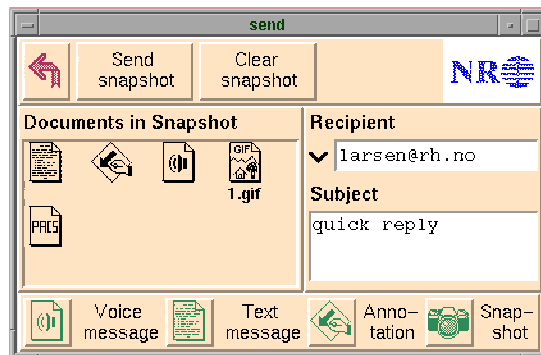
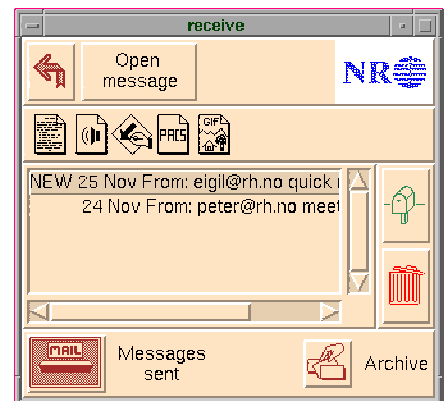


The Snapshot Composer:

The MRD's Multimedia Messaging Facility



&



ITIP/01/95
Peter D. Holmes
Hilde Lovett
Frode Løbersli
Eigil Tapio Skorve

Oslo
December 1995

Tittel/Title:
The Snapshot Composer:
The MRD's Multimedia Messaging Facility

Dato/Date: December
År/Year: 1995
Notat nr/
Note no: ITIP/01/95

Forfatter/Author:
Peter D. Holmes, Hilde Lovett, Frode Løbersli, Eigil Tapio Skorve

Sammendrag/Abstract:

The Middle Road Demonstrator (MRD) is a system for communication support within cooperative work settings. It supports cooperative work by allowing users to share applications independent of time and space. The MRD supports both synchronous and asynchronous communication modalities realized through its desktop conferencing facility and its multimedia messaging facility, respectively.

The Snapshot Composer is the application within the MRD which empowers the system with its multimedia messaging functionality. The Snapshot Composer supports the composition, reconstruction and presentation of multimedia messages. It does not involve itself with the transmission of multimedia messages — that job is performed by standard e-mail services.

This paper briefly presents aspects of the MRD as a whole, then focuses upon the functionality within the Snapshot Composer. The paper closes with a scenario-based description as to how the Snapshot Composer is used.

Emneord/Keywords: CSCW, multimedia messaging, human-computer interface, desktop conferencing, communication support, computer-assisted radiology

Målgruppe/Target group: potential user organizations, software industry, research institutions, NR

Tilgjengelighet/Availability: Open

Prosjektdata/Project data: Funded 50% by NR and 50% by the Research Council of Norway via ESPRIT Project 6155 - EuroCODE: CSCW Open Development Environment.

Prosjektnr/Project no: 902300

The Snapshot Composer

The MRD's Multimedia Messaging Facility

Peter D. Holmes
Hilde Lovett
Frode Løbersli
Eigil Tapio Skorve

Norsk Regnesentral
December 20, 1995

Table of Contents

1	Introduction	1
2	The MRD	1
	The MRD and communication support	2
	Synchronous contact: desktop conferencing	2
	Asynchronous contact: multimedia messaging	3
3	Snapshot Composer Toolkit Functionality	3
	Basic Concepts and Clarifications	3
	Functionality implemented in the Toolkit	5
	Composing multimedia messages	5
	Transmission and reception of multimedia messages	6
	Opening and viewing multimedia messages	6
4	Use of the Snapshot Composer	6
	Composition of messages	7
	Reception of messages	10
5	Conclusions	11
	Acknowledgements	12
	References	12

1 Introduction

The Middle Road Demonstrator (MRD) is a system for communication support within cooperative work settings [10]. The MRD is one of three demonstrator systems developed within the framework of ESPRIT Project 6155 - EuroCODE: CSCW Open Development Environment [5]. The system was designed and evaluated in cooperation with a pilot group from the Department of Radiology at Rikshospitalet [2, 11, 9].

The MRD supports cooperative work by allowing users to share applications independently of time and space. The MRD supports both synchronous and asynchronous communication modalities. In both of these modalities, the MRD provides communication facilities based upon a “show, point and talk” functional profile.

The MRD’s synchronous communication modality is realized through its desktop conferencing facility, while its asynchronous modality is realized through a multimedia messaging facility called the Snapshot Composer (SSC). The Snapshot Composer was developed using EuroCODE’s Snapshot Composer Toolkit (SSCTK). The functional elements within the SSCTK can be assembled as desired by an application developers. As such, the SSC must be understood to be a specific assembly of the SSCTK’s functional elements. Thus there exist operations, interface elements, etc., which are available exclusively through the SSC¹.

The SSCTK contains functional elements designed to support the composition, reconstruction and presentation of multimedia messages. The SSC does not involve itself with the transmission of multimedia messages — that job is performed by the standard e-mail services to which it is integrated.

The SSCTK is based on UNIX/X. It consists of the public-domain packages called MH [14] and `metamail`, as well as new software written to exploit the features within those packages. One powerful feature within the SSCTK is that when creating multimedia messages, the SSCTK creates a logical representation of the message. Using that representation, it becomes possible to create alternative kinds of transmission-ready instances of the message (e.g., MIME, HTML, etc.).

The Snapshot Composer is fully integrated within the MRD; moreover, it is on equal footing with the MRD’s Desktop Conferencing system. These two subsystems are integrated with one another in such a way that the user can seamlessly shift from one communication modality to another.

2 The MRD

The MRD has been designed to fit into cooperative work processes as unobtrusively as possible. Within such processes, there arise settings and situations in which a person needs

1) In writing this document, care has been taken to try and distinguish exactly which operations, functions, etc. are available exclusively through the SSC; in such cases, these are specifically noted as being so. If not noted, the functional elements described herein are a general part of the SSCTK

to, or is obliged or requested to, communicate with another person. It is at these moments that the MRD is used to support the initiation and reception of communication contacts.

With regard to inter-personal communication contacts, at least two contextual dimensions can be distinguished:

- whether the communicative interaction transpires in real-time or not (i.e., *synchronous* vs. *asynchronous* contact); and,
- whether an individual is the one *initiating* a communication contact, or *receiving* such a contact.

The interaction of these two dimensions creates four communication contexts; these are illustrated in figure 1(a). Within a cooperative work setting, this figure depicts how these communication contexts are experienced from the individual's point-of-view, rather than that of the group as a whole.

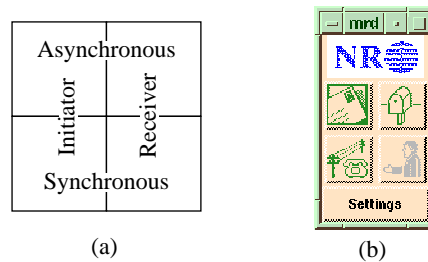


Figure 1 : *Communication contexts within cooperative work settings: (a) as experienced from the individual's point-of-view; and, (b) as reflected within the MRD's top-level interface.*

Figure 1(b) depicts how the design of the MRD's top-level interface directly reflects the four communication contexts illustrated in figure 1(a).

2.1 The MRD and communication support

2.1.1 Synchronous contact: desktop conferencing

The MRD's desktop conferencing support is primarily intended for situations in which persons need to discuss task-related material(s) "right at that moment". It provides for application sharing, telepointing and multi-party audio conversations via the data network. The application sharing and telepointing mechanisms are implemented within EuroCODE's Global Window Toolkit [1]. The conference audio facility is implemented within EuroCODE's Digitized Sound Toolkit [8].

All of the MRD's desktop conferencing facilities are orchestrated by the EuroCODE conferencing architecture, implemented within EuroCODE's Conference Toolkit [15]. The conferencing architecture enables a uniform means for coordinating of a number of conferencing applications. It also offers a well-defined interface through which it is possible to integrate third-party applications.

Many systems exist today which offer this kind of communication support. The focus of this paper is therefore that of the MRD's multimedia messaging facility, the Snapshot Composer.

2.1.2 Asynchronous contact: multimedia messaging

The MRD's multimedia messaging support is primarily intended for less time-critical situations in which persons need to exchange and/or pose questions about task-related material(s). Multimedia messaging can also be effectively used when someone is not available for a real-time conference.

Like the MRD's desktop conferencing facility, the Snapshot Composer also provides show, point and talk functionality — though without the feedback characteristics inherent in real-time communication. When composing a multimedia message, these facilities allow one to:

- create a “snapshot” (i.e., a set of selected documents, or document references, along with certain application state information for each document) to be sent to and viewed by others;
- place simple annotation marks (e.g., arrows) atop² the documents within a snapshot; and,
- include an audio and/or text message along with the snapshot and annotations.

The simple text editor, annotation and snapshot mechanisms are implemented within EuroCODE's Snapshot Composer Toolkit, described below. The simple audio recorder/player is implemented within EuroCODE's Digitized Sound Toolkit [8].

3 Snapshot Composer Toolkit Functionality

3.1 Basic Concepts and Clarifications

This section introduces and briefly explains special concepts and terminology relevant to the Snapshot Composer Toolkit.

- *Snapshot Composer*: The SSC is the primary application within the SSCTK; it comprises essentially all of the Toolkit's fundamental functionality. This functionality is offered through the SSC's top-level windows, `send` and `receive`.
- *Multimedia messaging*: Within the context of the SSCTK, this is the ability to compose, send, receive and reconstruct messages containing more than one media type.
- *Message composition*: From the user's point-of-view, this is the creation of a multimedia message. From the SSCTK's perspective, this is (1) the creation of a logical representation of a multimedia message; followed by (2) the construction of an electronic file having a format suitable for transmission by standard e-mail services.

2) Note: these annotations do not in any way become written into a document's contents. They appear atop a document, but a different logical layer is used to present them on the computer screen.

- *Snapshot*: From the user's point-of-view, this is a set of user-selected documents (e.g. images, spreadsheets, text files, etc.) to be included as part of a multimedia message. From the SSCTK's perspective, this is a logical representation of a collection of document references, along with corresponding application state information for each.
- *Document*: Herein, this term intends to denote (e.g., textual, audio, video) material. Usually, this material can be presented/played by some application and is associated with some file. 'Document' can also refer to a collection of such material, where the collection can be addressed via a set of references.
- *Application state information*: A set of values describing the state of a running application. With respect to the SSCTK's snapshot mechanism, the type and specificity of state information which can be captured often varies from one X application to another. Still, it is usually always possible to obtain the name of the document being presented and the application's display geometry.
Whenever complete application state information is sent along with documents (or document references) within a multimedia message, it is possible to present the documents and reconstruct the state the applications were in when the message was created. Less-than-complete state information simply implies less-than-complete reconstruction of application state when a document is presented.
- *Message reconstruction and presentation*: Herein, 'reconstruction' concerns the re-presentation of a multimedia message. During such re-presentation, applications are started and specific documents are loaded into them. Applications are displayed at specific locations upon the user's screen and, in some cases, the associated documents may be turned to specific pages and scrolled to specific positions. Other state-relevant information may be used during message reconstruction; this depends upon the manner and degree to which an application is snapshot-integrated.
Within the SSCTK, the instructions for message reconstruction are found in each user's .mailcap file, a file describing the relationship between content-types and action to be taken for each type.
- *Content-type*: Within MIME messages, this is a header field which can be used to specify the type and subtype of data in the body of a message, as well as fully specify the native representation (encoding) of such data. (Further technical details can be found in RFC 1521.)
- *Annotations*: Within the context of the SSCTK, these are markers (e.g. arrows) associated with numbered labels for which x- and y-positions are known. A basic annotation application is integrated the SSC such that annotations can be used to help distinguish areas-of-interest within multimedia messages.
- *Snapshot-aware application*: The SSCTK's snapshot mechanism preferentially looks in an X application's "WM_COMMAND" window property when trying to capture state information from a running application. Snapshot-aware applications are those which purposefully update that window property when relevant changes in application state take place.

- *Snapshot-integrated application*: For an X application which doesn't update its "WM_COMMAND" window property, the SSCTK includes exception-handling software in order to integrate that X application with the SSCTK. As the name implies, snapshot-integrated applications are those which are integrated with the SSCTK. The set of snapshot-integrated applications is a superset to that of snapshot-aware applications. For all snapshot-integrated applications it is clearly defined what state information can be captured and how an application's state can be restored using that information.

3.2 Functionality implemented in the Toolkit

This section describes the functionality implemented in SSCTK. The different kinds of multimedia messaging support it offers will be elaborated below.

3.2.1 Composing multimedia messages

Users may often find it useful to include snapshots within multimedia messages. The SSCTK therefore includes a snapshot mechanism — a central part of the SSC.

Composition of a multimedia message, via the SSCTK's snapshot mechanism, transpires in two phases. In the first phase, the snapshot mechanism allows the user to select the documents to be included in the message. When selecting, the user simply points within the application window in which the document is being presented and clicks once. For each document selected, the snapshot mechanism captures certain information about the presentation state of the application. A "snapshot" is a *logical* representation of the collection of document references along with the corresponding application state information for each.

For each document selected, the snapshot mechanism attempts to capture as much information as possible about the presentation state of the application; the type and specificity of state information captured depends upon the degree to which the application is snapshot-integrated. In addition to including a snapshot within a multimedia message, the user may also include an audio recording, text message and/or simple annotation marks. Generation of these message parts is accomplished via simple text and audio editors, and a rather basic annotation mechanism.

The second phase of multimedia message creation involves the construction of a multimedia message having a format suitable for transmission by standard e-mail services. Messages suitable for such transmission will also be called "physical messages" herein. In this SSCTK implementation, functions exist for the creation of MIME-formatted messages. The SSCTK's logical message format is easy to interpret, in order to suit different MIME message composers. As an alternative, it would be easy to generate HTML-formatted messages from the logical message. At present, this is not implemented within the SSCTK.

3.2.2 Transmission and reception of multimedia messages

Once the MIME messages are constructed, they can be sent with ordinary e-mail to the named recipients. In order to be able to receive and open SSCTK-based messages, the recipients require a mail reader which is MIME compliant.

The MRD is designed to filter SSCTK-based messages and direct them to special mail folders. This is accomplished using MH routines. From a general perspective however, the management and handling of incoming e-mail for a user-site is not part of the SSCTK. To correctly install the SSCTK at a new site, system managers should consider the e-mail integration alternatives offered within the SSCTK's design and select the technique most suitable for their site.

For interpreting SSCTK-based multimedia messages, the SSCTK includes a function which reads a file containing a MIME message, and employs a user's .mailcap file to reconstruct and present each part according to its content-type.

3.2.3 Opening and viewing multimedia messages

When reviewing multimedia messages, the SSCTK provides facilities for: obtaining a logical overview of each message, based upon that message's content-types; and, opening (i.e., reconstructing and presenting) each message in its entirety.

The logical overview (or, "content-type overview") for a message is displayed as a set of icons representing the message's content-types; these icons normally look like the icons associated with various applications (e.g., netscape, xv, etc.). The association as to which icon appears for each content-type is tailorable.

The purpose of the content-type overview is such that a user can obtain a rough idea as to the contents and potential size of received messages (before opening them). The logical overview also can serve to provide a reminder as to the contents of a message once it has been read.

The SSCTK naturally provides an operation by which to open multimedia messages, in order to review their respective contents in full. This function reads a file containing a MIME message, and uses the .mailcap file to open each part according to its content-type. Each part of the message is reconstructed and presented in parallel.

When reviewing multimedia messages, it is of course possible to retrieve and/or delete old messages.

4 Use of the Snapshot Composer

This section reintroduces a scenario presented in the original MRD design document [11]. The scenario illustrate how the MRD's Snapshot Composer helps support communication and cooperation needs between doctors at Rikshospitalet. The scenario was one of several identified by the pilot group during the MRD's requirement acquisition and preliminary design phases. It was developed using scenario-based design principles described in [4].

This scenario describes a situation in which a pediatric radiologist needs to receive some advice from a neuroradiologist. Since the neuroradiologist is not available, the pediatric radiologist creates and sends a multimedia message. Two aspects of this communication form are described: the composition of the message and its reception at the other end.

4.1 Composition of messages

Dr. Eigil, the pediatric radiologist, wants to discuss an examination with Dr. Larsen, a specialist in neuroradiology. He tries to reach her using the MRD's conferencing facility, but she does not immediately answer his request. Instead of waiting for Dr. Larsen, Dr. Eigil starts MRD-PACS within the conference³. He leaves active his request that Dr. Larsen join the conference, in the hope that she might soon be back and join him.

Dr. Eigil has the opinion that image 1 indicates a slight change in the state of the patient compared to an earlier image. Dr. Eigil wants to discuss this finding with Dr. Larsen, but she has still not joined the conference. Therefore Dr. Eigil decides to send a message to Dr. Larsen. He begins by arranging the windows on his screen as he prefers that Dr. Larsen shall see them when receiving the message. Already, he has planned to send her image 1 (since he believes it contains sufficient information for Dr. Larsen), some annotation marks and some short textual and audio remarks.

Dr. Eigil pushes the "letter" button (upper-left) to activate the MRD's Snapshot Composer. The Snapshot Composer lets the user select the documents he wishes to send, as well as place annotation marks atop those documents and include oral and written messages. After pushing the letter button, Dr. Eigil starts the small annotation application by clicking upon the "annotation" button within the send window. At this time, Dr. Eigil's screen appears as shown in figure 2.

There are two points of interest within image 1. To mark these points, Dr. Eigil uses the annotation application, activating it by clicking on the "start annotation" button. (This button then turns into the "stop annotation" button.) The application enters a mode in which the mouse is grabbed. A click on the left mouse button leaves a labelled arrow where the mouse is pointing.

When Dr. Eigil is finished annotating he clicks on the "stop annotation" button. To briefly describe when he needs a reply from Dr. Larsen, he starts a simple text editor by clicking on the "text message" button. He then enters a text message asking Dr. Larsen to reply by 2:00 pm, if possible. Dr. Eigil is aware that he could have written the same text within the send window's "Subject" field; however, he believes that a large visible note to Dr. Larsen will have less chance of being overlooked.

3) Here, Dr. Eigil has started a conference in which he is currently the only member.

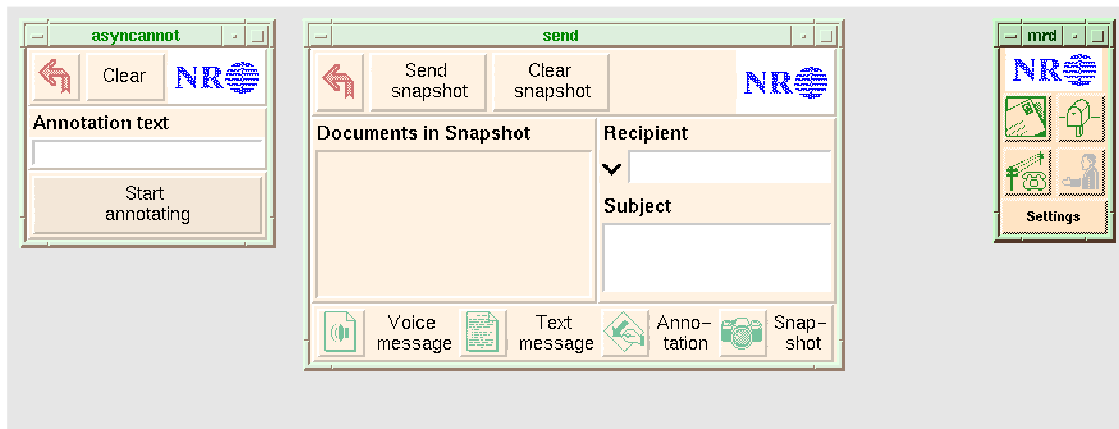


Figure 2 : The Snapshot Composer's send window along with asynccannot, an application for creating simple annotation marks.

When the brief text message is written, Dr. Eigil wants to record a voice message concerning his finding to Dr. Larsen. By posing his queries orally, Dr. Eigil saves himself a great deal of time typing. In addition the annotation marks help him speed the formulation of his query: he can refer to 'mark 1' and 'mark 2' rather than having to provide precise anatomical descriptions. Dr. Eigil starts the simple audio recorder by clicking upon the "voice message" button.

Once the audio application is started, Dr. Eigil pushes the recording button (i.e., the button with the red circle) and starts start talking. To stop recording, he pushes the stop button (i.e., the button with the black square). On those occasions in which he sends several images to someone, he uses natural terms to describe the relative locations of the images (e.g., "the image to the left"); he knows that when the message is opened and viewed by the receiver, the images will appear at the same positions they were in when the message was created.

Lastly, Dr. Eigil selects the documents to be sent to Dr. Larsen. To do so, Dr. Eigil clicks on the "snapshot" button. The cursor turns into a crosshair symbol ('+'). In this mode, the mouse position is used to determine within which window it is pointing. When Dr. Eigil clicks upon image 1, the reference to that image is recorded along with information about the state of the application presenting that image. Dr. Eigil knows that this "click" will include image 1 within the multimedia message.

Dr. Eigil then clicks upon the MRD-PACS application, which has the effect that information about MRD-PACS' application state will be included within the message. What Dr. Eigil understands is that by clicking upon the MRD-PACS application, it will be started automatically for Dr. Larsen when she opens the message; furthermore, he knows that the application will appear for Dr. Larsen just like it is now: Gaute Eide's examinations will be listed in the upper panel, and the listing of images associated with the February 14th examination will be listed in the lower panel.

When Dr. Eigil has selected these two elements, he quits the selection mode by simply pushing the snapshot button again. (The button has remained depressed while in the selection mode; pressing it again raises the button.)

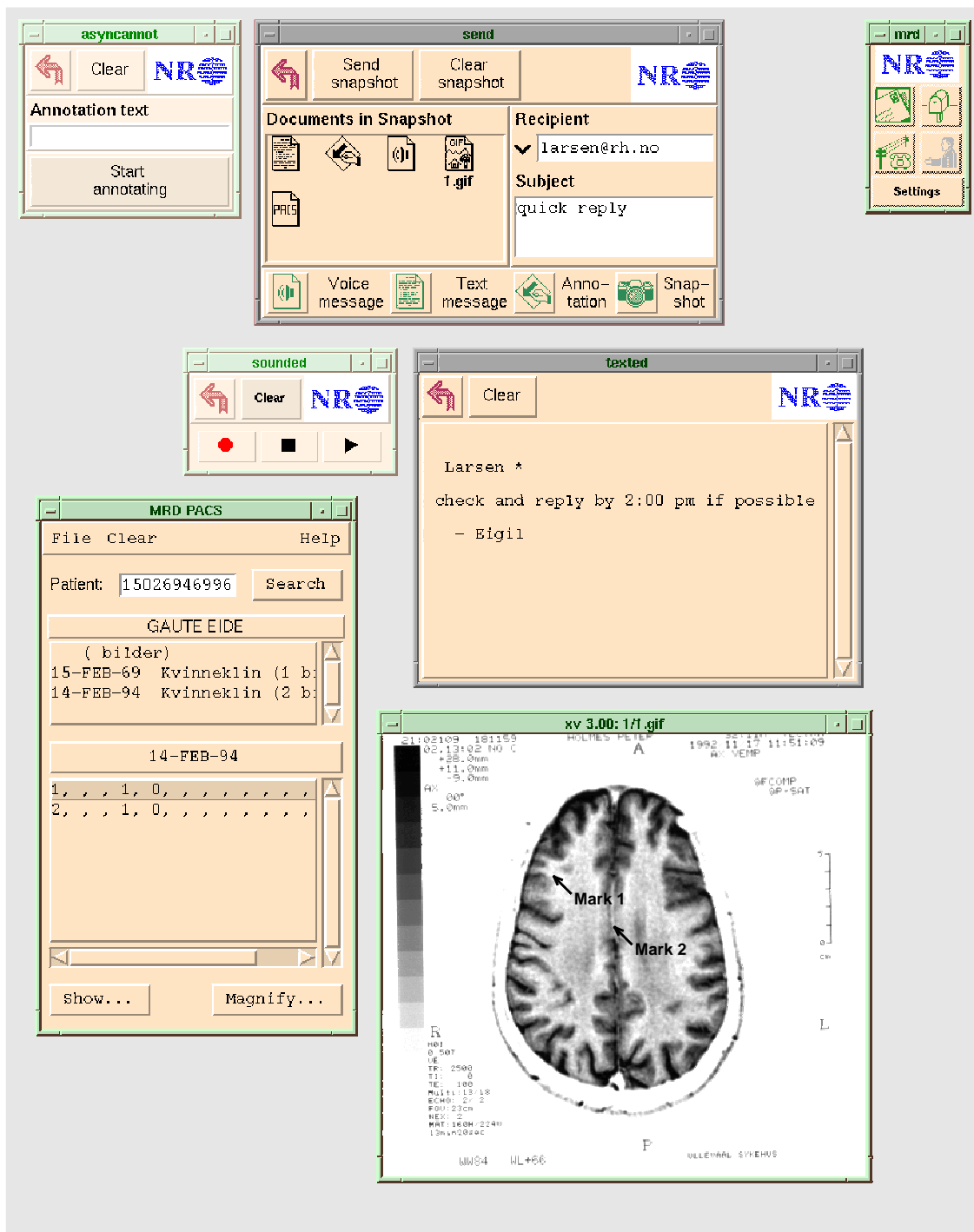


Figure 3: A multimedia message created using the MRD's Snapshot Composer.

The send window's left panel presents the user with a logical representation of the information elements to be included in the message. This representation consists of an icon reflecting the kind of element included. Whenever text messages, voice messages and/or annotations are included in a multimedia message, icons are included for each of them. Icons representing selected documents are also included. In this way, the user has an easy way to see what's included within the multimedia message.

The `send` window offers a field in which it is possible to fill in a subject line; of course a field is available in which to specify a (set of) recipient(s). The `send` window's "clear snapshot" button makes it possible to clear the message and start again.

Dr. Eigil specifies Dr. Larsen as the receiver and types a short note in the subject field indicating the urgency of the message. Just prior to sending the message, Dr. Eigil's screen appears as shown in figure 3.

4.2 Reception of messages

In the early afternoon, Dr. Larsen enters her office and sits down in front of her workstation. She notices immediately that the MRD's "mailbox" button (upper-right) has turned yellow. This indicates that new multimedia messages have arrived. Dr. Larsen clicks on the mailbox button. In doing so, the MRD's `receive` window opens, see figure 4.

In the `receive` window panel, the messages are listed. The listing specifies the date, sender and subject for each message; unread messages are marked as "NEW". By selecting (i.e., single-clicking upon) a message in the listing, a logical representation of that message's contents is presented just above the panel. As in the `send` window, this representation is provided as a set of icons which reflect the kinds of information elements within the message.



Figure 4: The Snapshot Composer's receive window.

Dr. Larsen sees the new message from Dr. Eigil which asks for a quick reply. Having already selected the message, she opens it by pushing the "open message" button. This operation starts a process which "reconstructs" the message composed by Dr. Eigil. For each information element within the message, an application is started in order to present that element; furthermore, these applications are placed on Dr. Larsen's screen using the positions in effect when Dr. Eigil composed the message.

Since Dr. Eigil's multimedia message includes a text element, the same simple text editor is started in order to present that element. For the audio element recorded and sent within the multimedia message, the same audio application used by Dr. Eigil is started on Dr. Larsen's workstation. In a similar fashion, the `makeannot`⁴ application is started in order to present the annotation marks. The MRD-PACS application starts and sets its state to

be equivalent to the state it was in when Dr. Eigil, during composition of the message, clicked upon it.

When the multimedia message is fully opened, Dr. Larsen screen appears as shown in figure 3, *except for two differences*:

- Dr. Larsen has a `receive` window instead of a `send` window; and,
- she has the `makeannot` application running and visible at her workstation instead of the `asynccannot` application.

Dr. Larsen sees that Dr. Eigil hopes for an answer before 2:00 pm. Having time to deliver a reply by then, she listens to his ideas about the findings in image 1. While listening, she studies the image, especially the areas annotated by Dr. Eigil.

Dr. Larsen essentially agrees with Dr. Eigil's findings, but wants further confirmation. Since the MRD-PACS has already been started, she simply selects and displays another image within the same examination. Upon seeing this second image, she's convinced about the matter. The time is almost 2:00 pm, so she deems it best to try and reach Dr. Eigil right away. In order to try and initiate a real-time conference with Dr. Eigil, Dr. Larsen pushes the "telephone" button within the MRD.

5 Conclusions

With its mechanisms for real-time application sharing, telepointing and conference audio, the MRD's desktop conferencing facility helps eliminate barriers normally associated with geographically-distributed cooperative efforts. The MRD's Snapshot Composer also helps eliminate barriers; in this case, barriers primarily associated with temporal distribution. This multimedia messaging facility provides the equivalent of a desktop conference, except for the real-time feedback. In both of its communication modalities, the MRD allows:

- work within applications to be shared;
- pointers to be used for directing attention; and,
- questions to be asked and answered orally.

The MRD's Snapshot Composer presents a new, simple and intuitive way for users to construct multimedia messages. Use of standards ensures that integration of the Snapshot Composer within existing, UNIX/X based software environments can be achieved easily.

4) The `makeannot` application can only present annotations, it cannot create them. The `makeannot` application allows the user to hide (and then redisplay) annotations sent within a message.

Acknowledgements

The authors would like to recognize Pål Sørgaard, Elmer Sandvad and Kim Halskov Madsen for their creativity and hard work during the development and refinement of the MRD scenarios. The authors would also like to recognize the focal members of the pilot user group within the Department of Radiology, Rikshospitalet, Oslo: Søren Bakke, Bjarne Smevik, Hans Jørgen Smith and Andreas Abildgaard.

References

- [1] Aas, G., “Design of the Global Window Toolkit”, CODE-NR-93-17, Norsk Regnesentral, Oslo, December, 1993.
- [2] Aas, G., Holmes, P., Lovett, H., Møller-Pedersen, B., Sørgaard, P., EuroCODE deliverable: “D-5.1: CSCW Shell Requirements for the Middle Road Demonstrator”, CODE-NR- 92-8, Norsk Regnesentral, Oslo, December 30, 1992.
- [3] Braa, J., Sørgaard, P., Holmes, P., Mogensen, P., Kyng, M., Thüning, M., Robinson, S., Kreifelts, T., Mackay, W., EuroCODE deliverable: “D-1.2: Requirements for EuroCODE Systems”, March, 1, 1993.
- [4] Bødker, S., Christiansen, E., Grønæk, K., Madsen, K.H., Mogensen, P., Robinson, M., Kühn, H., Robinson, S., Thüning, M., Hinrichs, E., Sørgaard, P., Hennessy, P., EuroCODE deliverable: “D-1.1: The EuroCODE Conceptual Framework: Preliminary”, June, 17, 1993.
- [5] EuroCODE: CSCW Open Development Environment, ESPRIT Project 6155, Technical Annex.
- [6] EuroCODE Consortium (various authors from AU, GMD, NR, RXE, NX, ICL), EuroCODE deliverable D-2.5: “Design of the Toolkit Layer of the CSCW Shell”, CODE-AU-93-23, Aarhus University, Aarhus, Danmark, February 14, 1994.
- [7] Gritzman, M., Kluge, A., Lovett, H., “Task Orientation in User Interface Design”, in *Human Computer Interaction - Interact '95*, K. Nordby, P. Helmersen, D.J. Gilmore, S.A. Arnesen (eds.), Chapman and Hall, London, 1995, pp. 97-102.
- [8] Holmes, P., “Design of the Digitized Sound Toolkit”, CODE-NR-93-18, Norsk Regnesentral, Oslo, December, 1993.
- [9] Holmes, P., Lovett, H., Løbersli, F., “Evaluation of the Middle Road Demonstrator at Rikshospitalet”, CODE-NR-95-5, Norsk Regnesentral, Oslo, August 31, 1995. Also found in EuroCODE deliverable: “D-1.4.2: EuroCODE Demonstrator Evaluation Report”, EuroCODE Consortium, August 31, 1995.
- [10] Holmes, P., Lovett, H., Løbersli, F., Skorve, E.T., “The Middle Road Demonstrator: Concept, use, technical foundation and evaluation of a system for supporting communication within cooperative work settings”, NR Report 903, CODE-NR-95-10, Norsk Regnesentral, Oslo, December 20, 1995.

- [11] Holmes, P., Lovett, H., Møller-Pedersen, B., Sørgaard, P., Aas, G., Madsen, K.H., and Sandvad, E., “D-5.2: Design of the Middle Road Demonstrator”, CODE-NR-93-12, Norsk Regnesentral, Oslo, August 31, 1993.
- [12] Holmes, P., Sørgaard P., “Hypermedia, Desktop Conferencing and PACS”, EuroPACS Newsletter, Vol. 8, Nr. 1, 1995, (<http://www.nr.no/eurocode/papers/EUroPACS/>).
- [13] Løbersli, F., “Kommunikasjon og samarbeid omkring radiologiske bilder — Konsekvenser for distribuerte sanntids-konferensesystemer” (Communication and cooperation about radiological images — Implications for distributed real-time conferencing systems), Hovedfagsoppgave, Universitet i Oslo, Institutt for informatikk, 13 May 1994.
- [14] Peek, J.D., *MH & xmh — Email for Users and Programmers* (Third Edition), O’Reilly and Associates, Inc., Sebastopol, CA, 1992.
- [15] Sørgaard, P., “Design of the Conference Toolkit”, CODE-NR-93-16, Norsk Regnesentral, Oslo, December 9, 1993.

