MUI: A Window Based User Interface for Multi-Media Mail

Stephen A. Uhler
Computer Systems Research Division
Bellcore

ABSTRACT

MUI is a prototype Multimedia Mail user interface that enhances ordinary text based mail by providing multi-media capability. It operates in concert with the existing UNIX mail system to provide an easy to use multi-media electronic mail interface whose messages are compatible with the existing internet mail transport facilities. To demonstrate interoperability across different user interface platforms, MUI runs on the MGR window system for composing mail, and on either the X or MGR window systems for viewing the mail.

Introduction

Powerful workstations with sophisticated graphical user interfaces (GUI's) have replaced text-only ASCII terminals on desk-tops. With desk-top computers connected by faster, more reliable networks than ever before, computer users are dependent on electronic mail (email) as a vital communications medium. However email still consists primarily of ordinary ASCII text. Multimedia email - the ability to send more than just text as electronic mail - provides the opportunity to enhance the value of email, and upgrade its functionality to be more in tune with the capabilities of modern workstations. An easy to use user interface is critical to the success of multimedia email. If multi-media email is too hard to use, or disrupts the users computing environment, then email risks remaining just plain old text.

MUI (**M**ulti-media-mail **U**ser **I**nterface) is a prototype user interface for multi-media email that Runs on Sun SPARCstations. It permits users to create, edit, and view multi-media mail, while functioning as a tested for experimenting with new media types. It is compatible with emerging internet standards for transmitting multi-media mail, and works with existing text-only mailers. Its design enables the installation and evaluation of new types of multi-media and user interfaces without disrupting the user's current mail environment.

Design Criteria

The most important consideration involved in the design of MUI is compatibility: compatibility with the existing mail system, compatibility with the existing mail user interface, and compatibility with the overall look and feel of the workstation user interface. Compatibility is maximized to increase the likelihood that people will use the system to create mail, and that recipients of mail will be able to read it.

UNIX is a registered trademark of UNIX Systems Laboratories, Inc.

SPARCstation is a trademark of Sun Microsystems, Inc.

MUI is compatible with existing mail systems. The messages created by MUI conform to the existing internet mail transport protocols[*1*], so existing mail transport agents and mail gateways can handle multi-media mail without being modified. Text-only mailers can still read the text portions of mail produced by MUI. The multi-media or non-text portions of the message, although not viewable, won't break existing mail systems. The existing mail user interface is retained; users can use MUI for text-only mail without being aware they are using multi-media mail, as the Multi-media features are presented to the user unobtrusively. These multi-media features are consistent with the overall look and feel of the workstation user interface.

Because MUI interacts in a natural way with existing programs that manipulate various multi-media objects, it is easy to incorporate those objects into a mail message. Similarly, MUI is easy to learn, because it incorporates the existing mail user interface.

Architecture and Implementation

To maximize compatibility with existing mailers, and to keep MUI easy to use, MUI is implemented as a suite of separate programs or *components* that work in conjunction with the existing text-only mail system. It presents a uniform GUI for composing, sending, and viewing multi-media electronic mail. Figures 1 and 2 show the components of MUI, for sending and receiving mail respectively.

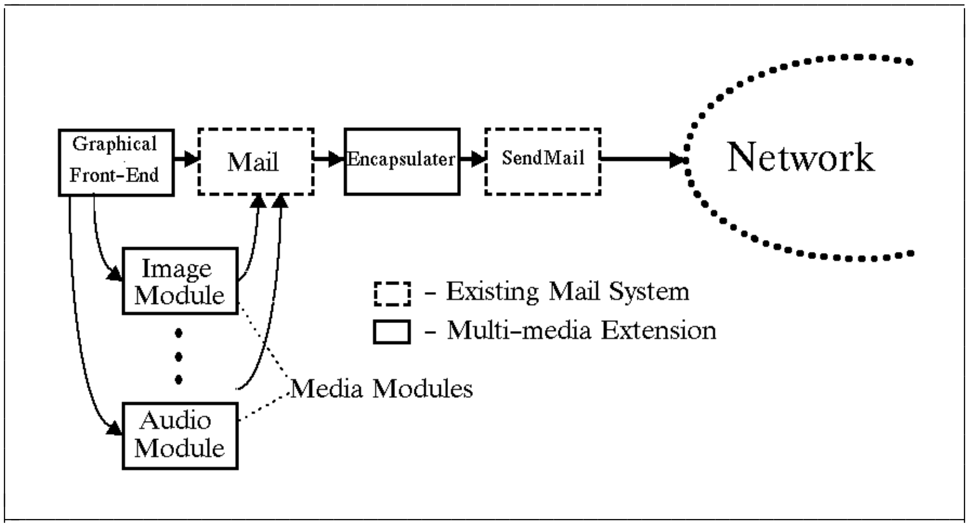


Figure 1: MUI Components for Sending Multi-media Mail

The components are: a graphical front end, media interface modules, an ordinary UNIX mail program, a mail encapsulator, a media identification module, and a mail delivery agent.

The first module is the graphical front end to the mail system. It provides a mouse click and pop-up menu style user interface for mail interaction, generating commands that an ordinary mailer expects. The menus allow the user to select among various media types to be included into the current mail message.

The second module is the normal mail system. The current prototype uses Berkeley Mail[*2*], chosen because it is readily available on most UNIX workstations. Consequently, most ordinary text mail users are already familiar with its user interface which, as mentioned above, is retained for ordinary text based mail interactions. Berkeley Mail is capable of running a program and including its output as part of the current mail message, simplifying the interface to MUI.

The *media modules* each provide a specific media capability for the mail system. Each is implemented as a separate program. The media modules provide the user interface for composing

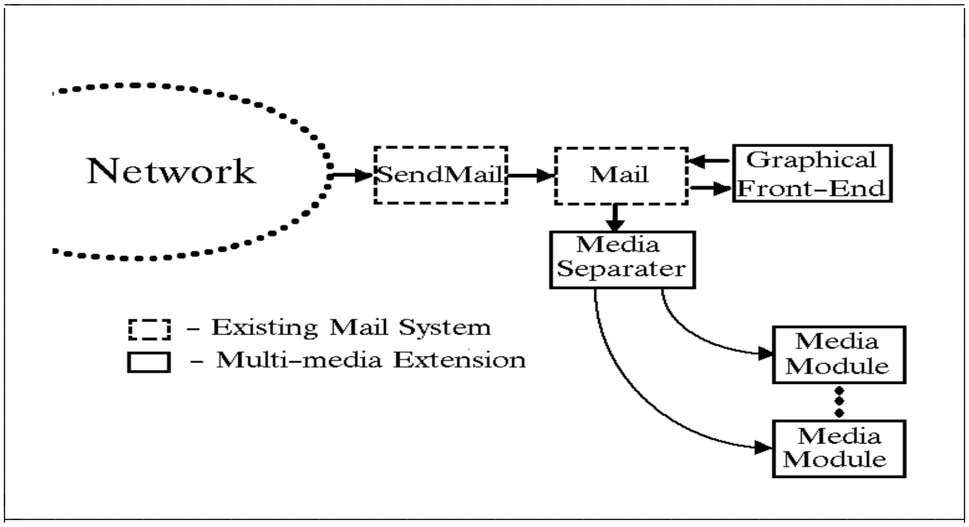


Figure 2: MUI Components for Receiving Multi-media Mail

and editing their particular type of media. Currently there are three media modules, one for including images into a message, one for including audio and another for generating annotations (a new type of media that will be described later). These three media modules are written either in shell or C, and range from 20 to 300 lines of code. For viewing mail, a different set of modules is used, one for each of the three media types.

The *encapsulator* is a post processor that inspects completed outgoing mail messages. If the message contains any multi-media components, the encapsulator adds a line to the mail header to identify the message as *multipart* as specified by the proposed Internet multi-media mail standard[3]. If the mail message contains no multi-media parts, the messages goes out as ordinary text mail. In addition, sine every message passes through the *encapsulator,* statistics about the type and frequency of multi-media mail usage can be gathered.

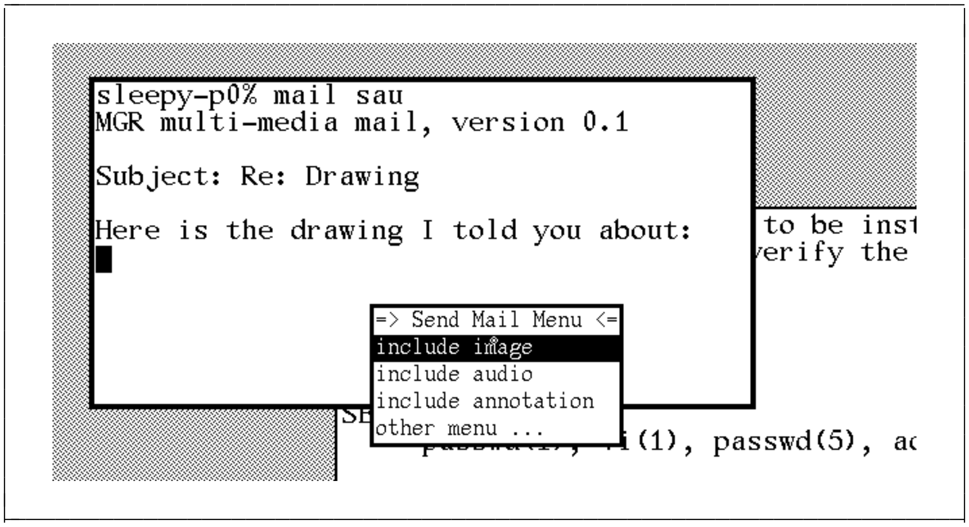
The *media identification module* scans the incoming mail. If any multi-media parts are found in a message, the type of the media is identified from its header, the appropriate media handler is located in a database, and the corresponding media viewer is invoked. MUI uses *metamail[4]* to perform the media identification function.

The final component, the *Mail delivery agent*, is responsible for taking the completed mail message and delivering it to the proper recipients. Since the multi-media mail message format conforms to the internet standard, any internet mail delivery program will do. The prototype version of MUI uses *Sendmail* [5], which is the default mail delivery program on many UNIX workstations.

User Interface Description

The user composes mail with MUI using the keyboard, mouse, a menu and icons. The keyboard is used to enter ordinary text mail, the mouse is used to either select items from the menu, or click on icons. The menu lists the available built-in media modules, while icons represent independently developed programs (enhanced to communicate with MUI) that manipulate their particular media types.

Figure 3 is an example of a MUI menu, activated by depressing a mouse button. If the first menu item, *include image* is selected, the user is prompted to click the mouse on any window currently on the display. The contents of that window are automatically included in the message as an image. The next menu item, *include audio* prompts the user to turn on the microphone,



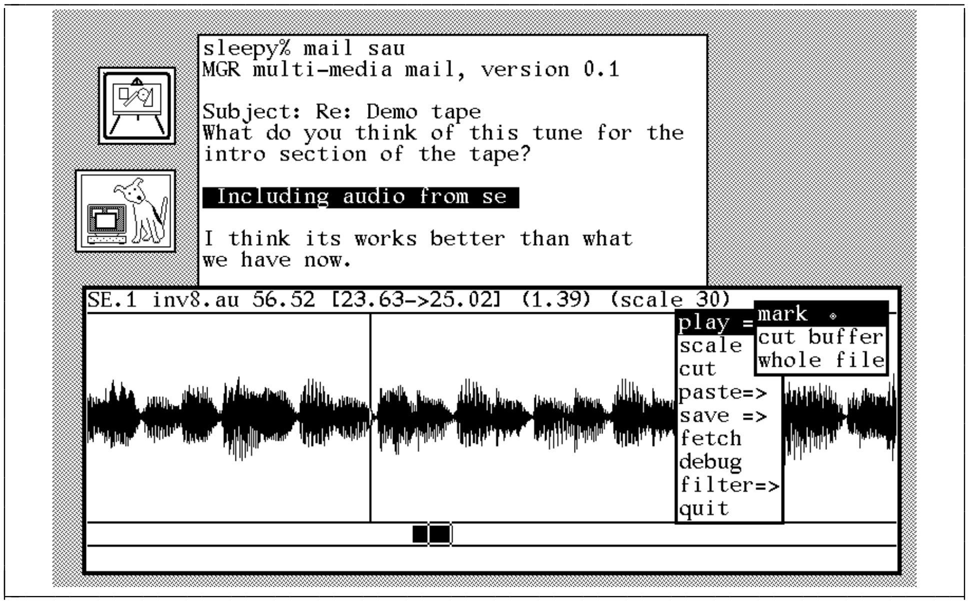
```
sleepy-p0% mail sau
MGR multi-media mail, version 0.1

Subject: Re: Drawing

Here is the drawing I told you about:
```

Figure 3: Sample MUI Menu (From a Portion of the Workstation Display)

and key return to begin recording a message. When the user completes the recording by pressing return, the sound is then included into the mail message. The final menu item, *include annotation*, represents an experimental media type, an annotation, which combines an image, audio, and real time annotation. When *annotation* is selected, the user is prompted to select a window on the display. The selected window's content is frozen, and the microphone is turned on. The user may then draw on the window using the mouse while speaking into the microphone. Several different brush strokes are available, such as thin lines, thick lines, or dotted lines. After completing the annotation, the window being annotated is returned to normal, and the *annotation* is included into the current message. When the *annotation* is viewed by the mail recipient, the window image appears and the audio plays while the brush strokes are drawn, recreating the annotation session in real time. Annotations have proven to be useful for marking corrections on documents and diagrams quickly, where editing the actual document is sufficiently time consuming that concentration is lost.

Figure 4 shows two icons. They represent (from the top) the image editor, *snap* and the sound editor, *se* respectively. The window at the bottom is *se*, which is being used to edit a sound fragment for inclusion into the message. Icons are used as an alternate method for including multi-media pieces into messages, and typically have more sophisticated interfaces than their menu activated cousins. The icons represent application programs, or stand-alone tools that function independently from the mail system. If the user wishes to include a type of media that one of these tools manipulates, the user clicks on the icon from within the mail system. That activates the tool application, which sends the completed media object back to the mailer when the user indicates editing is complete. The program *se* in Figure 4 was activated by clicking in its icon from within MUI.

The image editor, *snap*, is represented by an easel icon. When the user clicks on the easel, *snap* wakes up, permitting the user to capture and edit an image. Snap allows the user to cut, paste, and edit images from multiple sources. When the user completes the image, *snap* automatically includes it in the mail message. Should the user click on the *se* icon while composing mail, the sound editor is started. The user then interacts with the sound editor in the usual way, cutting pasting, and editing sounds. When the editing is complete, MUI resumes operation with the sound fragment included as part of the mail message.



```
sleepy% mail sau
MGR multi-media mail, version 0.1

Subject: Re: Demo tape
What do you think of this tune for the
intro section of the tape?

 Including audio from se

I think its works better than what
we have now.
```

```
SE.1 inv8.au 56.52 [23.63->25.02] (1.39) (scale 30)
```

Figure 4: Sample MUI Icons and *se* (From a Portion of the Workstation Display)

Conclusions

MUI currently consists of about 1500 lines of C and shell code, not including the icon based services such as *snap* or *se*. The current implementation operates in conjunction with the MGR[*6*] window system for creating and viewing mail. In addition, to demonstrate interoperability among different GUI platforms, as well as to encourage its use, all of the media types currently created by MUI may be viewed using the X[*7*] window system as well as MGR.

Multimedia mail is easy to create: the images, sounds, and annotations can be freely intermixed with text in any mail message. The resulting mail message is compatible with existing mailers on the internet, so users of MUI can exchange multi-media mail wherever regular email is possible. The technique of using separate *media modules* to implement different media types is flexible, and new media types, such as annotations, are easy to integrate into this framework. Future plans call for the deployment of MUI, and the evaluation and testing of the various *media module* interfaces. MUI is still an evolving system, and the implementation of new capabilities, such as the integration of facsimile and telephony into multi-media mail is underway.

References

[*1*] Crocker, D. H. *RFC 822, Standard for the Format of ARPA Internet Text Messages* August, 1982.

[*2*] Shoens, K. *The Mail Reference Manual* 1980, Berkeley, Ca.

[*3*] Borenstein, N. S., Freed, N. *Mechanisms for specifying and Describing the Format of Internet Message Bodies* Proposed internet RFC, not yet published, August, 1991.

[*4*] Borenstein, N. S. *Adding Diverse Multimedia Format Support to Established RFC822 Mail Readers* Bellcore Internal Technical Memorandum TM ARH-018430, January, 1991.

[*5*] Allman, E *Sendmail - An Internetwork Mail Router* Berkeley Ca., 1983.

[*6*] Uhler, S.A., *MGR - C Language Application Interface,* Bellcore Internal Technical Memorandum TM-ARH-010796, December, 1988.

[*7*] Scheifler, R. W. Gettys J., *The X Window System* ACM Transactions on Graphics 5(2), p79-109, April 1986.