

Mail Facilities in a Multimedia Environment

Providing a multimedia email facility required that the well-established processes of creating, sending, receiving, printing, and replying to email messages be maintained and applied to messages containing multimedia objects.

by Robert B. Williams, Harry K. Phinney, and Kenneth L. Steege

The advent of tools and capabilities that allow users to manipulate and create multimedia objects on a workstation mandated the need to make it possible to send these objects through electronic mail, or email. The user interface for creating, reading, and sending text messages through email is well-established. For multimedia email to be effective the same sort of process flow must be in place. For example, just as a user can use the more command to view a standard text email message, an equivalent facility must be available to view a multimedia email message. What this implies is that the user should not have to be concerned with invoking the correct software to deal with a particular media type because this should be handled by the mail facility.

The HP MPower mail facility, which is represented by the envelope icon on the HP MPower front panel, provides support for sending, replying, viewing, and printing of multimedia mail.

For sending multimedia email, HP MPower provides two approaches: dragging and dropping a file on the envelope icon or clicking on the envelope icon. With the drag-and-drop method, the user is presented with the dialog box shown in Fig. 1a. From this box the user can either send the dropped file to its destination by selecting the Send button or edit the file by selecting the Edit button. If the Edit button is selected, the mail composer (editor) window is displayed showing the contents of the file that was dropped on the mail icon (see Fig. 1b).

To read mail the user can click on the mail icon and be presented with the two windows shown in Fig. 2. The first window contains the standard elm screen and the other contains the HP MPower viewer screen. Elm, which is the HP-UX* screen-oriented electronic mail processing system, provides the user interface for the user to interact with the HP MPower mail system. To edit or create a multimedia mail message, the user can access the HP MPower mail editor (composer) by selecting Mail Msg from the elm screen. This will provide the screen shown in Fig. 1b.

To read or view a mail message, the user would select one of the messages in the message list and then press Read Msg menu item from the elm screen shown in Fig. 2. The selected mail message will appear in the HP MPower viewer window for reading.

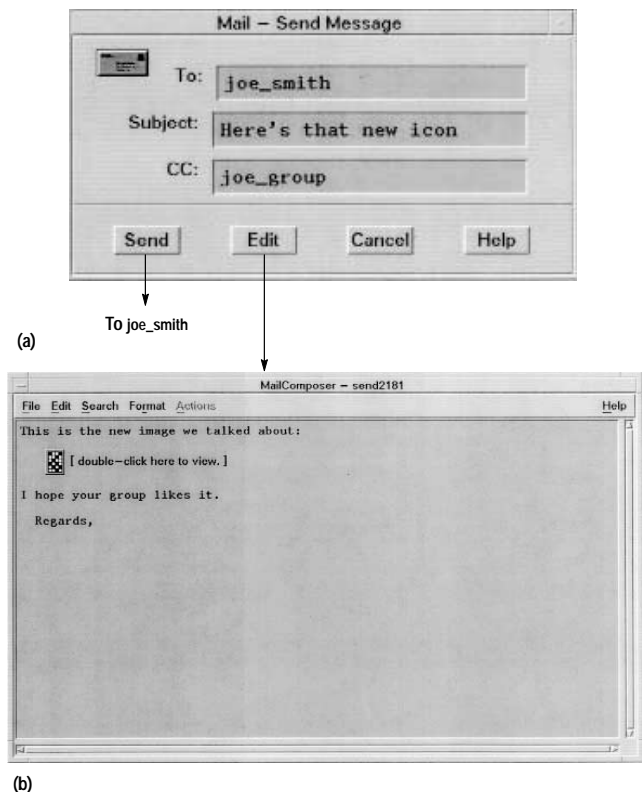
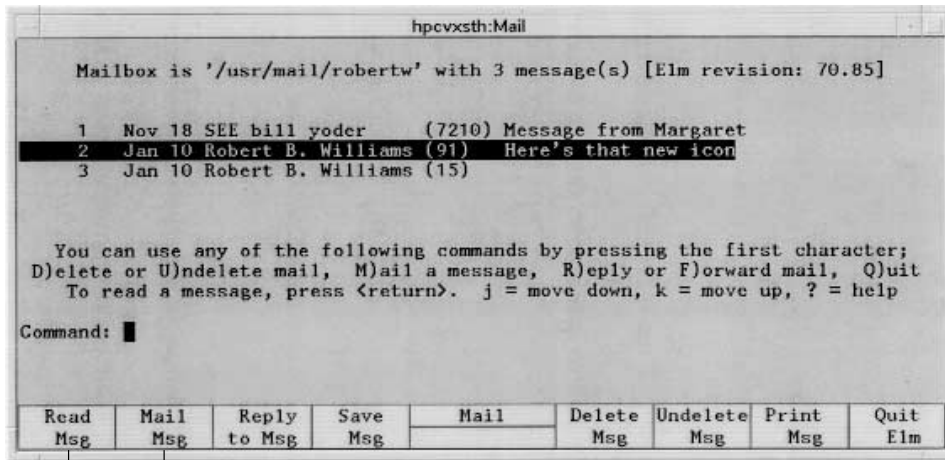


Fig. 1. (a) The dialog box that appears after a mail message is dragged and dropped on the front-panel mail icon. Selecting the Send button will send the message to its destination. (b) Selecting the Edit button produces the mail composer screen.

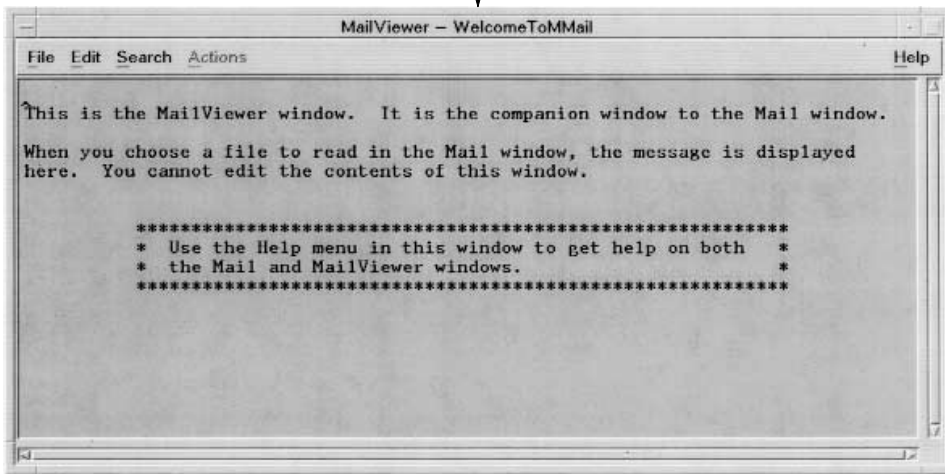
HP MPower Mail System Components

The main components of HP MPower mail include the HP-UX elm mail user agent, a multimedia editor, several shell scripts, a standard multimedia file format and supporting software, and HP VUE actions and file types. Fig. 3 shows a simplified diagram of some of the main HP MPower mail components responsible for providing the user interface actions described above.

With the exception of vuemime, which among other things handles the encoding and decoding of multimedia data, the



(a) MailEditor



(b)

Fig. 2. The windows produced by just clicking the mail icon. The top window contains elm and the lower window contains the HP MPower viewer.

components shown in Fig. 3 that send or receive data require no specific knowledge about dealing with multimedia data. For example, when a media-rich file is sent to `sendmail`, the file is treated like any other file going onto the network. Another example is when the composer or viewer needs to render a multimedia object (e.g., play an audio file or draw an image), the media icon is mapped to the process (or actions) capable of handling the particular media object by the HP VUE action database.

Vuepad. This is an enhanced version of the HP VUE editor. It provides two modes of operation in the HP MPower mail system: composing (creating a multimedia mail message to send) and viewing (reading a multimedia mail message). `Vuepad` is described in detail later in this article.

HP VUE Action Database. When a particular file type is selected the HP VUE action database provides the mechanism for invoking the appropriate action associated with that file type. For example, when the user double-clicks on an icon representing an audio file, the HP VUE action database tells `vuepad` that the audio player should be invoked to play the contents of the file.

MIME, Vuemime, and Metamail. To handle the interchange and storage of different types of data, HP MPower uses an internet standard known as MIME (Multipurpose Internet Mail Extensions). MIME is an extension to the basic internet mail standard RFC 822, which specifies the format for internet text messages. MIME defines the format for multipart, multimedia mail messages. `Vuemime` and `metamail` are utilities that provide support to HP MPower for MIME data. `Vuemime` provides a single interface for HP MPower components using MIME data, and `metamail` is a public-domain utility that we modified to act as a MIME filter. In this role `metamail` decodes and flattens MIME messages from non-HP MPower sources and encodes messages going to the mail transport agent `sendmail`. `Vuemime` is the only HP MPower component that communicates directly with `metamail`. MIME, `vuemime`, and `metamail` are described in more detail later in this article.

Mail User Agents. HP MPower has two mail user agents, `elm` and a shell script called `mmsend`. `Elm` is the main user agent, providing the usual `elm` capabilities of viewing, printing, saving, or deleting incoming or stored mail, replying or forwarding mail, and originating mail. `Elm` is the standard mail

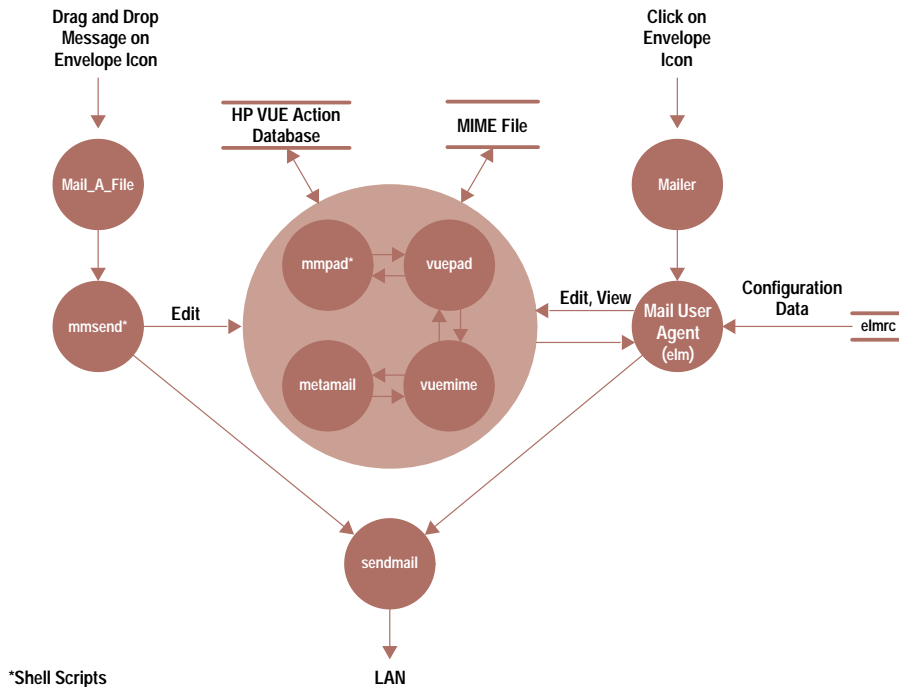


Fig. 3. Some of the main components contained in the HP MPower mail facility.

processor that is shipped with every HP workstation. This greatly reduces the support load and avoids duplication of development effort. Unfortunately it also put us in the position of offering a keyboard-based mail agent amidst a large collection of applications with graphical interfaces. Access to `elm` is through clicking on the mail icon, producing the screens shown in Fig. 2.

One of the obvious changes to `elm`'s normal interaction model is the tools used to view and edit mail. In the HP MPower environment, since we have to deal with mail that may contain nontextual items, the multimedia composer (editor) and viewer in `vuepad` are used to edit and view mail. This is done by modifying the user's `elrc` (configuration) file so that instead of invoking `vi` or `more` for editing or viewing a file, HP MPower's editor or viewer are invoked. Most other user customizations of `elm` behavior in the `elrc` file are preserved.

The second mail user agent, `mmsend`, provides an interface for drag and drop mailing of multimedia objects. Dropping a file or files on the mail icon causes the software to invoke an HP VUE action that indirectly calls the `mmsend` shell script. `Mmsend` drives the processes that are responsible for displaying and interacting with the screens shown in Fig. 1. The `mmsend` script also invokes the processes that convert multimedia files into MIME format. When the user selects the Send button from the mail dialog box shown in Fig. 1, `mmsend` invokes the mail transfer agent `sendmail` to dispatch the message over the network.

Mail Transfer Agent. The primary mail transfer agent in the current version of HP MPower is `sendmail`, the HP-UX facility for sending mail over the internet. Although there are gateways from `sendmail` SMTP (Simple Mail Transport Protocol) to X.400 and OpenMail, these gateways are not supported because they have not been enhanced to understand MIME on the SMTP (outbound) side.

Instant ignition systems (see page 17) usually do not have `sendmail` enabled. To overcome this the HP MPower installation scripts set up a default `sendmail.fc` and alias database. This is only done if `sendmail` is not already enabled. It does not address `sendmail` connectivity issues in complex or restricted environments. Ideally, `sendmail` would be turned off on mini-client systems because the mail agent runs on the server with the rest of HP VUE.

Printing. Multimedia printing involves decomposing the MIME formatted file into its parts and invoking the HP MPower print action (HP SharedPrint) on each of the parts. HP SharedPrint is described in the article on page 44. Invoking HP SharedPrint for mail is handled by the shell script `mmprint`.

Several types of multimedia cannot be printed (e.g., audio files). This is handled by a print action for the unprintable file that maps to the special action NONE.

Since printing actions happen on the client (HP SharedPrint forces this), `vuemime` and its database `mimetypes` must exist on the client as well as the server. This has implications for installations adding multimedia types.

Multimedia Editor

An important aspect of a multimedia mail facility is the ability to compose and view documents containing both text and nontextual items. In the HP MPower environment this ability is provided by a specially evolved version of the HP VUE text editor, `vuepad`.† All of the multimedia editing and viewing facilities available in HP MPower are built on this new version of `vuepad`.

The development of this component was constrained by many different factors. The schedule was set to allow the

† Unless stated otherwise references to `vuepad` in the rest of this article will be referring to the new version of the program.

introduction of the product to coincide with the release of new workstation computers. This ensured good public exposure for the product, but limited the amount of time and the number of engineers available for product development. The schedule and staffing level provided a strong motivation for using the functionality of existing products rather than wholesale development of new components.

Vuepad Modes. The `vuepad` program provides two modes. One is the HP MPower mail composer (mentioned previously) for creating and editing multimedia documents. The other mode is the HP MPower mail viewer (also mentioned previously) for viewing or listening to multimedia documents. The viewer is simply a read-only version of the composer. These facilities are accessible from the HP MPower mail icon or the HP VUE panel edit icon. The activation of either of these two modes is determined by the arguments passed to the `vuepad` program when it is invoked from the `mmpad` script shown in Fig. 3.

The HP MPower mail composer takes files created by the various media editing tools such as the HP SharedX Whiteboard or the audio editor and allows the user to incorporate the files generated by these tools into a document containing text. The final output of a message created by the composer is in a format compatible with the MIME standard. This approach required no modifications to any of the media editing tools and no changes to the core email application (`elm`). Use of the MIME message format helps provide some interoperability with other mail systems and gives us access to an already well-documented and carefully defined structure for messages.

Fig. 1 shows that the `vuepad` editor provides the composer and viewer with a visually appealing “iconic” view of nontextual data. The `vuepad` editor ensures that a user’s normal editing actions function as the user expects in the presence of nontextual items. It also ensures that the operations available for manipulating nontextual items are clearly visible.

Modifications to Vuepad. The required modifications to the old `vuepad` editor included adding the ability to invoke a filtering program for reading and writing multimedia data. The choice to do this filtering in a separate program was primarily driven by the need to develop the filtering functionality in parallel with the enhanced editor, and a desire to use different development programming languages for the editor and filter.

The filter is the `vuemime` program mentioned earlier. When `vuepad` is reading a multimedia file, `vuemime` strips out the media data, writing each media object to a separate temporary file. The media data is replaced within the original data stream by a special character sequence indicating that a media object existed in that location. This character sequence, known as a tag, contains the name of the temporary file that holds the media data. This file is used by `vuepad` and the HP VUE file-typing facilities to determine the graphical icon to display in place of the media object. When `vuepad` writes to a multimedia file, `vuemime` inspects the file for any tags and replaces the tags with the actual media data and the necessary MIME header information.

It was necessary to augment or override some of the internal functions of the OSF/Motif text widget that `vuepad` is based on. Before the widget draws a line of text, `vuepad` checks to see if the text corresponds to part of a media icon.

If it does, then `vuepad` draws the necessary portion of the icon. If the text does not contain part of an icon, then the text widget is allowed to render the whole line of text. Whenever the widget writes or reads data to or from either the OSF/Motif clipboard or the X primary selection window (a clipboard that can hold one item at a time), `vuepad` runs the data through the `vuemime` filter program to reinsert or strip any embedded media data as required. When the user selects a portion of text for an editing operation, `vuepad` tracks the selected region to provide meaningful highlighting of any selected media icons. `Vuepad` also observes all deletions and additions to the document to accurately track the position of the media objects within the document. Special care is also taken during the spell checking and formatting operations. The spell checking code must exclude the rather cryptic media tag data from the text sent to the spell checking program, and the formatting code has to do its work while preserving the relative locations of the media icons.

Compatibility. The behavior of `vuepad` is controlled by X Window System resources and command line options that allow `vuepad` to behave identically to the previous nonmedia-enhanced version of the HP VUE editor. The HP VUE file typing† and action database mechanisms were used to speed development and to provide consistency with HP VUE’s file manager appearance. The HP VUE action database provides a simple means for invoking an appropriate action associated with any particular file type. Use of the action database in `vuepad` ensures consistency with the default action accessed from the file manager for a particular file type.

Composer Features. The HP MPower multimedia composer presents a flat, scrollable view of a document. This allows the user to access any and all document parts rapidly with no enforced sequential ordering. The media objects are embedded in the document, that is, the actual media data is copied into the resulting file rather than having links maintained to the original data. This ensures that the recipient of a message not only has immediate access to the data, but also results in larger messages and freezes the data at the time of composition. The media objects can be embedded at any location within the document, providing significantly more flexibility than the “attachment” model used by many mailers. The attachment model maintains links to other parts of a document.

To incorporate a media object into a document being composed, the user can either use the Include dialog facility of `vuepad` or drag the object from a file manager view and drop it on the composer’s window. If the user wishes to create a new media object while composing a message, the appropriate editing tool (e.g., the audio editor or the image editor) must be used to create the object within that editor. After the media object is created and saved in the file system, `vuepad`’s Include dialog or drag and drop facilities can be used to include the new object in the message being composed.

As shown in Fig. 1, the composer displays icons in place of all nontextual portions of the document. This results in a reasonably attractive appearance and good user recognition because of the similarity with the HP VUE file manager

† The HP VUE file typing mechanism provides the capability to define classes of files. For example, it can define all files ending with `.tif` to be of class TIFF.

iconic view. The user can either double-click with the mouse, or press the **Return** key to activate the currently selected icon. This provides a simple and quick means of viewing or playing the media item and allows easy access from the keyboard for those users who prefer not to move their hands to the mouse.

The composer also provides an **Actions** menu item which is sensitive whenever a single media item is selected. This menu contains two items. The first item is **Open**, which duplicates the functionality of double-clicking on the selected icon. The second item is **Save As**, which allows the user to save the selected media object to another file separately, with no MIME structure. The **Save As** action is also the default "open" action of the unknown data type. The unknown data type allows users to pass arbitrary binary data through the mail system undisturbed.

The current implementation of the composer does have some limitations, but a foundation has been built upon which to improve and add features to the current version.

Multimedia Data Extensions

As mentioned previously, HP MPower supports the interchange and storage of several different types of multimedia data contained in a single message or a file via a format conforming to an extension to internet mail known as MIME (Multipurpose Internet Mail Extensions). This multipart, multimedia support is implemented in HP MPower by two utilities: *metamail* and *vuemime*. *Metamail* is a public-domain, sample implementation of a full-featured MIME agent. *Vuemime* is a MIME filter that was created specifically for HP MPower and HP VUE to simplify generation and manipulation of multipart, multimedia data from the various HP MPower components.

We selected MIME for its strong support within the internet mail community, and we found it to be also useful outside the mail domain. The HP MPower environment recognizes files structured in compliance with the MIME specifications and will deal with them transparently for editing and printing. The composer allows the user to insert nontextual objects into any document being edited, and the resulting file will be saved in MIME format. The MIME message format is also used for cut and paste operations between editing windows. This allows the user to treat editing of mixed-media data in the same manner as plain text. Unfortunately, the mail composer is currently the only application that understands MIME data in cut and paste operations.

This section first provides some background on MIME, describes *metamail* in the context of HP MPower, and then discusses *vuemime* in some detail.

MIME Background. Since 1982 the standards that form the basis for internet mail have been defined by RFC 822 and the SMTP (Simple Mail Transport Protocol) defined by RFC 821.† RFC 822 was intended to specify a format for text messages and, as such, did not explicitly allow for inclusion of multimedia messages such as audio or image data. In particular, RFC 822 defines a message as consisting of two parts: a header and a body. The header consists of a series of specific field names and field values followed by a blank line that

marks the end of the header and the beginning of the body. The body is restricted to relatively short lines (1000 characters) of seven-bit ASCII characters which cannot exceed a certain length. Users who wish to include nontextual data have to convert the data to seven-bit ASCII before submitting the data to their mail user agent or mail program.

RFC 1049 attempted to rectify some of the deficiencies of RFC 822 by defining a header field and a content type that marks the entire message body as being a certain type of data (e.g., text, audio, video, etc.). In the absence of a content-type field, the body was assumed to be U.S. ASCII text, as before. Although RFC 1049 has been used by several implementations, it is not without problems. The most severe problem is its total lack of support for multipart mail. RFC 1049 allows a message body to be specified as containing something other than text, but only one such thing.

RFC 1341, or MIME, generalizes and extends RFC 1049 in several ways. Most important, it defines a new content type called *multipart*, which can be used to encapsulate several body parts within a single RFC 822 message body. It also goes far beyond RFC 1049 in explicitly describing the set of allowable content types by defining a subtype mechanism for content types that includes provisions for addressing standardized encoding of non-ASCII character sets. It should be noted that RFC 1341 is an extension to, rather than a revision of RFC 822 in that it defines these new features (including text of unlimited line and overall length, characters sets other than ASCII, and multifont messages) within the confines of RFC 822.

The new header fields and content types defined in the MIME extension are described on the next page.

Metamail. *Metamail* is a public-domain, sample implementation of a MIME agent that was designed to function as a back end for an existing mail user agent. It can be incorporated into virtually any mail reading (or bulletin board) program (e.g., *xmail*, *xmh*, *elm*, etc.), enabling it to become a multimedia reading interface. *Metamail* knows how to parse a structured MIME message, flatten any hierarchy of nested messages, decode the various parts, and optionally, dispatch the appropriate handlers or viewers for the different parts. The commands used to dispatch the handler or viewer for each content type are specified in one or more *mailcap* (configuration) files, which allow a great deal of flexibility in adding and configuring handlers. As a viewer, *metamail* can be thought of as a multimedia counterpart to the ASCII paging tool, *more*, except that it enlists the aid of additional handlers and viewers when paging through a message in a linear fashion.

In HP MPower we needed more flexibility when composing, viewing, printing, and sending mail so we did not incorporate *metamail* directly in our mail user agent (*elm*). Instead, we use the enhanced version of *vuepad* described above as our mail viewer and composer and we delegate *metamail* to the role of a MIME filter or preprocessor and postprocessor. In this role, *metamail* serves primarily to simplify the digestion and generation of MIME compliant messages by other HP MPower components. Specifically, *metamail* is used to flatten potential message hierarchies and to encode and decode each message part according to its encoding scheme. On the input side (mail receiving), for example, if we receive a MIME message from a non-HP MPower sender, we pass it to

† Each internet standard is defined by one or more standards each known as a Request for Comment, or RFC.

MIME Header Fields

The MIME extension (RF 1341) to the basic internet mail standard RFC 822 created the following new header fields:

- **MIME Version.** This field is used to specify a version number that declares a message conformant with the MIME standard
- **Content Type.** This field is used to specify the type and subtype of data in the body of a message and to specify the complete encoding of such data
- **Content Transfer Encoding.** This field is used to specify auxiliary encoding applied to data to allow it to pass through mail transports having data or character set limitations
- **Content Identifier and Content Description.** These fields are used to further describe data in the body of the message.

Content Types

MIME enumerates precisely seven valid content types and requires that any additions to this set be specified in a new, similarly formal document. This restriction is a major change from RFC 1049, which allowed for much freer definition of new content types. Instead, the new mechanism for extensions is to define new subtypes of established content types. In general, implementors are required to register new subtypes with the Internet Assigned Numbers Authority (IANA) to avoid name conflicts. (An exception is private subtypes beginning with the letter X, which can be used freely and without registration.)

The seven defined content type values are:

- **Text.** This is the default content type. The default subtype is plain text. This content type has a `charset` attribute that has the default value `us-ascii`.
- **Image.** This content type is for still images. Subtypes are image format names (e.g., `image/gif` and `image/jpeg`).
- **Audio.** This content type is for audio information. Subtypes are audio format names. For example, `audio/basic` denotes single channel 8000-Hz μ -law audio data.
- **Video.** This content type is for video frames. Subtypes correspond to video format names such as `video/mpeg`.
- **Message.** This content type is used to encapsulate an entire RFC 822 format message. For example, it can be used in forwarding or rejecting mail. The standard defines two subtypes of message: `message/partial`, which can be used to break a large message into several pieces for transport so that they can be put back together automatically on the other end, and `message/external-body`, which can be used to pass a very large message body by reference, rather than including its entire contents.

It should be noted that a message with a message content type can contain a message that has its own, different content-type field, meaning that the message structure can be recursive.

- **Multipart.** This content type is used to pack several parts, of possibly differing types and subtypes, into a single RFC 822 message body. The content-type field

specifying a multipart type also includes a delimiter, which is used to separate each consecutive body part. Each body part is itself structured more or less as an RFC 822 message in miniature, possibly containing its own content-type field to describe its type. Subtypes of multipart types are specifically required to have the same syntax as the basic multipart type, guaranteeing that all implementations can successfully break a multipart message into its component parts. An expected use of multipart subtypes is to add further structure to the parts and to permit a more integrated structure of multipart messages among cooperating user agents.

- **Application.** This content type is for most other kinds of data that do not fit into any of the above categories, such as list servers, mail-based information servers, and PostScript.TM

A separate part of the content-type header field can be used to convey supplemental information that may be either optional or required, depending on the content type. Such parameters are given in `keyword = value` format, and are used, for example, to convey information about character sets for text objects. Thus, the default message type for internet mail can be given a MIME content type of:

```
Content-type: text/plain; charset=us-ascii
```

Content Transfer Encoding

If internet mail transport (SMTP, as described by RFC 821) is ever upgraded to permit arbitrary binary data of unlimited line length in message bodies, the issue of encoding a message for transport will go away. However, even those who advocate such changes to SMTP generally recognize that they will be slow in coming. In the interim, there is wide perception that a standard mechanism for encoding arbitrary binary data for mail transport is needed.

The content transfer encoding header field can be used to specify the encoding technique used to render binary data in short lines of seven-bit data. After much debate, the working group settled on two encodings, which may be used interchangeably. One of them, the base-64 encoding, encodes each three bytes of binary data as four bytes of 7-bit data, using a base-64 alphabet selected for maximum portability across SMTP implementations, including ASCII to EBCDIC gateways. The other encoding scheme, `quoted-printable`, is a less efficient representation that preserves nearly all 7-bit ASCII characters as themselves. It is expected that base-64 will be preferred for genuine binary data, while `quoted-printable` will be preferred for data that is largely U.S. ASCII, but has scattered non-ASCII characters within it. In particular, this may be the preferred encoding for textual email in the national-use variants of ASCII, ISO 8859-X.

If the content transfer encoding field appears in the RFC 822 message header, it refers to the body of the message. If it appears in the header area of one part of a multipart message, it refers to the body area of that part only. The content transfer encoding field is prohibited when the content-type field has a value of `multipart` or `message`. This is necessary to prevent nested encodings.

`metamail`, which filters it by flattening any nested message hierarchies and decoding each message part. The result is a simplified MIME template that can be easily dealt with by other HP MPower components. On the output side (mail sending) `metamail` is used mainly to encode message parts for handling by the SMTP transport used by our mail transport agent `sendmail`. While we can handle nested messages on the input side, we only generate flat, single-level messages on the output side.

The official `metamail` documentation and software, including a draft of RFC 1341, can be found in the `pub/nsb` directory on `thumper.bellcore.com`.

Vuemime. In HP MPower, interaction with MIME messages is not the sole domain of the mail user agent (`elm`) and its viewer and composer (`vuepad`). Since MIME is used as the storage format for all multipart, multimedia data, printing

and sending MIME data as well as composition and viewing have to be performed independently of `elm`. Printing and sending, for example, can be done by simply dropping a MIME file on the HP VUE front-panel printer or mail icons without invoking `elm` or `vuepad`.

`Vuemime` was created specifically for HP MPower components to provide a single interface to MIME data and, as such, `vuemime` is the only HP MPower component that directly enlists the services of `metamail`. Essentially `vuemime` can be viewed as a higher-level MIME filter which, in addition to providing the type of filtering performed by `metamail`, provides intermediate formatting that is easily digested by various HP MPower components. What has resulted, after analysis of the needs of various HP MPower components, is essentially five levels of formatted data which can be viewed as stages in a file's morphosis from raw data (level 0) into a fully formatted



Fig. 4. Processing an incoming message. (a) Level-3 file format and the actions of vuemime to move from a MIME format to a level-2 file format. (b) Level-2 file format and the actions performed by vuemime to transform the data to a level-1 representation. (c) Level-1 file format representation and the actions taken by vuepad to display textual and nontextual parts of the message.

MIME message ready to be handed off to the SMTP mail transport.

Fig. 4 shows the steps involved in transforming an incoming message in MIME format (level 3) through the various levels of formatted data to a tagged (level 1) format. For an outgoing message, vuemime will take a tagged message and generate a self-contained MIME message that can be passed to the mail transfer agent.

The five levels of formatted data shown in Fig. 4 are defined as follows:

- Level 0. This is raw data consisting of ASCII text and nontextual information such as image, audio, and video data.
- Level 1. Data at this level, which is used by vuepad and the mmprint action, is in a special tagged format that contains only ASCII text, summarized mail heading information, and special tag lines representing multimedia objects. The tag lines indicate the multimedia type and a path to a file containing the raw data. See the level-1 file representation in Fig. 4c.
- Level 2. This is an external-body file format. The format at this level is very similar to level 1 except that MIME header and control lines have been added to make it fully MIME

compliant. All multimedia objects are still external (and raw) but are now identified by Content-type: <message>/<external-body> and Content-type: <type>/<subtype> lines. See the level-2 file representation in Fig. 4b.

HP MPower does not currently send messages in level-2 file format, but this level could be useful in at least two situations. First, level-2 format could greatly reduce the size of a message by not sending the actual contents of something like a video clip. In this case, the video information is not sent until and unless the receiver wishes to view the video. The other important use of level-2 file format is to provide a “hot link” to the current version of some data. For instance, a message might contain an external-body reference to some data that is updated hourly. The receiver of the message would see valid data at the time the message is read, instead of the data as of the time the message was sent.

- Level 3. This is a MIME file. All references to external multimedia data files are replaced by encoded data. See the level-3 file representation in Fig. 4a.
- Level 4. This is a MIME file with an outgoing mail address and header information (from the mail user agent) prepended to it. At this level the data file is ready to be transported to the network services.

Vuemime accepts command line options to transform data in either direction and between any levels from 0 to 3. Level 4 is generated solely by the mail user agent (mmsend) for outgoing mail. The HP MPower component (vuepad) simply indicates to vuemime on the command line the level of the file being transformed and the level to which it is to be formatted.

Vuemime generates only base-64 encoded data when going from a level-2 (or lower-level) file to a level-3 (MIME) file even though it can accept other types of encoded data when going from a level-3 file to a lower level. Also, vuemime only generates flat multipart and mixed level-2 and level-3 messages even though it can accept and digest (via metamail) nested level-3 messages generated by a non-HP MPower MIME system.

Mimetypes. Transformation to and from raw data by vuemime is controlled by a file called `mimetypes`. The `mimetypes` file is a cross between `metamail`'s `mailcap` files and HP VUE's file-type definition files. Table I shows the contents of the `mimetypes` file supplied with HP MPower 1.0.

Vuemime uses the data in the first column to map level-0 (raw data) files to the content type for files that are being transformed to a higher level. The third column is the content-type value inserted in files of level 1 and above. For level-1 files and above that are being transformed to level-0 files, the column 2 entry associated with a particular content type is used to determine the file name extension to be applied to the new raw data file.

Conclusion

Our multimedia email project is an example of a highly leveraged effort that combined existing blocks of functionality, some modifications, and new “glue” to meet a market need quickly.

Table I
Contents of the HP MPower Mimetypes File

Search Pattern for Raw Data Files	File Extension for New Raw Data Files	Content-Type Value
.\.xwd	%s.xwd	image/x-xwd
.\.xd	%s.xd	image/x-xwd
.\.tiff	%s.tiff	image/x-tiff
.\.tiff	%s.tiff	image/x-tiff
.\.eps	%s.eps	image/x-eps
.\.pcl	%s.pcl	image/x-pcl
.\.gif	%s.gif	image/gif
.\.jpg	%s.jpg	image/jpeg
.\.jpeg	%s.jpeg	image/jpeg
.\.pm	%s.pm	image/x-xpm
.\.xpm	%s.xpm	image/x-xpm
.\.bm	%s.bm	image/x-bitmap
.\.xbm	%s.xbm	image/x-bitmap
.\.bmf	%s.bmf	image/x-bmf
.\.ps	%s.ps	application/PostScript
.\.txt	%s.txt	text/plain
.\.au	%s.au	audio/basic
.\.l16	%s.l16	audio/x-Linear16
.\.l8	%s.l8	audio/x-Linear8
.\.lo8	%s.lo8	audio/x-Linear-8Offset
.\.wav	%s.wav	audio/x-microsoft-RIFF
.\.snd	%s.snd	audio/x-NeXT
.\.u	%s.u	audio/x-MuLaw
.\.al	%s.al	audio/x-ALaw
.\.Z	%s.Z	application/x-compress
.\.tar	%s.tar	application/x-tar
.\.unk	%s.unk	application/octet-stream

Acknowledgments

The multimedia mail team offer a special thanks to Gabe Begeg-Dov, a former HP engineer, for discovering the leverage possibilities in the MIME internet standardization efforts.

HP-UX is based on and is compatible with UNIX System Laboratories' UNIX* operating system. It also complies with X/Open's* XPG3, POSIX 1003.1 and SVID2 interface specifications.

UNIX is a registered trademark of UNIX System Laboratories Inc. in the U.S.A. and other countries.

X/Open is a trademark of X/Open Company Limited in the UK and other countries.

PostScript is a trademark of Adobe Systems Incorporated which may be registered in certain jurisdictions.