
Architecture of a shared-image electronic whiteboard in telemedicine
Michael Fromme*, Helmut Pralle
Lehrgebiet Rechnernetze und Verteilte Systeme, University of Hannover, Schloßwender Str. 5, 30159 Hannover, Germany

Abstract
In this paper we present a software whiteboard for multimedia conferences. The whiteboard is specialised to load and share high-resolution colour images while using colour management functions to provide an accurate colour representation. The whiteboard is used in the telemedicine project INTER-FACE, which builds a virtual environment for tele-consultations between specialists in pre-operative treatment and the planning of the cranio-maxillofacial surgery for patients with facial abnormalities.

Keywords: Multimedia; Conferencing; Whiteboard; Colour management systems; Telemedicine

1. A shared image whiteboard

1.1. Whiteboard applications

Shared electronic whiteboards are common tools for desktop conferencing systems. These tools are able to load documents, such as postscript pages, graphics or images and distribute them to the participants of an online conference. Most whiteboards allow annotations, small pieces of graphic or text that are shown on top of the document to mimic the use of original paper documents in discussions. Other whiteboards offer electronic markers to draw attention to points of special interest while talking about the topic.

Today many online conferences have more than two participants. This can be achieved by using a multipoint server system (Multipoint Control Unit) or by using a multipoint communication network like IP multicast. As conferences are multi-party, whiteboards should also support this type of conference.

One of the first whiteboards that appeared in the desktop conferencing world was wb, an MBone application that was written by Van Jacobson and Steve McCanne at the Lawrence Berkeley National Laboratory [1]. wb has been in heavy use by MBone users in teleteaching applications. It uses plain graphics or postscript pages as documents and was able to annotate the document with graphics and text. To support multi-party conferences, wb uses IP multicast and was designed to allow a large number of participants in a session: more than 100 participants were not too unusual. wb uses a transmission protocol that was designed for this purpose, Scaleable Reliable Multicast.

Another popular whiteboard is the whiteboard tool incorporated in Microsoft’s NetMeeting™ [9]. It has similar functionality to wb, but does not load postscript documents. This whiteboard uses ITU’s T.120 and T.126 protocol, which were designed to support ITU H.320 or H.323 videconferencing systems with still
image distribution and annotation [5–8]. In contrast to wb, T.120 does not use IP multicast for multi-party conferences, but uses conferencing server (the MCU) or a meshed communications approach.

1.2. Related work

In the past 5 years whiteboards have been popular work topics in the computer online conferencing community. Several projects built whiteboards that cover different aspects of collaborative tools:

• WBD [13] is a whiteboard compatible with wb and is available for current operating systems.
• TeleDraw [14] (also known as TeleCanvas) was developed by Andreas Rozek of the communication systems and BeWiWu Development organisation of the Stuttgart University Computer Centre for the MERCI project. TeleCanvas is a general-purpose whiteboard with limited bitmap handling facilities.
• The Department for Mathematics and Computer Science at the University of Mannheim, Germany, have developed dlb, the Digital Lecture Board [15]. The dlb has features specific to a lecture environment, like student annotations, telepointer, voting, etc. Further, preparation, actual teaching and post-production phases are supported.
• The “Authoring on the Fly” (AOF) [16] whiteboard allows recording of a session and storing them for later replay, for example, in an asynchronous learning environment.
• dlb and AOF have been merged to the project mlb, the multimedia lecture board [17].

1.3. Graphic- or image-centric whiteboards

IWB [3] (see Fig. 1) regards images as documents; annotations are drawn on a transparent layer above the image, not destroying image information. This approach has some advantages: the image is protected against accidental moves, deletions or other changes. It can be processed easily with image processing algorithms like luminance control, contrast filtering, etc. without selecting it. Object capture of annotations is not affected by the underlying image.

Fig. 1. The image whiteboard.
1.4. Colour management

Visualisation of colour images is a serious topic, as all colour-processing devices operate in a different colour space. The colour spaces define the range of colours, which can be expressed by the device and the coding of the colours. As scanners, digital cameras, displays or printers have very different physical basics for colour reception or production; these devices have different colour characteristics and therefore different colour spaces.

Differences in colour processing cause errors in colour reproduction. To maintain the best possible quality, these errors have to be minimised to a fractional, not visible degree. This task can be accomplished with a colour management system (CMS). A CMS uses device profiles, which are stored along with device drivers. These profiles and their file formats are defined by the International Colour Consortium (ICC)[4]. After input devices such as a scanner read a colour image, the input device profile is converted to an image profile and is stored along with the image. This image profile can be used together with an output device profile to correct errors that are introduced by input and output characteristics of the devices.

Another approach is to convert the image to a device-independent colour space after loading. This eliminates the need to transfer the ICC profile along with the file, as profiles for standard colour spaces are constant. A common standard colour space is sRGB, which has been defined by Microsoft and Hewlett Packard to meet most characteristics of display devices. For output, the image is converted once again according to the profile of the output profile. For highest accuracy, the first method obtains better results, as the image needs to be converted only once.

Modern operating systems like recent Microsoft Windows or MacOS releases have incorporated CMS functions. But to take advantage of this functionality, image-processing software has to take ICC profiles into account and to use CMS functions. This requirement is also true for image processing whiteboards.

1.5. Manipulating images

Besides the original reproduction, images often benefit from image processing such as changes in contrast or luminance to reveal details. As current display technology offers rather small displays with resolutions of typically 1024 × 768 pixels, the image must be zoomed to get to the actual resolution. Whiteboards should offer this kind of functionality.

In many collaborative scenarios, the participants like to make these changes in their private space, without affecting the view of other participants in an online session. After finding a detail, all whiteboard views need to be synchronised to this view: image processing, zoom factor and view position should be transmitted to all other whiteboards, forcing all session member to see the same image part with the same characteristics.

Besides image manipulation, whiteboards allow annotations like simple geometric shapes, text fields or markers. Markers are icons that are fixed to an image location and display the name of a session member to allow easy identification of the marker. Annotations should be stored on a different layer to avoid damage to image data.

2. Network architectures for whiteboard use-cases

2.1. Point to point

The simplest configuration of a whiteboard session is a point-to-point connection between two session members. In this case, both whiteboard instances use a TCP connection to an arbitrary port. Both members configure the peer addresses and start the connection. The whiteboard sends all data over this TCP connection.

2.2. Multicast

IP multicast is used for multi-party conferences. While point-to-point connections use TCP for a secure data connection, IP multicast is restricted to UDP. A reliable transport protocol must be integrated into the application program. The whiteboard uses the Lightweight Reliable Transport Protocol from INRIA to implement this [10].

In order to set up a multicast connection, the conference members choose a common IP multicast group address and a port. After that, members can join and
leave the online conference by opening or closing the connection.

2.3. Server-driven point to multipoint

Currently, a major drawback of IP multicast is the rare deployment of IP multicast routing facilities. While, most academic networks allow worldwide IP multicast connectivity, common access networks of providers with dial-in or xDSL connections do not support multicast routing. To allow subscribers of commercial Internet providers access to an online conference, connections are restricted to unicast addresses.

Firewalls are another insuperable barrier for IP multicast networks. Most firewalls do not support translation or filtering of IP multicast packets and divide IP multicast networks in different partitions. Using unicast addresses is the easiest solution for connecting machines that are located behind firewalls.

To allow for this, we can deploy a logical star topology for whiteboard connections. The node in the centre of the star is a multipoint distribution service like the H.323 MCU. It accepts connections from whiteboards and mimics the IP multicast service as it copies received data to all other participants. The image whiteboard can use this simple service, because the protocol is inherently multi-party capable. While the transport connection is a point-to-point connection to the whiteboard server, the logical connection is a group connection between all members.

3. System architecture of the image whiteboard

3.1. Architecture of image handling and visualisation

Fig. 2 shows the architecture of the image whiteboard. The image whiteboard is implemented in Java and uses the Java Advanced Imaging (JAI) API for loading, displaying and processing image data. Annotations are handled separately from the image and are displayed on a separate layer in front of the image. JAI uses native functions to compute intensive operations in order to get sufficient performance, independent of the underlying virtual machine. All operations are available in plain Java. JAI can also be used if these native functions are absent.

JAI supports a many image file format. This allows the whiteboard to import images from most sources without conversion. JAI also supports ICC profiles. The whiteboard transforms all images into the standard colour space sRGB while they are loaded. All shared pictures are transmitted in sRGB format. Output devices should be calibrated to display or print sRGB images in order to get the correct colour representation.

3.2. Group communication architecture

For the design and implementation of the group communication protocol, we used a group-aware middleware technology research prototype, which was developed at RVS some years ago and had first been used in the Confman project [2]. This middleware, called Remote Object Invocation (ROI), is available for C++ and Java and uses a CORBA-style Interface Definition Language (IDL) to define remote accessible interfaces, data types and method signatures. ROI extends remote method invocation, as it is widely known by CORBA or Java RMI to group communication. With ROI, a remote invocation of a method is not restricted to one remote object, but can be carried out synchronously for a group of objects at the same time. The group-call features of ROI allow the design of peer-to-peer protocols with M:N cardinality at IDL level, which greatly simplifies the implementation of these protocols. Programmers have to deal with methods and method calls and do not have to design and implement protocol data units.

In the past, most middleware technology only allowed remote function or method calls to be carried out, neglecting the need of transporting or migrating
objects from one process space to another. ROI offers object by value semantics and therefore object transfers and migration from the start. By offering the full semantics of method calls, by reference and by value, the application designer can decide whether data is transferred from one process to another or not. This decision has a great impact on application performance, especially in group communication scenarios, where a reference-only semantic leads to a bottleneck at the object owner, if every other group member needs to make remote calls to get information from that object.

Another important part of ROI is the support of various transport protocols like TCP, UDP or reliable IP multicast protocols. These protocols can be used interchangeably and fairly transparently by the application. The whiteboard implementation uses TCP and LRMP for peer-to-peer and IP multicast connections.

3.3. The whiteboard protocol

The whiteboard group messaging protocol is made up of three functional components:

- Group management (join and leave conferences, discover current members);
- Image or annotation retrieval (retrieve already distributed images or annotations for late joiners);
- Image and annotation sharing (share new images and annotations).

The first two components are used in the synchronisation process, which takes place at the start of the communication. This phase is shown in Fig. 3. That figure shows three objects: the group server object IwbServer, which receives ROI calls, the ROI connection stub, which represents the remote server objects, and another connection stub, which is created as a TCP connection to transfer the images that are already known by the other members.

The synchronisation phase is examined step by step:

1. The new member sends a hi() to the group, announcing himself as a new member.
2. Every member of the group replies with a welcome() message.
3. After receiving welcome() the new member adds the sender of the welcome() to his local group member list. The group member list is built this way.
4. After sending the welcome(), every peer also sends a joinGroup() message carrying member
information such as user name. This message is used by the whiteboard to identify and display human-readable member information in the member list or in annotations.

5. After receiving the first `welcome()`, the new member establishes a unicast connection to the sender. This connection is used for the update process.

6. The new member calls `getUpdate()` on the unicast connection peer. The peer sends the data of all images and all annotations to the new member. After receiving this data, the peer is up-to-date.

There are two cases that deal with missing information. These situations occur if updates like new images or new annotations are sent to the group while the new member is getting the initial information. In that case, the new member may lose updates. Then following actions are taken:

- If a member discovers a missing image (which was distributed while the synchronisation process takes place) a TCP connection to an arbitrary member is used to get this image data.
- If a member discovers a missing annotation, the annotation is requested from the group by issuing a group call. Only the first result of the group call is used. This imposes some overhead but is feasible, as annotations are general small in size.

For identification, images and annotations get unique numbers. The numbering scheme uses a number that is unique in the numbering space of a local instance and a number that uniquely identifies the instance. To avoid problems with concurrent image distribution, which may result in overloads of the network, a session-wide token is used to serialise image distribution. This is done with the `requestToken()` call.

Fig. 4 shows a simplified version of the protocol definition in ROI IDL. In this IDL, call semantics is `call-by-value` on default, as it is for most programming languages (Java is an exception). To get a `call-by-reference` semantic, the formal arguments are prefixed with a “ref”. Method calls are synchronous by default. Synchronous group calls would return the first result that was received by the caller. Another kind is the `group-call` semantic. In this case, the call returns an array of all results the caller has received. The application maintains a group member list of peers, which are expected to return a result. If results from a peer are missing after a timeout, due to a failure, an exception occurs.

As the application depends on reliable and ordered delivery of the multicast requests, the whiteboard uses a reliable multicast protocol, LRMP. This protocol

```
interface iwbConnection {
    // Multicast calls
    oneway hi();
    oneway welcome();
    groupcall bool requestToken();
    oneway setImageData(SharedImage si);
    groupcall int bye();
    oneway joinGroup(GroupMember gm);
    ref iwbConnection requestUnicastConnection();
    groupcall ref iwbConnection requestUnicastConnection(SharedImage si);
    groupcall SharedAnnotation getAnnotation(SharedAnnotation sa);

    // Unicast calls
    oneway updateReady(SharedImage si);
    oneway getUpdate(ref iwbConnection ic);
    oneway getImage(ref iwbConnection ic, SharedImage si);
    oneway getImageData(SharedImage si);
}
```

Fig. 4. Simplified IDL definition of the protocol.
guarantees ordered and reliable delivery of multicast packets within a dynamically managed group and performs bandwidth adaptation to current network characteristics.

4. Application of IWB in INTER-FACE

4.1. The project INTER-FACE

The INTER-FACE project is carried out by the Department of Oral & Maxillofacial Surgery, University of Technology, Munich, and the RVS/RRZN, University of Hannover. Project partners are the Fraunhofer Institute FIT, the Polyclinic of Orthodontics, Heinrich Heine University, Düsseldorf, Marienhospital Stuttgart and three orthodontists in the Munich area.

The clinic at the TU Munich does about 100–130 maxillofacial surgeries with bone rearrangements each year. All surgeries are planned with the aid of computer tools [11]. Patients come mainly from Bavaria and are treated by orthodontists. INTER-FACE is going to set up an experimental environment, in which orthodontists can work together with oral surgeons of university hospitals. This allows pre-operative treatment and the planning of the cranio-maxillofacial surgery for patients with facial abnormalities to be carried out by means of a video/multimedia conference. The images of the orthodontist, the surgeon and most important of all, of the patient are transmitted by the videoconference, while accompanying documents like still images of an intra-oral camera, X-rays or tomography are transmitted and displayed with the image whiteboard.

4.2. The network structure

INTER-FACE combines two kinds of users: academic specialists in university hospitals and orthodontists that work in their practices. Both user communities have very different possibilities for access to networks: University hospitals are generally equipped with high-speed campus networks, which are connected to a research network like the German GWIN. Orthodontists and hospitals are located in cities, and usually do not have access to a high-speed Internet connection. One technical aim of INTER-FACE is to provide orthodontists and hospitals with at least medium-speed Internet access to allow online videoconferencing.

To achieve this, we equipped the orthodontists and hospitals, which do not already have a high-speed Internet link with a 1 Mbps SDSL connection. This connection provides permanent IP addresses and feasible bandwidth, which is suitable for quality-grade videoconferencing. Fig. 5 shows the topology of the network.

As common SDSL access networks do not provide an IP multicast service, we need to use the whiteboard MCU server. Its use is also required due to firewalls at the network borders of the university hospital networks.

4.3. The use-cases for the whiteboard

The INTER-FACE scenarios demand the exchange of image data quite often: photographic pictures of the patient, X-rays and pictures from medical applications are used in clinical discussions. These images provide
vital information sources for the orthodontist and the surgeon in addition to the video image of the patient. The whiteboard supports discussions on these images with its annotation features. Most orthodontists and hospitals store medical data and images of the patient in digital form. This allows easy access of these images for loading them into the whiteboard. Other images, stored on film or in printed form, have to be digitised before the consultation starts.

Novel medical applications allow interactive rendering of assumed post-operative results. These applications should be usable in the discussion process along with still images. It has to be evaluated, whether we need to create an interface between these medical applications and the whiteboard or if general application sharing, for example, with NetMeeting, is suitable.

5. Further development

We have presented software that allows interactive sharing and discussion of high-resolution images and that allows accurate colour reproduction by using CMSs. This whiteboard is available for machines that run Java 1.2 software. It supports multi-party conferences in various network topologies by using IP multicast or a TCP connected whiteboard MCU server.

In the INTER-FACE context, the whiteboard should be adapted as closely as possible to the needs of the medical consultation process. Enhancements include changes in the graphical interface, the input formats and functionality. The specific requirements are taken from the project itself, so evolution of the whiteboard would be an iterative process. The same applies to interfaces to medical applications or appliances, which might be necessary in the future.

For INTER-FACE, as we are working in a pilot project environment, encryption of data and A/V streams is not necessary. Of course, further use of this new medical application needs security features, which have to build into A/V codec’s, videocconferencing set up systems and other tele-medical applications like the whiteboard. We are looking forward to using the ROI security system, which is based on public key certificates, to ensure secure communication for whiteboard data.

Acknowledgements

The project INTER-FACE is supported by the German Academic Network Organization (DFN-Verein) with funds from the Federal Ministry of Education, Science, Research and Technology. The whiteboard basics were developed in the MBone-Tools-Project [12], a project that ran from 1997 to 2000 and was supported by the DFN-Verein. Jan Riechers and Timon Heinze contributed much work to ROI and to the implementation of the whiteboard.

References

Michael Fromme studied electrical engineering at the University of Hanover and received his diploma in 1996. Since 1997 he works as a research staff member at the Institute for Computer Networks and Distributed Systems. His major interests include distributed multimedia systems and group communication in high-speed networks.

Helmut Pralle, University Professor, Dr.-Ing., studied Engineering Sciences at the Technische Hochschule Karlsruhe and Hannover. He worked as an assistant at the Institute for Applied Mathematics in the Division for Electronic Calculation and at the Computer Centre of the Technischen University of Hannover. He received his doctorate in control theory in 1967. From 1968 to 1971 he was Head of the Computer Centre of the University of Hannover and since 1973, Director of the Regional Computing Centre of Lower Saxony. In 1986, he was appointed Chair of the Institute for Computer Networks and Distributed Systems of the Department of Electrical Engineering and Information Technology at the University of Hannover. Since 1996 he has been a member of the administrative council and the executive committee of the DFN Association. He delivers expert reports and vocational guidance for the German Research Council (DFG), ministries and the industry. His major interests include distributed information systems and high-speed data communication.