Image Communications and Workstations

PROCEEDINGS
SPIE—The International Society for Optical Engineering

Image Communications and Workstations

Walter Bender
Mitsuo Saito
Chairs/Editors

12–13 February 1990
Santa Clara, California

Sponsored by
SPIE—The International Society for Optical Engineering
SPSE—The Society for Imaging Science and Technology

Cooperating Organization
TAGA—Technical Association of the Graphic Arts

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA

Volume 1258

SPIE (The Society of Photo-Optical Instrumentation Engineers) is a nonprofit society dedicated to advancing engineering and scientific applications of optical, electro-optical, and optoelectronic instrumentation, systems, and technology.

The papers appearing in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and are published as presented and without change, in the interests of timely dissemination. Their inclusion in this publication does not necessarily constitute endorsement by the editors or by SPIE.

Please use the following format to cite material from this book:
Author(s), "Title of Paper," *Image Communications and Workstations,* Walter Bender, Mitsuo Saito, Editors, Proc. SPIE 1258, page numbers (1990).

Library of Congress Catalog Card No. 90-52689
ISBN 0-8194-0305-9

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 206/676-3290 (Pacific Time) • Fax 206/647-1445

Printed in the United States of America

IMAGE COMMUNICATIONS AND WORKSTATIONS

Volume 1258

CONFERENCE COMMITTEE

Conference Chairs
Walter Bender, Media Laboratory/MIT
Mitsuo Saito, Toshiba Research and Development Center (Japan)

Cochairs
Kirk Martinez, University of London (UK)
Eric Walowit, Mead Imaging

Session Chairs
Session 1—Image Workstations
Walter Bender, Media Laboratory/MIT

Session 2—Office/Document Processing
Mitsuo Saito, Toshiba Research and Development Center (Japan)

Session 3—Image Analysis Systems
Walter Bender, Media Laboratory/MIT

Session 4—Desktop System Integration for the Graphic Arts
Eric Walowit, Mead Imaging

Conference 1258, *Image Communications and Workstations*, was part of a three-conference program on Image Systems and Applications held at the 1990 SPIE/SPSE Symposium on Electronic Imaging Science and Technology, 11-16 February 1990, in Santa Clara, California. The other conferences were:

Conference 1259, *Extracting Information from Complex Data: Processing, Display, Interaction*
Conference 1260, *Sensing and Reconstruction of Three-Dimensional Objects and Scenes.*

Program Chair: **Walter Bender,** Media Laboratory/MIT

IMAGE COMMUNICATIONS AND WORKSTATIONS

Volume 1258

INTRODUCTION

The focus of this conference was systems, as opposed to components. Toward this end, presentations focused on three areas: mechanisms of representing and manipulating multimedia data types, essential issues in the design of document and image segmentation, and mechanisms specific to the efficient interchange of color images. The presentations ranged from descriptions of innovations regarding basic technologies to descriptions of complete systems.

Highlights of the Image Workstations session include the detailed discussion of a multimedia environment in which an electronic mail system was developed. Several systems that incorporate motion video are also described in this session. One of these systems focuses on the sequencing of events at the application level, while the other addresses the issue of scalability of both temporal and spatial resolution.

The Office/Document Processing session focused on the essential issues in the design of shared document libraries, as well as on issues of document segmentation and analysis. A particularly interesting description of an all-digital scientific library was given in this session.

Finally, in Desktop System Integration for the Graphics Arts, presentations emphasized efficiency in the handling of color images, both in terms of compression and computation. In addition, a memory-efficient algorithm for doing scan conversion of page description languages was described.

Walter Bender
Media Laboratory/MIT

Mitsuo Saito
Toshiba Research and Development Center (Japan)

IMAGE COMMUNICATIONS AND WORKSTATIONS

Volume 1258

CONTENTS

(continued)

IMAGE COMMUNICATIONS AND WORKSTATIONS

Volume 1258

IMAGE COMMUNICATIONS AND WORKSTATIONS

Volume 1258

SESSION 1

Image Workstations

Chair
Walter Bender
Media Laboratory/MIT

Multi-media electronic mail

Bryan Yamamoto

NeXT, Inc.
Redwood City, California

ABSTRACT

Electronic mail is a facility, analogous to postal mail, in which computers are used to compose, deliver, and receive messages. Traditional electronic mail systems rely solely on text as the medium of communication. A multi-media electronic mail application, Mail, combines the media of Rich Text, voice, images, and electronic documents to facilitate interpersonal communication.

With Mail, these various media can be integrated into a single message. The variety of available media and the complexity of the messages that result from their combination make it important for Mail to have a simple user interface. It was possible to develop a simple, graphically-based interface that would accommodate Mail's message complexity.

The integration of these diverse media is made practical by the rich operating environment in which Mail runs. Modern advances in hardware, operating systems, libraries, and servers make possible this powerful multi-media electronic mail application.

1. INTRODUCTION

Electronic mail began on time-sharing computers, allowing users who were logged into their accounts to send other users messages. On 24-row by 80-column terminals, the only medium available for mail messages was text. It wasn't until the advent of the workstation, with high-resolution bitmapped displays on a local area network, that it became practical to mail anything more than text. In 1981, the Xerox Star had the first multi-media mail capability, with integrated text and graphics and a variety of fonts.[1] Multi-media mail applications on personal computers followed, some with fonts, some with graphics, and some with the ability to embed a file in the message. These mail applications tend to be less integrated, often requiring the user to quit the mail application to view the embedded document.

One of two approaches is usually taken for implementing multi-media document editors: either a tightly coupled or a loosely coupled approach.[2,3] The tightly coupled approach, taken by the Star, provides an editor for each of the different media. The advantage of this approach is that an integrated document editor results, so switching between the various media can be streamlined and intuitive. The disadvantage of this approach is that the combined code for each editor makes for a monolithic application, which in the software industry is equivalent to a software engineering nightmare.

The loosely coupled approach, most often taken by mail applications on personal computers, doesn't have an editor for each medium. Instead the data is imported from other applications, either through the cut and paste paradigm, or by the user specifying a file to be incorporated into the mail message. This approach results in an simpler mail program, while sacrificing the streamlined user interface and integration of the different media.

The NeXT electronic mail application, named simply Mail, uses a hybrid of these two implementation approaches. But before this approach can be described, some background about the application and its operating environment, as well as its user interface, is necessary.

2. BACKGROUND

The initial goal of NeXT Mail was to provide a window-based interface to electronic mail in NextStep®, the graphical user interface of the operating environment on the NeXT Computer.[4] Like traditional UNIX® mail readers, the first version of Mail allowed the user to compose, deliver, and receive messages, as well as maintain messages in different mailbox files. But because of the richness of the operating environment on the NeXT Computer, there was an opportunity to enrich the basic messaging capability.

NextStep consists of a window server, the Application Kit™, and a file manager called the Workspace Manager™ (Fig. 1). The window server uses PostScript® as its imaging model.[5] The PostScript language is extended to include operators that create and manage windows. However, applications in the NextStep environment aren't programmed in the PostScript language; they are programmed in the Objective-C® language and use the Application Kit to interface with the window server. The Application Kit is an object-oriented toolkit that provides a framework for applications. The Application Kit defines objects that perform the traditional window manager functions, as well as objects that provide user-interface controls, manipulate text and graphics on the screen and printer, and communicate with other applications and the Workspace Manager. The Workspace Manager is an application that can launch applications, as well as manage files.

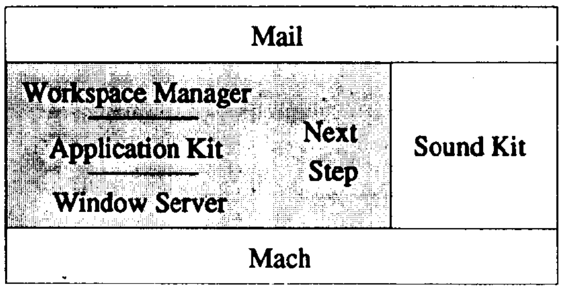

Fig. 1. Operating Environment

In addition to NextStep, the operating environment includes the Sound Kit™. Like the Application Kit, the Sound Kit is an object-oriented toolkit. It defines objects that manage the recording, playing, and editing of sound. Sound is recorded through a microphone connected to an analog-to-digital converter called the CODEC ("COder-DECorder"), at a sampling rate of approximately 8 KHz using an 8-bit mu-law encoding. The Sound object, a component of the Sound Kit, plays the 8 KHz signal by interpolating it into a 22 KHz signal using the Digital Signal Processor (DSP), and sending the resulting signal to the digital-to-analog converters. The Sound Kit also includes a SoundView object, an object that provides graphical viewing and editing of a recorded signal.

The operating system for Mail is Mach, a message-based kernel that is binary-compatible with UNIX 4.3BSD.[6] Mach is a multitasking operating system, capable of simultaneously running more than one application. Applications run independently of each other, and Mach gives each one its own copy of virtual memory and its own flow of control. Users typically leave Mail running in the background as they use other applications. Mach also provides a facility for applications to communicate with each other. Mail uses of this facility to employ other applications as editors for various types of media.

3. USER INTERFACE

The typical Mail user launches the application upon logging into the computer, and leaves the application running during the entire login session. When Mail detects that the user has some new mail, it retrieves the mail and places it in a mailbox file in the user's home directory. After retrieving the mail, the application performs an animation to signal the user that some new mail has arrived.

NextStep is a registered trademark of NeXT, Inc.
UNIX is a registered trademark of AT&T
Application Kit, Workspace Manager, and Sound Kit are trademarks of NeXT, Inc.
PostScript is a registered trademark of Adobe Systems Incorporated
Objective-C is a registered trademark of The Stepstone Corporation

3.1 Reading messages

A user's mail messages are read in the Mailbox Window, shown in Fig. 2. It consists of three parts: the command area, the summary area, and the message area. The command area has buttons that operate on the Mailbox Window, as well as some information about the *current message*, the message displayed in the message area. At the right of the command area is an image of the person who sent the current message. To the immediate left of the image is a graphical representation of the current message's date and time.

The summary area lists each of the messages in the mailbox. For each message, the summary gives the message number, the date it was sent, the sender's name, and the subject of the message. This area is like the table of contents of the mailbox file; it gives quick access to any message in the mailbox file. Clicking a summary line causes that message to be displayed in the message area, as well as causing the date of the message and the picture of the sender to be loaded into the command area. Multiple messages may be selected in the summary area so that cut, copy, and paste operations may be applied to those messages.

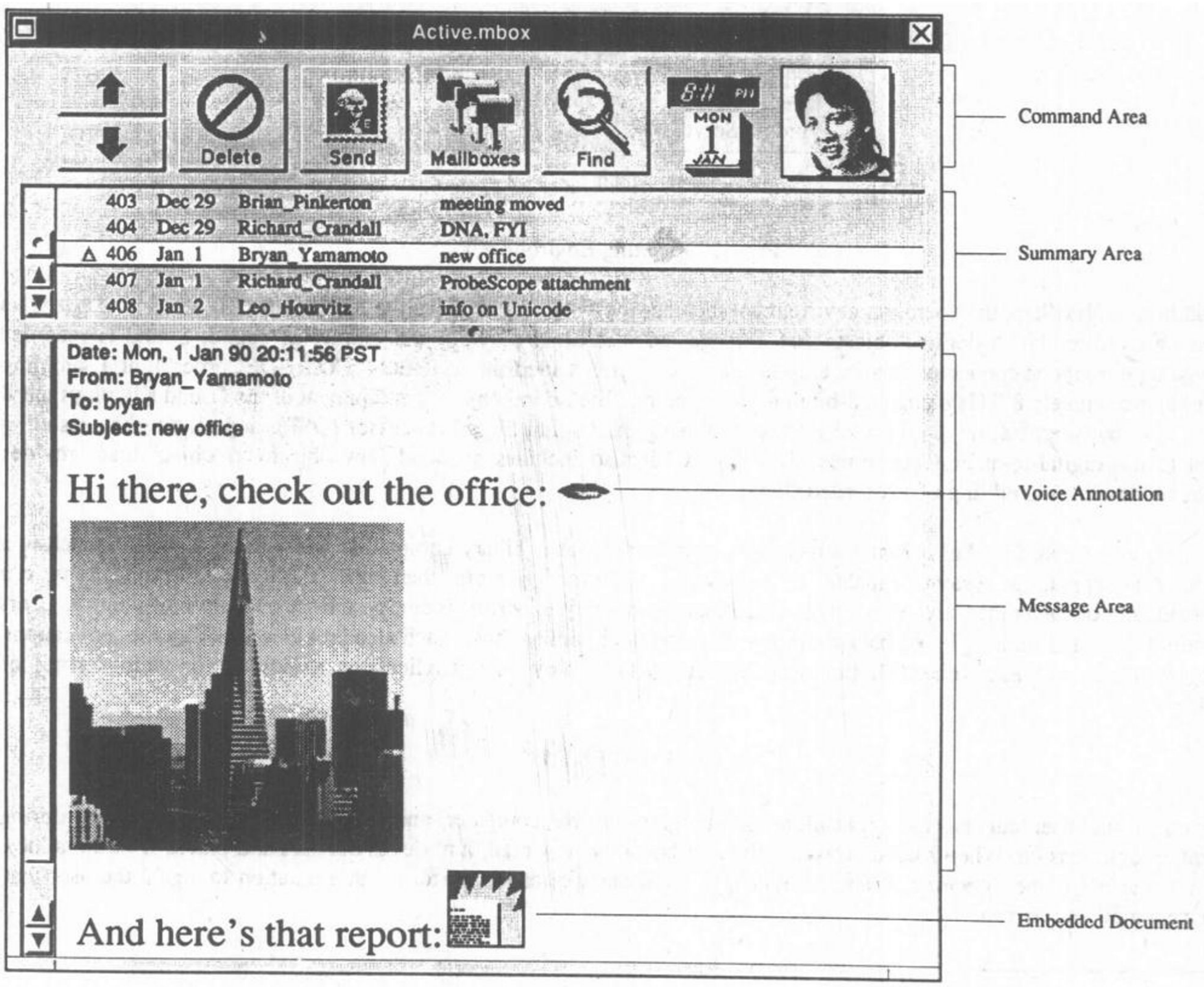

Fig. 2. Mailbox Window

Fig. 2 shows a message that contains text, a voice annotation, an image, and an embedded file. A lip icon represents a voice annotation. To listen to the voice annotation, the user double-clicks the lip icon and the voice editor, called Lip Service, appears (Fig. 3). Clicking the Play button in the Lip Service window plays the voice annotation. The embedded file is represented by its own icon, in this case a document from the WriteNow™ word processor. Double-clicking the icon causes the document to be opened by WriteNow. An embedded document can be either a file or a directory. Messages may contain any number of voice annotations, images, and embedded documents.

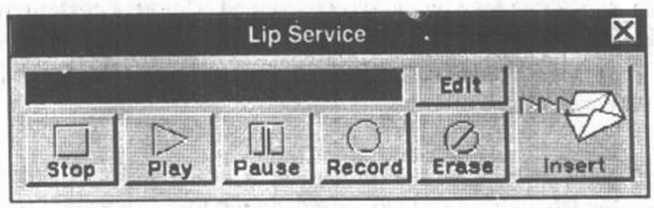

Fig. 3. Lip Service window

3.2 Sending messages

Messages are composed and delivered via the Send Window. Clicking the Send button in the command area of the Mailbox Window creates a new Send Window (Fig. 4). To send a message, a user fills in the *To*, *Subject*, and *cc* fields, types a message in the bottom of the window, and clicks the Deliver button. A message that has voice annotations, graphics, or embedded documents requires a bit more effort on the part of the user.

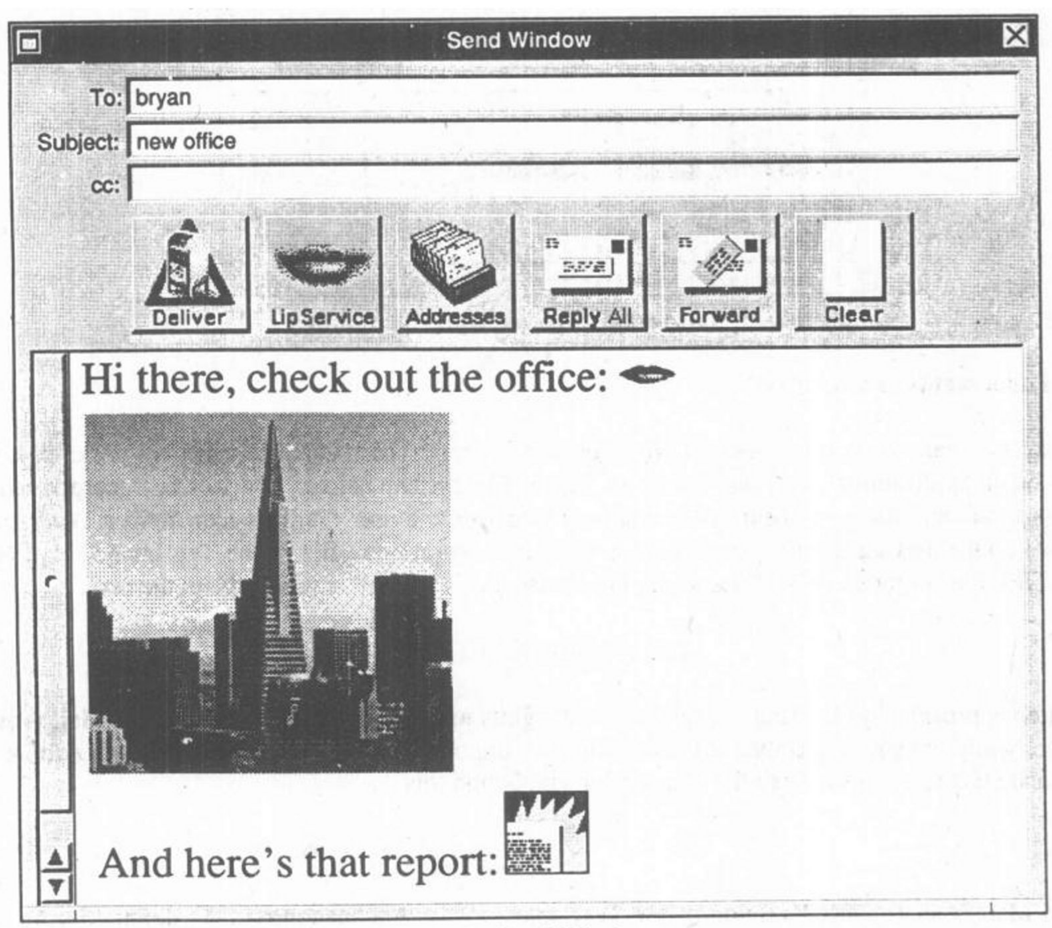

Fig. 4. Send Window

WriteNow is a trademark of T/Maker Company

3.2.1 Voice annotations

Voice annotations can be sprinkled throughout the mail message. A voice annotation is created using the Lip Service window (Fig. 3), which appears when the Lip Service button is clicked. The Lip Service window works much like a cassette tape recorder. Clicking the Record button starts the recording process, until the Stop or Pause button is clicked. The Play button plays back the message, which can also be paused or stopped. Unlike a cassette recorder, the recorded message can be edited graphically. Clicking the Edit button causes the Lip Service window to grow to include a graphical representation of the recording (Fig. 5). This representation is a graph of the recording's minimum and maximum amplitude over time. It allows the user to edit the recording.

Editing a sound recording is modeled after editing text with a window-based editor. Using the mouse, the user can select regions of the recording, which can then be cut, copied, and pasted. In the text editor, typing overwrites the selected text. In this sound editor, a recording overwrites the selected sound.

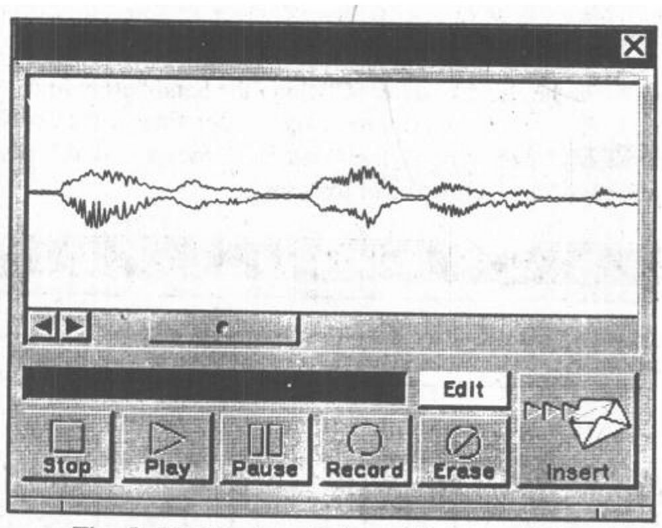

Fig. 5. Lip Service window with voice editing

3.2.2 Embedded documents and images

Messages can also be enhanced with images and embedded documents. These enhancements are not created in Mail, but are created with other applications and must be saved in the file system before they can be incorporated into a mail message. To incorporate an image or document into a mail message, the user drags its icon from a Workspace Manager directory window into the message in the Send Window. If the icon represents a file in the Tag Image File Format (TIFF), the image in the TIFF file is rendered in the message.[7] Otherwise, the icon itself is inserted into the text.

4. IMPLEMENTATION

The implementation approach used by Mail is a hybrid of the tightly and loosely coupled approaches. Mail provides editors for text and voice, while images and embedded documents are imported from other applications. As editors for text and voice, Mail uses and enhances objects from the Application and Sound kits.

4.1 The editors

The text editor in Mail is an enhanced version of the Text object in the Application Kit. As defined in the Kit, the Text object gives users the ability to enter and edit text, change fonts, and cut and paste text. Mail enhances this basic functionality using an object-oriented technique called subclassing, which allows code in Mail to supplement the Text object and, where appropriate, override some of the Text object's functionality. The enhancements mainly allow an arbitrary image in the text stream. Each image is made to appear as a single large character to the Text object. The Text object manipulates the image as it would a large character, hence the image can be cut and pasted along with other characters, it behaves like other characters when words are wrapped, and it moves in the text stream when the user types in front of it. The text is saved in the file system in Rich Text Format (RTF), which adds font and formatting directives to ASCII text.[8] RTF is an extensible format, and Mail extends it to include embedded files.

Lip Service, the voice editor, uses objects from the Sound Kit without modification. Lip Service consists of a Sound object, a SoundView object, and a SoundMeter object from the Sound Kit. Recording and playing of sound is handled by the Sound object, which is shared between the SoundView and the SoundMeter. The SoundView renders the recorded voice and implements the graphical editing of the signal. A VU meter, which gives the user feedback while recording and playing a message, is implemented by the SoundMeter. The Sound object stores recorded voice, which ultimately is embedded in the mail message in the same way that images and documents are embedded.

4.2 Embedded documents

Mail does not have editors for a message's embedded documents. However, viewing the documents is streamlined using interprocess communication (IPC). When the user double-clicks the icon for an embedded document, Mail sends a Mach message to the Workspace Manager requesting that the embedded file be opened. If the application that edits documents of that type is already running, the Workspace Manager sends it a message to open the document. If the application is not running, the Workspace Manager starts the application, and then sends it the message to open the document.

Dragging an icon into a message is also accomplished with messaging between Mail and the Workspace Manager. There is some setup required for this messaging to occur. First, Mail must register its Send Window with the Workspace Manager so that the Workspace Manager can detect when an icon has been dragged over a window that accepts an icon. Next, when a user drags an icon over a Send Window, the Workspace Manager notifies Mail that an icon has entered the window, and gives Mail the image of the icon and the name of the file the icon represents. Mail then copies the file and inserts the icon into the text of the message. It is not necessary for Mail to copy the image of the icon in addition to the file because the Workspace Manager can be queried about the icon for a given file and the icon can then be rendered.

4.3 Inter-application messaging

Messaging between Mail and the Workspace Manager is implemented by two objects in the Application Kit, the Listener object and the Speaker object. These objects implement the necessary messages for the Mail-Workspace Manager interaction using the Mach IPC facility. The Speaker object implements a set of messages that, when sent, are translated into a Mach message to a corresponding Listener object in another application.

The Application Kit provides a Listener object for every application so that the Workspace Manager can communicate with the application. In particular, an application that edits a document responds to the *openFile* message, a message that requests a document be opened. An incoming *openFile* message is dispatched by the Listener object to its delegate. The delegate is an object, set up by the application, that can implement and answer messages sent to the Listener object.

Applications that send messages to other applications do so with a Speaker object. The Speaker object facilitates inter-application messaging by providing the underlying communications support for remote messages. This involves establishing the communication, as well as generating the inter-application messages.

4.3 The file format

When a user double-clicks the icon for an embedded document, Mail requests that another application open the document. For another application to provide this service, it is necessary for the document to exist in the file system as a discrete file, rather than as a part of a larger file. Because of this, Mail uses UNIX file directories to store a message with images, documents, or voice attachments. The directory is given a unique name derived from the subject line of the message header.

Each image, document, and voice attachment is stored as a separate file in this directory. The directory also includes a file that holds the body of the message. Fig. 6 shows the contents of the directory for the message shown in Fig. 2.

Fig. 6. Mail message directory for message in Fig. 2.

The type of data in each file is identified by the file name extension – "rtf" for Rich Text Format, "wn" for WriteNow documents, "tiff" for Tag Image File Format, and "vox" for the voice annotation. The body of the message is stored in Rich Text Format, as shown in Figure 7. Mail enhances RTF to include a directive for an embedded file. The most important information in the directive is the name of the file, from which Mail can render an icon to represent the file in the text or find the file to open it. For TIFF files, Mail opens the file and renders the image in the text. Mail uses its own lip icon to represent voice annotation files. For all other file types, the Workspace Manager provides the icon images through interprocess communication.

```
{\rtf0\ansi{\fonttbl\f1\froman
Times;\f0\fswiss Helvetica;\f2\fmodern
Courier;}
\margl120
\margr120
{\f1\fs48\fi0\li0\ql\gray0 Hi there,
check out the office:}
{\f0\fs24\fi0\li0\ql\gray0  }
{\attachment32 VoiceMail_bryan0.vox}    ---- Voice annotation
{\f1\fs48\fi0\li0\ql\gray0 \
}
{\attachment33 trans.tiff}              ---- Image
{\f1\fs48\fi0\li0\ql\gray0 \
 And here's that report:}
{\attachment58 report.wn}               ---- Embedded document
{\f1\fs48\fi0\li0\ql\gray0 \
}
}
```

Fig. 7. RTF for a mail message

4.4 Message transmission

Mail uses the UNIX **sendmail** utility to deliver its messages.[9] This utility uses the simple mail transfer protocol (SMTP) to mail messages between sites.[10] However, SMTP only provides for sending 7-bit ASCII character-encoded messages in the protocol, and there is no provision for mailing multiple files or directories. To overcome these deficiencies, Mail uses the **tar** (tape archiver), **compress**, and **uuencode** UNIX utilities.[11] Used in concert, these utilities convert the message directory into a single file, apply data compression to that file, and then encode the result into 7-bit ASCII. The resulting ASCII is formatted into a mail message and given to the **sendmail** utility. The header of the mail message is tagged in a way that Mail can detect that it contains an encoded message. When Mail detects a tagged message as it reads new mail, it applies the **uudecode, uncompress,** and **tar** UNIX utilities to convert the encoded message into the original file format.

5. CONCLUSION

Using a hybrid of the loosely and tightly coupled implementation approaches works well for Mail. The benefits of the two approaches are gained while avoiding many of the pitfalls. Mail has a streamlined and intuitive user interface for embedded documents because of multitasking and interprocess communication. A monolithic application is avoided because the editors in Mail use objects provided in the various kits, rather than implementing each editor from scratch.

6. ACKNOWLEDGEMENTS

I would like to thank Don Larkin, Richard Crandall, Caroline Rose, Leo Hourvitz, Trey Matteson, Bertrand Serlet and Matt Morse for reviewing and editing this paper, and Bud Tribble for giving me a break from my normal duties so that I could write it.

7. REFERENCES

1. J. Johnson, T.L. Roberts, W. Verplank, D.C. Smith, C.H. Irby, M. Beard, and K. Mackey, "The Xerox Star: A Retrospective", *COMPUTER*, Vol 22, No. 9, pp. 11-29, September 1989.
2. H. Lison and T. Crowley, "Sight and Sound", UNIX Review, vol. 7 no. 10, pp. 76-86, October 1989.
3. N. Borenstein, C. Everhart, J. Rosenberg, and A. Stoller, "A Muiti-media Message System for Andew", *USENIX Association Winter Conference 1988 Proceedings*, pp. 37-42, February 1988.
4. NeXT, Inc., *The NeXT System Reference Manual*, Part 1: Concepts, Palo Alto, California, May 1989.
5. Adobe System Incorporated, *PostScript Language Reference Manual*, Addison-Wesley, Reading Massachusetts, 1985.
6. M. Accetta, R. Baron, W. Bolosky, D. Golub, A. Tevanian, and M. Young, "Mach: A New Kernel Foundation fo UNIX Development", *Proceedings of Summer USENIX*, July 1986.
7. Aldus Corporation and Microsoft Corporation, "Tag Image File Format Specification, Revision 5.0", *An Aldus/Microsoft Technical Memorandum*, August 1988.
8. Microsoft Corporation, "Rich Text Format Specification", 1987.
9. E. Allman, "SENDMAIL – An Internetwork Mail Router", *Unix System Manager's Manual*, 4.3 BSD Virtual VAX-11 Version, ch. 16, AT&T Bell Laboratories, Inc., modified by the University of California, Berkeley, April 1986.
10. J. B. Postel, "Simple Mail Transfer Protocol; RFC 821", *Internet Working Group, Request For Comments, No. 821*, Network Information Center, SRI International, Menlo Park, California, June 1982.
11. *UNIX User's Reference Manual*, 4.3 BSD Virtual VAX-11 Version, AT&T Bell Laboratories, Inc., modified by the University of California, Berkeley, April 1986.

Digital movies for "off-the-shelf" displays.*

Walter Bender and Robert Mollitor

Massachusetts Institute of Technology, Media Laboratory
20 Ames Street, Cambridge, Massachusetts 02139

ABSTRACT

There are a variety of emerging technologies which will facilitate the incorporation of video into the work station environment. These technologies, whether analog or digital, very often require the addition of special purpose hardware to be added to the work station or personal computer. We have been investigating algorithms for the display and integration of video image sequences into a PC using an off-the-shelf graphics adapter. The techniques we employ are not as powerful as those taking advantage of special purpose hardware. In particular, no provisions are made for either real-time capture or full screen motion. However, by providing the simultaneous display of multiple, scalable, moving movies, "multi-media" applications which use preprocessed, window-sized video sequences can be designed.

In this paper we describe both techniques for image reduction and coding and various strategies for decoding, playing, and smoothly moving video image sequences.

1. INTRODUCTION

A plethora of multi-media systems are emerging which enhance the work station by the incorporation of video. The systems which typified the 1970s, married analog video with digital displays. Many of these systems used two or more displays: one for digitally generated text and graphics, the others for video retrieved from external analog storage devices such as video tape recorders or optical video disc. Other systems used special effects hardware to overlay the graphics onto the video.[1] In the 1980s, the availability of inexpensive "picture-in-picture" technology has lead to the dissemination of window systems with analog video inserts. Although the utility of these multi-media systems is still subject to conjecture, they do provide addition degrees of freedom to the application designer. One can anticipate that we will learn to put windowed motion displays to productive application.

We are looking at the underlying technology of multi-media systems and how it impacts the potential applications. The goal of the technology is to provide tailorable mechanisms to incorporate video. The system variables include cost, quality, compatibility, and flexibility. Ideally, a technology will provide a maximum degree of freedom to the "software" provider and consumer at whatever level of cost and quality is desired.

1.1 Analog video

There are various problems with the application of analog video to multi-media systems, several of which are enumerated here. (1) There are limited means of distribution. Analog video is precluded from many digital distribution channels, such as computer local area networks and digital storage media. (2) The nature of the fixed raster format of analog signals makes the signal difficult to scale and puts an upper bound on image quality. (3) The integration of video into applications requires special hardware. (4) Access to the video material is indirect and limited. With the exception of the optical video disc, random access is forestalled.

Some of these problems are not inherent in analog systems. For instance, scalable, scan-rate independent, analog television systems have been proposed in the past. However, commercially available analog systems of today are married to the fixed format, and industry has neither the power nor the interest to redefine it.

*This work was supported in part by International Business Machines, Inc.