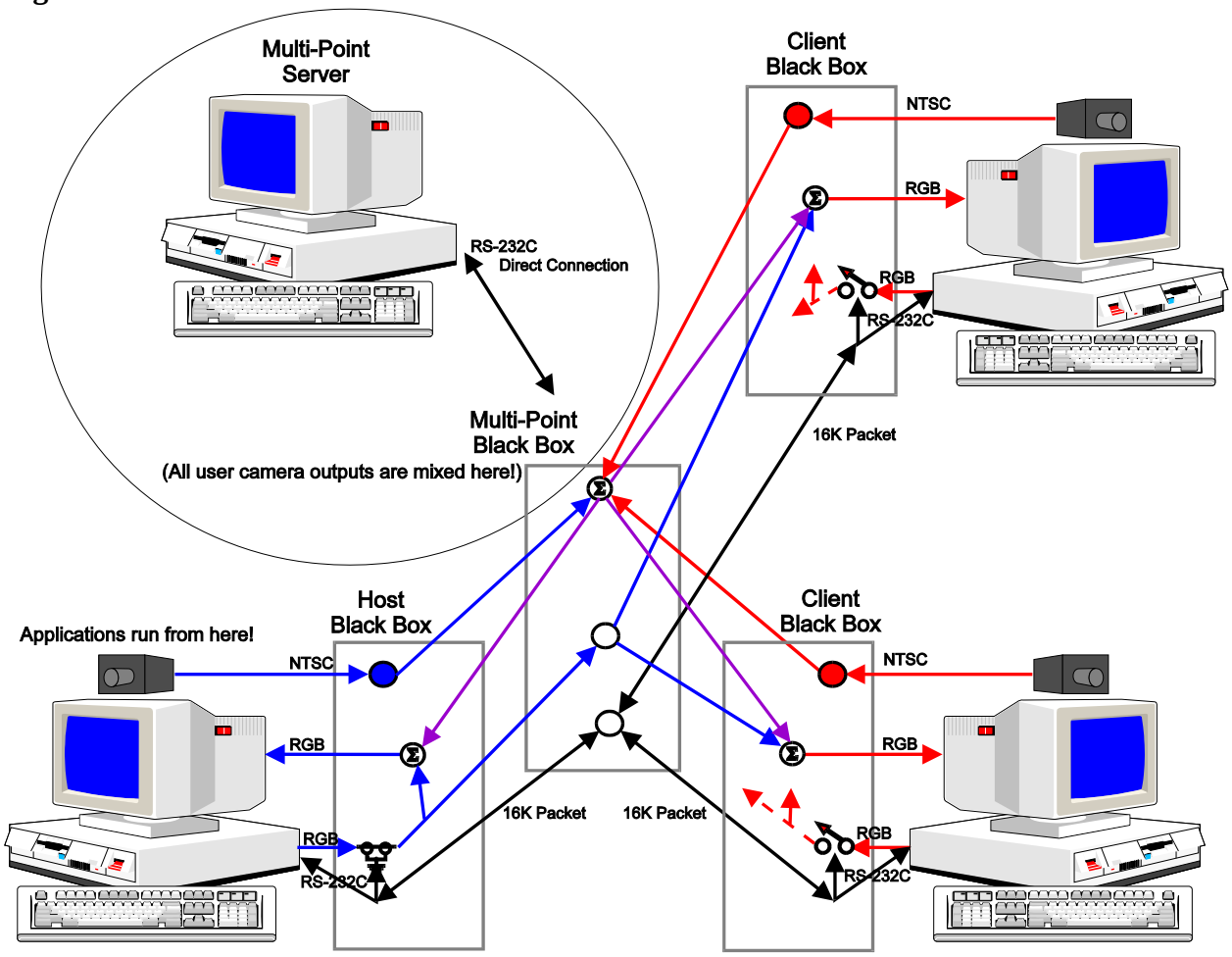
EEPROM written through installation software for a single specific monitor reduces its portability. Once this technical problem has been overcome the question remains, from a user level, on how a higher resolution image can be effectively displayed on a lower resolution monitor. There could be image interpolation, image panning or image lockout.

In conferencing mode, each user is sending and receving a camera video signal. But, in desktop sharing mode, the host is sending 2 video signals (shared-desktop and host camera) and receiving 1 video signal (client camera). The client, on the other hand, is sending 1 video signal (client camera), but receiving 2 video signals (shared-desktop and host camera).

3.4 Multi-Point Architectures

Multi-point architectures will be explored. Multi-point server hardware and software would need to be designed and prototyped. Since this opens an entirely new set of variables to be discussed, it will be defined in more detail as the other phases are completed.

Figure 2 Multi-Point Architecture



(end of document)