tation of the editor. Germano Caronni implemented the Browser, Stephan Hächler the network management and Urs Künzler the ACU user interfaces. Ariane Haller was helping me lot with finding the right words and grammar.

References

[1] Borenstein, N., "Multimedia Applications Development with the Andrew Toolkit", Prentice-Hall (1990).

[2] Burkhardt, C., Lubich, H. and Wilde, E., "Implementationsbeschreibung zum ZBF-Projekt 224 Z: MultimETH", Project report, Swiss Federal Institute of Technology (1993).

[3] CCITT Recommendation X.219, "Remote Operations: Model, Notation and Service Definition", Blue Book (1988).

[4] Gale, S., "Adding Audio and Video to an Office Environment", in: Bowers, J. and Benford, S. (ed.), *Studies in Computer Supported Cooperative Work*, North-Holland, pp. 49–62 (1991).

[5] Hächler, S., "Management-Tool für eine LAN-Erweiterung über ISDN", Master's Thesis, Swiss Federal Institute of Technology (1993).

[6] Lubich, H.P., "MultimETH: Ein Beitrag zur Konzeption eines Echtzeit-Multimedia-Konferenzsystems", Ph.D. Thesis, Swiss Federal Institute of Technology (1990).

[7] Minör, S. and Magnusson, B., "A Model for Semi-(a)Synchronous Collaborative Editing", in: de Michelis, G., Simone, C. and Schmidt, K. (ed.), *Proceedings of the Third European Conference on Computer-Supported Cooperative Work – ECSCW'93*, Milano, Italy, Kluwer Academic Publishers, pp. 219–231 (1993).

[8] Robinson, M., "Design for unanticipated use...", in: de Michelis, G., Simone, C. and Schmidt, K. (ed.), *Proceedings of the Third European Conference on Computer-Supported Cooperative Work – ECSCW'93*, Milano, Italy, Kluwer Academic Publishers, pp. 187–202 (1993).

[9] Wilde, E., "Multimedia Joint Editing Based on Reservations", in: *Proceedings of the 3rd Australian Multi-Media Communications, Applications and Technology Workshop*, Wollongong, Australia, pp. 198–209 (1993).

- *Synchronous collaborative editing*
  If two or more users are editing synchronously not only the same document but also the same part of it, the communication facilities of the conferencing system (i.e. chat, broadcast or audio communications via telephone) can be used. All users are viewing the same reservation (which incidentally is reserved by the user who actually performs the changes), which is updated frequently with an autosubmit setting which initiates updates in short intervals.
- *Asynchronous collaborative editing*
  A more asynchronous style of collaborative editing can be achieved by exploiting the open window command. For each of his reservations, the owner opens a separate window and works within it as if it were a complete document. Using this style of working, users have one window with the view of the whole document and several others with the contents of their reservations. The complete document's window can then be used to see which parts are already reserved and which ones can still be reserved. The interaction with the other conference members can be restricted to querying other users to commit or unreserve reserved parts.

Because of the diversity of available mechanisms for interaction and collaboration, it is easy to adapt the system to the working style preferred by very different users. While a subgroup of people is editing one chapter of a document in a very tightly coupled, nearly synchronous style, using the audio communications facilities, other chapters can be edited by remote users who only submit actual versions every 10 minutes or so and do not communicate with the other users in any way than implicitly by submitting actual versions.

5.0 Conclusions

It has been shown that rather slow communication lines can be used to design and implement a multimedia conferencing and editing system. The data communications are handled by the base system and there are several other parts of the system which are used to provide more advanced functions. One of these parts is the interface to a special device which allows easy configurable and dynamically reconfigurable telephone conferences. The editor, another special part, employs a multimedia document model together with a reservation scheme to provide the users of the system with multi-user editing. The combination of both a conferencing system and a multi-user editor makes it much easier for the user to synchronize his work whenever possible. Asynchronous working styles, however, are also applicable by using the editor only, i.e. without any of the conferencing system's communication methods.

Acknowledgments

I owe my thanks to all the members of the MultimETH research team, either permanent or temporary, for the work devoted on the design and on the implementation of the system. Hannes Lubich was responsible for the design and implementation of the base conferencing system. Christoph Burkhardt was responsible for designing and implementing the audio communications facilities, whereas Werner Almesberger, Markus Wild and Daniel Bauer helped a lot during the implemen-

changes, the content of the reservation is selectable and thus also reservable by any other user of the editing session.

- *unreserve*
  Whenever it is necessary to restore the original content of a reservation, i.e. to discard all changes made during the existence of this reservation, the unreserve operation can be performed. It inserts the original content of the reservation into the document. With the unreserve operation, a reservation-level undo can be applied.

This design of the editor makes it easy to perform collaborative editing thanks to the simplicity of the operations provided. Another useful feature is the *open window* operation, which also is available for each reservation. Normally, reservations are displayed in separate windows, however inside the document's window. From a logical point of view, this is correct, considering the reservation's content still belongs to the document, but sometimes it may be preferable to have a separate window for a reservation. This becomes possible with the open window operation, which allows to open a new window for a reservation. The reservation within the document is then displayed as an icon reminding the user that at this place there is a reserved part of the document. The following chapter will illustrate the usefulness of this feature.

4.0 Using the System

Robinson [8] describes why software should allow users to carry out procedures according to their needs and should not dictate only one sequential path of possible operations. Rather than prescribing one particular way of carrying out a task, there should be a variety of possible ways which are all supported by the system. Each user then has the opportunity of working in the way he wants to and he is used to.

When the MultimETH conferencing and editing system is applied, many possible ways of workgroup interactions are supported. The basic conferencing system provides the user with the functions necessary to define new users, to form conferences, to use special roles within conferences (such as, e.g. chairperson or secretary), to query the system about existing and running conferences, etc. Users located at remote locations can be integrated by applying the network management part of the system which enables an especially authorized user to connect distinct LANs over ISDN and therefore to make OSI connections possible.

Another level of collaborative work can be achieved by using the browser as a means to store and manage all documents and other data relevant for a conference. The browser makes it easy to transfer data from a local filestore to the SWS and back and can be seen as a special file manager which disposes of MultimETH documents, users and of conferences. Moreover, it shows whether a document in the SWS is currently edited or not, which makes it possible to see which documents are locked and which can be moved or deleted. The editor can be started by selecting a document and pressing a button, the browser then decides whether a new editing session is started or the user joins an already existing one (provided his access rights are sufficient).

The editor itself supports collaborative work on documents in many different ways. Depending on how the editor's features and the mechanisms of the conferencing system are used, it is possible to achieve a more synchronous or asynchronous way of editing.

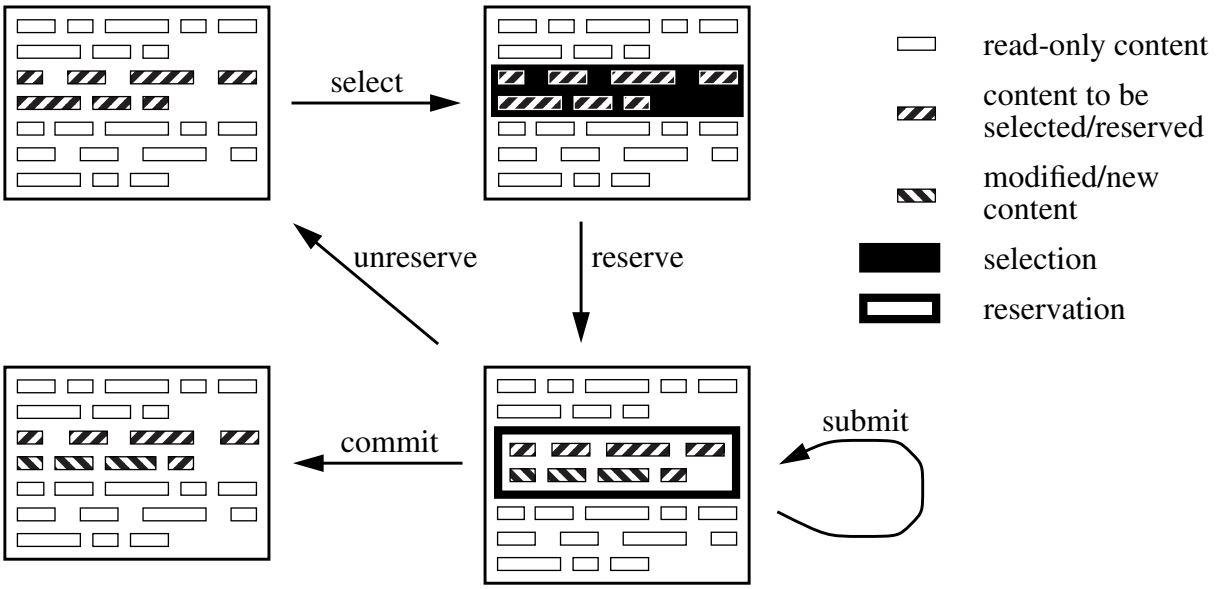

Figure 2: Reservation Model

wait until the part is again available for reservation. Within the scope of reservations, the document can be worked upon in the same way as when using the normal Andrew editor.

- *submit*
  All the changes made within the reservations are only visible locally, for all other users of a multi-user editing sessions do not notice any changes. This strategy saves bandwidth and is based on the assumption that most multi-user editing only needs a restricted level of synchronicity. However, if for any reason a user wants to see the current version of a second user's reservation or wants to propagate his changes to other users without giving up a reservation, the submit operation may be performed. Submit makes an update of a given reservation for all the users concerned, regardless of the fact who requested this update.

  For some applications of the system it may be recommendable to have a closer coupling between the users of a collaborative editing session. For this purpose, it is possible to activate an autosubmit function individually for each reservation. Autosubmit periodically submits the actual state of a reservation to all users, which is why it becomes unnecessary to carry out the submit operation manually. Autosubmit can be configured with an upper and a lower bound for the frequency of updates and reacts also sensible to the amount of changes made. Therefore it is possible to configure autosubmit in a way especially suited to the characteristics (and to the associated costs) of the underlying transport medium.

- *commit*
  When a user is ready with modifying a reservation, these changes can be accepted and be included as part of the document itself. The commit operation gives up an existing reservation and substitutes the former content (when the reservation was claimed) with the present version of the owner of the reservation. After having made and having committed the

3.0 The Editor

Although the editor is a special part of the MultimETH system, it can also be used as a stand-alone program. This is possible due to the subdivisioning of the system into the special parts and the transport service provided by the base parts. Another transport medium available is TCP/IP, which we also used for testing purposes during the editor design. The editor is based mainly on two models: the document model and the reservation model.

The document model is an extension of the Andrew Toolkit's (ATK) document model. The ATK, as well as its design and its programming, is described in detail in [1]. In ATK, every document consists of a collection of *objects* which may contain other objects with no limitations about the nesting of objects. An advantage when using the ATK is its extensibility. New media types (or *classes*[4]) can be added dynamically without changing the system itself. This is possible since each object (or media) type handles its own data and view without any processing by the application which uses this object type.

The MultimETH project added mainly two new insets to the ATK system and to its editor: one representing the document's outmost level, which may be considered the document itself, the other one representing reserved parts (see below) which are necessary for our model of multi-user editing. The rest of the ATK document model remained unchanged, which makes it easy for experienced Andrew users to work with the editor since there are only few new operations to learn.

As stated above, the concept of multi-user editing employs a reservation scheme based on a simple model of user actions. The reservation model of the multi-user editor is shown in figure 2. When considering this figure, it is important to notice that the content of the document is not restricted to text but may consist of any media type implemented within the editor system (i.e. available with the ATK[5]). The editing cycle consists of a standard selection mechanism (e.g. making a selection with a mouse) and of four special operations which implement the multi-user awareness. All operations are context sensitive, i.e. they can only be performed in the appropriate context. It is important to recognize that without a reservation, the document is read-only and no user is allowed to alter the document. In this state, the only operation possible is the reserve operation which may be carried out in order to gain write access to part of a document.

- *reserve*
  This operation reserves a selected part of a document exclusively for the user who performed the reservation. Reservations are granted by the editor server. A part of a document being reserved, it is not possible for other users to reserve and consequently to change this part. Reservations are granted in a first-come-first-serve manner, which means that if two users simultaneously perform a reserve operation for the same or for two intersecting parts of a document, the user wins whose request is processed first. The second user then has to

[4] Media types are actually implemented as pairs of classes, one class implementing the dataobject, the other class implementing the view. In ATK terminology, each pair is referred to as an *inset*.

[5] Because multi-user editing requires some kind of access protection, media types which should be properly used with the editor have to implement a procedure for setting them read-only. Unfortunately, most of the media types distributed with the Andrew system do not implement such a procedure.

This list gives an overall impression of the functional units of the conferencing system. These units all communicate by using the base part. The base part of every client is responsible for setting up an OSI connection between the server and the client. Data exchange is entirely carried out via OSI remote operations. All special parts make usage of the base part to execute these remote operations. The conferencing system can therefore be regarded as a special multiplexor which uses one network connection for several purposes. See [9] or [2] for a detailed description of the data communication issues.

2.2 Audio Communications

Due to the importance of speech in the collaboration of people (see [4] for a thorough study), audio communications are also incorporated within the MultimETH system. This design decision together with the usage of low bandwidth data communication lines made it necessary to utilize separate audio channels. Within the system, audio is transmitted via normal telephone lines, either analog or digital. A special device, namely the ACU, is responsible for initializing and accepting connections for a conference and for mixing the correct channels together, optionally with some normalization of the volume for different telephone connections[1]. The ACU is equipped with analog and digital telephone line boards.

The ACU employs a most flexible method in connecting conference members with each other by means of the *connection matrix* (CM). Applying the CM, it is easy to specify who should speak to whom and who shall hear whom. The CM is simply a matrix of entries which give a weight of how loud a given channel should be heard by every other member with an audio connection[2] to the ACU.

There are some operations which can be performed by the users of audio connections[3], some of these are only possible for one privileged user who is responsible for managing the audio connections of a conference. The privileged operations include *connect* and *disconnect,* which are used to set up and finish telephone calls to users. The conference system itself stores a directory of the user's phone numbers, i.e. if there is no change to this default, the connect operation can be performed without any additional parameters. Temporary users (which do not belong to the computer conference) can be called. e.g. to have discussions with specialists. At all times, the CM can be manipulated in any way to change the characteristics of how members are connected.

A special mode of audio operation is chatting, where two users are entering a temporary one-to-one communication. Every member of an audio conference can request a chat to any other member, and provided the selected chatting partner confirms the request (which is displayed on his terminal), these two users can talk to each other separated from the rest of the conference. When the chatting is finished, they can join the audio conference again. Chatting can be viewed as a special case of forming subgroups within audio conferences.

---

1) These functions are implemented using digital audio signals and a DSP for manipulating them.

2) Since not all of the conference members necessarily need both communication channels, it is possible to join a conference only by means of the data communication or the audio communication, depending however also on the available hardware and the communication needs.

3) These operations are accessible by using the ACU management special part of the system.

2.1 Data Communications

The MultimETH server and the corresponding clients (which are used to represent users) communicate over the data communications line by using the OSI Remote Operations Service Element (ROSE) [3]. This infrastructure was chosen in consideration of the project's main goal, which is to demonstrate that OSI protocols can be used for cooperative work with multimedia information, even in the case of slow communication lines. The MultimETH base system provides the transport infrastructure for all other components of the conferencing system, except the audio communications.

Every component of the system (i.e. the server and all clients) contains on the one hand a base part which is used for communication and the basic conferencing functionality, and special parts on the other, which are employed to provide advanced functionality. The base system, consisting of a base server and base clients, is used for the management of users and conferences as well as the access control for conferences. Moreover, the base system provides some conferencing mechanisms such as chatting (a UNIX talk-like communication between two members of a conference), voting (making queries to all conference members) and broadcasting (sending messages to all members of a conference).

The special parts of the components, which are required for advanced functions, also have client and server components which are to be found with the respective base parts. The special parts of each component employ the respective base part (i.e. either client or server) to communicate over the data connection. There are the following special parts:

- Browser
  This part is intended to provide access to a *shared workspace* (SWS). It can be considered a central filestore accessible to all members of a conference. This SWS can be used to store documents and other data of interest to the conference. The members of a conference can move data from and to the SWS by applying the browser.
- Network Management
  Since MultimETH intends to demonstrate the possibility of transmitting multimedia information over slow communication lines, there is special support for using MultimETH over ISDN links. ISDN connections can be used to connect two separate LANs which are then logically one Network (on layer 2 and above). More details concerning this bridging over ISDN are outlined in [5].
- ACU Management
  This part is required to manage the audio connections. As it will be explained in section 2.2, audio communications are established and managed by a special device: the ACU. This device can be controlled by using the interface for the ACU.
- Multi-user Multimedia Editor
  This part will be described in detail in chapter 3.0. For the conferencing system, however, it is a special part like the others described above. It is necessary to provide the users with the multi-user editing facilities which are required for the collaborative tasks in working groups.

the conferencing system is based on the assumption that low bandwidth channels are sufficient to make multimedia conferencing and multi-user editing possible. This design of the system thus allows the latter to be used over communication lines with 64 kbit/s (i.e. ISDN links) capacity or with even less (like modem connections which may be 19200 or 9800 bit/s). The scalability of the amount of information exchanged makes it easy to adapt the system to the underlying network. The following two chapters present the two main building blocks of this system, which are the conferencing system itself and the editor.

2.0 The Conferencing System

This section will only discuss the properties of the MultimETH conferencing system, which are of relevance with regard to the task of multi-user editing and the support of multi-user editing in general. The system itself is described in detail in [6].

The main goal of the system is to support teleconferencing for small workgroups. There are two types of connections between users and the system: one connection for data transport and the other for transmission of audio. Since the system is conceived for today's low-bandwidth networks, audio is transmitted over normal telephone lines. The general topology of the system is shown in figure 1. The MultimETH server contains a special unit, namely the *audio conferencing*

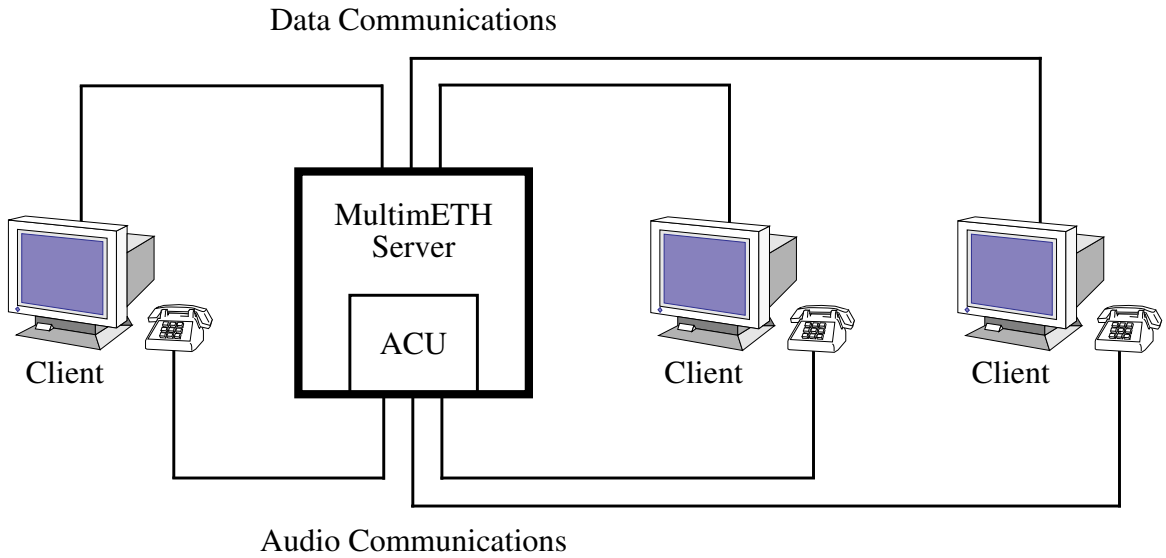

Figure 1: General MultimETH topology

unit (ACU) which manages the connections of the conference members over telephone lines. The following sections discuss the design and the implementation of the MultimETH Server and its components.

Multi-User Multimedia Editing with the MultimETH System

Erik Wilde

Swiss Federal Institute of Technology (ETH Zürich)
Computer Engineering and Networks Laboratory
CH – 8092 Zürich
wilde@tik.ethz.ch

Abstract

Multi-user multimedia editing ought to be supported by different means. Besides technical means required for editing a given document by several users simultaneously, there is also a demand for communication mechanisms which are able to support the synchronization of the users. The system presented in this paper does not only offer multi-user multimedia editing capabilities but also provides a shared workspace and an environment for communication both via terminals and via telephone. The shared workspace is a concept which allows the members of a conference to share documents and other data. Telephone communications enable members of a collaborative editing session to have conference connections and to dynamically form subgroups.

Keywords

multimedia editing, computer supported cooperative work, groupware, teleconferencing, audio conferencing

1.0 Introduction

Multi-user multimedia editing is a cooperative task which needs special support for synchronization. In [7] it is explained why strictly synchronous or asynchronous models may be considered inappropriate. Users want the possibility to work on their own, but need also technical support when synchronization with other users is required. Multimedia technology therefore can be used in two ways when modeling multi-user editing systems:

- The editor itself can be designed as a multimedia editor, i.e. it is possible to use several media types besides text when working on documents.
- The editor environment which is used to provide the users with the functionality they demand for multi-user editing sessions, may incorporate multimedia functionality.

The general model of the conferencing system discussed in this paper is described in detail in [6]. The central idea of this system is to create a *shared workspace* (SWS) that can be accessed by multiple users from different locations and that also provides the latter with an environment which can be used to regulate access to this workspace.

Because of the scarce availability of today's high-speed networks, which are up til now only accessible to people in the research community and only for very few outside of it, the design of