Recent multimedia research efforts have resulted in the slow emergence of multimedia mail systems as a viable enhancement and replacement of the traditional ASCII-text only mail systems. Existing multimedia systems, which are mainly homogeneous systems, are generally more concerned with providing the necessary tools to allow multimedia mail to be composed and electronic communication to take place among users. Provision of basic functionality alone is insufficient to assure long term success and large scale users' acceptance. There is a distinctive need to address both aspects of efficiency and effectiveness in multimedia electronic mail systems so that it can be a viable alternative to traditional text mail.

This paper proposes a number of key success factors pertaining to the system and user level in the delivery of the next generation of multimedia electronic mail systems. It demonstrates how these factors have been considered in the design and implementation of a prototype multimedia system, MEmail, to function over the heterogeneous academic environment of Nanyang Technological University (NTU), Singapore.

**keywords:** multimedia information system, electronic mail system, heterogeneity, interconnectivity, filtering, query, information retrieval, object oriented database

1. Introduction

Text-only electronic mail (email) is a widely accepted medium of interpersonal communication for many years. However, with the rapid advances in information processing capabilities and the availability of larger storage devices, text-only messages are likely to give way to multimedia messages that will allow richer and more varied forms of message representations (such as graphics, images, speech, animation, text and video). This changing scenario is evidenced by the amount of ongoing research in multimedia and the small but growing number of commercial multimedia email systems that are available today. However, these systems have not taken off in a big way as its predecessor due to their lack of ubiquity. Reasons for this failure can be attributed to the
lack of data interchange data format, the huge storage requirements of video and other media, and the lack of standard facilities for viewing and composing multimedia documents. Along with the advances of technology, email users of today have higher expectations. It is insufficient just to provide the basic functionality of mailing systems alone. There is a need to address the efficiency and effectiveness of email systems as a means to provide value added service in electronic communications.

This research is initiated to examine existing multimedia systems, to derive the requirements for the next generation multimedia email systems and to implement a prototype for further study and evaluation. This paper identifies and proposes a number of key success factors and demonstrates how they are factored in the design and implementation of a multimedia email system, MEmail, to function over the academic environment of Nanyang Technological University (NTU). MEmail provides a local transparent heterogeneous multimedia email facility across different computer platforms that are linked through different local area networks within NTU.

The paper is divided in six sections. In the following section, a survey of existing multimedia electronic mailing systems is carried out. Based on the results of the systems survey and a users' survey, key success factors from the system and user viewpoint are identified and proposed in Section 3. These key success factors form the basis for the design and implementation of MEmail. Section 4 examines each success factor in turn and illustrates how each factor is being treated and implemented in MEmail. An overview of the system and the status of the work, together with some preliminary findings are presented in Section 5. Finally, the paper concludes with a summary of the accomplishments achieved through the design and implementation of MEmail.

2. Survey of Multimedia Electronic Mailing Systems

A number of existing multimedia electronic mailing systems was examined as a means to aid the identification of key success factors in the delivery of multimedia email systems. Particular attention was focused on research prototypes to identify the main features, objectives and design considerations of these systems. This survey was not intended to be a comprehensive product survey as those reported in computing magazines where each product is gauged individually along a different number of categories. The multimedia mail systems surveyed included the Andrew Message system[1], Diamond system[2], Montage System[3], and commercial systems such as Microsoft/MSMail[4] and NeXT/MediaView[5]. However, only the first three systems are reported in this section as these are most significant in aiding our derivation of key success factors. In presenting the survey, the design objectives and main features of each system are first examined. Strengths and weaknesses of each system are then summarised and tabulated at the end of each system surveyed.

Each email message being transported from the sender to the receiver comprises three basic elements:
(1) Transport medium (TM) which involves the underlying layer protocols to provide the data communications between computers or systems (e.g. TCP/IP, uucp, X.25/X.29).

(2) Mail transport agent (MTA) which is responsible for transporting mail from source to destination. It is often considered as the back-end of an email system which contains several components which include the routing mechanisms, local delivery agent (LDA) and remote delivery agent (RDA). (e.g. MHS (Novell's Message Handling System), SNADS (IBM's System Network Architecture Distribution System), X.400 (CCITT's standard) and SMTP (Internet's Simple Mail Transfer Protocol)).

(3) User agent (UA) which is the interactive portion of an email system to provide the functionality of mail systems, such as message authoring, reading, distributing, storage, query and retrieval, and so on. Advanced UAs may also exhibit facilities to include complete automation of the message reception/response/storage process as well as message encryption/decryption. UAs are generally developed for individual mailing systems depending on the level of support and services offered by the system.

**Andrew Message System** (AMS) [1] was developed as part of the Andrew project of the Carnegie Mellon University to build a large scale multimedia mail and bulletin board system. AMS’s objectives and design considerations include the provision for the reliability of message delivery, support for enormous message databases, wide range of machine types, media objects, interface styles and active messages.

AMS is built using the two major components of the Andrew project, namely, the Andrew File System (AFS) and the Andrew Toolkit (ATK). The AFS is a distributed network file system designed to provide the illusion of a uniform central UNIX file system to a large number of users. The ATK is a window-system-independent programming library that supports the development of user interface (UI) software.
The AMS architecture, as shown in Fig. 1, contains a number of components. The client programs refer to the UA of the AMS. It is supported by the Common User Interface Library that encapsulates and hides the server calls of the client programs, simplifies the task of constructing new UIs and supports the implementation of active messages.

The remaining components form the MTA of the AMS. The message database, which resides in the central file server, is only accessible by the message server program. Client programs access the message database through the message server through RPCs (remote procedure calls). The delivery system is responsible to deliver the message from the sender machine to the recipients. This mail transport mechanism is entirely separated from the UA (client program) and database agent (message server process). Hence, the AMS is made delivery-system-independent. **White pages** is a database package for mapping names to mail addresses, to find and report addressee ambiguities and to suggest corrections for misspellings based on phonetic matching and heuristic knowledge. Translators for the ATK's data stream to the "lowest common denominator" format are written to cater for client machines that are unable to understand the multimedia message. For example, the translator will replace an animation by a sentence like "An animation appeared here, but could not be displayed". Such a facility is included as an attempt to solve the heterogeneity problem. However, this translator is not made transparent to the client as it is done only upon the request of the client. The strengths and weaknesses of the Andrew system are summarised in Table 1.
Table 1: Andrew Message System

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Architectural design results in clean split of the</td>
<td>• Dependence on the propriety ATK datastream hinders interconnectivity and</td>
</tr>
<tr>
<td>components of the system, thereby allowing ease of</td>
<td>popularity.</td>
</tr>
<tr>
<td>future improvements, extendibility without adverse</td>
<td>• Limited support for a heterogeneous environment.</td>
</tr>
<tr>
<td>effect on overall architecture.</td>
<td>• Document translations not made transparent to user.</td>
</tr>
<tr>
<td>• Supports different media types, including active</td>
<td>• Message delivery system uses whole file transfers which can easily</td>
</tr>
<tr>
<td>messaging and hierarchical objects.</td>
<td>overload the network.</td>
</tr>
</tbody>
</table>

The **Diamond System**[2], developed under DARPA, is a system for creating, editing, transmitting and managing multimedia documents. As in the Andrew Toolkit, the Diamond system runs on high-function workstations and provides the capability for dealing with a number of multimedia objects that include multi-font text, raster images, hierarchical line drawings, spreadsheets, equations and voice. In designing the system, the emphasis was placed on the UI design with support for different UIs, implementation using a distributed architecture, protection from unauthorized disclosure and interoperability support.

![Diamond System Architecture](image)

**Fig. 2: Diamond System Architecture**

The Diamond architecture, as shown in Fig. 2, contains a number of components. Each access point which is a single user workstation computer supports user access to the
system. When the access point needs services from other Diamond components, it interacts with them by means of interprocess communication. The document store is a citation-based hierarchical storage system for storing multimedia documents and folders. Citations for the same document can appear in more than one folder. Such a design supports document sharing and reduces storage requirements. The document store is managed by a collection of document manager processes. The document manager runs on each server host that stores part of the document store.

The registry database together with the authentication manager supports message addressing and delivery as well as user login and access control. As such, each Diamond user must have a name and password for system access. The import/export manager is responsible for exchanging messages with other message systems. It interacts with the document store and an Internet message processing module that takes responsibility for routing messages to their destinations. It also performs protocol conversions between Diamond's internal format and the interchange format used for Internet message support. Finally, the scanner and printer manager provide services of image scanning and printing to Diamond users. The strengths and weaknesses of the Diamond system are summarised in Table 2.

Table 2: Diamond Message System

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Minimum storage requirements via the design of the citation-based document store that enables document sharing.</td>
<td>• Access points are single user homogeneous systems that do not provide solutions to a heterogeneous environment.</td>
</tr>
<tr>
<td>• Well structured document format that allows support of other application data types (such as spreadsheets).</td>
<td></td>
</tr>
<tr>
<td>• Distributed architecture system allows ease of future extendibility</td>
<td></td>
</tr>
<tr>
<td>• Translators designed to interconnect with the Internet environment</td>
<td></td>
</tr>
</tbody>
</table>

The Montage system[3], developed in the Georgia Institute of Technology, is designed to be built on top of an existing email transfer system. In designing the system, the developers wanted to create a mail system to implement the Montage model of multimedia messages which at the same time could be widely used. The system should also be capable of meeting future needs and be extendible by end users so that mechanisms should be present to dynamically support new media types without the need to alter or recompile Montage. To meet these objectives, the system has to be built on top of existing protocols and standards wherever possible. As such, the system was built on the UNIX OS with HP's X-Windows based widget set for the UI and SMTP as the mail transport software. Since Montage was built on top of an existing mail transfer system, the major implementation issues were on its multimedia message format and interchange format.
In Montage's multimedia message model, all media are one of two classes: static or dynamic. Static media are simple text, formatted text or still images. Dynamic media are video and audio clips. Each message has a primary media class that is either static or dynamic. This primary media class is the type of media presenting the "main thrust" of the communication from sender to recipients. While Montage mail headers conform to the SMTP standard, Montage makes use of a custom format for the message body. The use of existing MTAs limits Montage messages to consist entirely of printable ASCII characters. Message bodies are encoded in ASCII using the standard UNIX uuencode program. As this process will increase the encoded file size by 35% after encoding, the message body is first compressed before it is mapped into ASCII. The strengths and weaknesses of the Montage system are summarised in Table 3.

### Table 3: Montage Message System

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
</table>
| • Portability and flexibility of system by making use of existing and popular software for mail transport (SMTP mail system) and user interface (X Windows system).  
• System extensible by end users by built-in mechanisms to dynamically support new media types without need of recompilation. | • Homogeneous system not designed to cater for heterogeneity environment.  
• Translation of interchange format requires too many operations including encoding and decoding operations.  
• Proprietary standard for interchange format hinders wide acceptance since standards already exists.  
• Introduction of the primary media class inhibits the authoring and communication process. |

In a more recent development, an evolution approach based on adapters [6] was used to build a multimedia email system on top of an existing ASCII text-only mail system. The four layer architectural model distinguishes multimedia mail and provides facilities for media insertion, data encoding/decoding and mail browsing. The drawback of this system is in its proprietary approach that does not support any standard. Additionally, users have no control on the authoring aspect with positioning since the system operates on a sequential insertion mode during mail composition.

From these surveys, it became apparent that in the development of these systems, more attention was focused on the system level where emphasis was put on delivering systems that address the problems of data organization, data storage, media support, and interconnectivity issues. Very little emphasis was directed at the end user requirements of these systems. Nonetheless, the review of these multimedia email systems proved useful in allowing the authors to derive a number of key success factors pertaining to the system level.
3. **Key Success Factors For The Delivery of Next-Generation Multimedia Email Systems**

The key success factors for a new generation multimedia email system can be divided broadly into two categories pertaining to the system level and the user level. To derive the key success factors at the user level, a simple user survey was conducted for a subset of email users in NTU that comprised 35 honours year students from the computer engineering discipline. In the survey, users were asked to respond to the type of characteristics they would like their "ideal" email system to exhibit and the type of functions it should support.

Based on the information obtained from the systems review, users' feedback and the authors' perception, a list of key success factors was derived. These factors are by no means exhaustive, but are considered as important elements in multimedia email design. Factors associated with the basic functionality of mail systems such as the provisions for composing, sending, receiving mail, creating and managing folders, filing, printing, deleting mail, and so on must obviously be present and therefore are not included in the category of success factors.

Not in any order of priority, the key success factors from a system viewpoint include:

- **Ability of the mailing system to function over a heterogeneous environment.** Organisations using a homogeneous set of computers throughout its entire organisation is a rarity. This is especially so for large organisations and multi-national corporations. Existing multimedia email systems have been successful in small local user communities but have not perhaps achieved the critical mass of general users due to the very nature of their homogenous design that requires the same make of computers to be used in order to exploit the multimedia capabilities.

  In a typical heterogeneous environment (such as NTU), personal computers, workstations and even mainframe computers of different makes can co-exist within the network. The exchange of multimedia documents becomes complex in such an environment due to hardware and data format incompatibilities among them. In addition, one should not expect all computers used within the email environment to be equipped with identical media handling facilities. Thus, some computers may support both audio and video media types while others may only support the audio media. In this instance, the mailing system should still be able to detect the deficiency in the latter and report it to the user, but at the same time still being able to play the audio part of a video clip, if necessary.

- **Ability of the system to communicate and exchange messages with other multimedia mailing systems** (Interconnectivity). Mailing systems should not be designed as closed-door systems if it wishes to gain long term support and usage from the masses. The main concern of this interconnectivity problem lies on the lack of a standard method for transporting multimedia electronic messages as well as lack of agreement on the data formats among the different mail systems. For example, users of Andrew, Diamond, NeXT and MS-Mail systems are currently incapable of
exchanging messages among each other. Vendors have vested commercial interest in particular representative formats so that the prospect of an early agreement is not conceivable.

This problem can be tackled using two approaches. First, is the use of the "top-down" approach where vendors produce their own standard format and enforce all users to switch to this standard. This approach as verified on Andrew, Diamond and other vendor specific systems is unlikely to succeed because users are reluctant to change their working environments and send messages if readability is uncertain. Second, is the use of the "bottom-up" approach where users will send a multimedia message in whatever format is used in their system. At the recipient end, the mail reading program will make provisions to recognise this mail as a foreign mail and will initiate the corresponding program (supplied by the originator) to handle and display the message. In order for this approach to work, these display and translation software must first be distributed by the sender to the receiver's mail system, installed and integrated with other mail systems' software into mail reading program before it can be used.

- **Provision of a good message handling and storage mechanism for multimedia mail.** In cases where a piece of mail is to be sent to more than one user, using a distribution list or otherwise, information sharing and handling becomes crucial due to the size of these multimedia messages. At such, the practice of physically sending a copy of the same message content to each recipient must be avoided. Network traffic should be minimised as much as possible. In addition, the system must be able to satisfy the potentially enormous storage requirements for multimedia messages. Therefore, the mailing system should be made accountable for the overall handling and storing of messages for better repository utilization. Data compression is also desirable to reduce the amount of storage requirements of multimedia mail.

  Only one single copy of the same message should be kept by the system to be shared by all users. To minimise network traffic even further, the message header can be the only piece of information sent to each recipient initially. The message body should be sent across the network only when it is specifically requested by the user. Along with this, the system should send only those media data contained in the message body that are supported at the recipient's machine. For instance, if the recipient's machine do not have audio handling facilities, the audio media information contained in the message body should not be sent unless it is been specifically requested by the user.

- **Extendibility to handle and support new media types.** The term "media type" although traditionally meant to denote the familiar text, graphics, image, audio and video, can also be extended to include other forms of information such as tables, formulae, spreadsheets, word-processed documents, CAD/CAM drawings, etc.. Such information is generally not properly supported in mail systems, instead, they are often treated as attachments to the mail. Therefore, it becomes desirable for the system to be extendible to accommodate and manipulate these new media types.
Such a facility is possible if a mechanism to incorporate and manipulate these new media types is facilitated in the system architecture design.

The key success factors from a user viewpoint include:

- **Presenting only relevant and important mail to the user and discarding all other forms of junk mail.** Users of mailing systems are too well aware of the use (or abuse!) of distribution lists and bulletin boards that have resulted in them being inundated with large amounts of junk mail that are intermixed with important ones. This resulting information overload results in two implications. First, it takes considerable time for the user to read and distinguish among the junk and important messages. Second, the large volume of messages also presents difficulty for the user to browse and find the desired previously read message.

  The presentation of only relevant mail is possible if some form of filtering can be introduced to intercept and verify all incoming mail. Junk messages are discarded while some form of appropriate action on useful messages, such as automatically routing to folders, auto-replying, printing it, and so on can be taken.

- **Availability of a set of resource management tools to aid management, browsing and retrieval of messages.** Existing email systems do not generally provide sufficient tools for browsing or retrieving messages that were read in the past. The use of folders to denote different categories is sufficient for message organisation but becomes increasingly difficult for message browsing and retrieval, especially when the number of kept messages grow over time.

  It is thus desirable for the system to aid the user locate and retrieve any message within the mailing system effortlessly. Some form of query and retrieval process can be employed to achieve this goal. This is a classic information retrieval (IR) problem for which a number of established solutions exist. For instance, the user's information needs is expressed through query formulation using keywords that are then matched against a set of index terms extracted from the message header or body. The number of keywords that matches with the index terms will determine the extent of match or relevance between the query and the message. The system, having found the relevant messages that comply with the user's query, will return this set of output together with other possible messages of potential interest to the user.

- **Provision of a "good" user interface to allow access and use of the system's facilities, while at the same time tailored to strike a balance for use by both groups of novice and expert users.** The user interface determines the quality of interaction between the end user and the mailing system. This can ultimately result in either the user enjoying and accepting the system or despising and rejecting the system. Good interface design requires attention to be focused on aspects at task level and dialogue level so that the correct devices and dialogue styles are employed. Additionally, the established good human factors in interface design, such as being
consistent, provide feedback, minimise error possibilities, provide error recovery, provide adequate on-line help, minimise memorization and accommodate multiple skills' level should all be incorporated in the design of the UI.

4. ME-mail - A Heterogeneous OODB-Based Multimedia E-Mail System

ME-mail [7]-[10] is a research prototype built to provide a transparent multimedia email facility across the heterogeneous computer platforms within the academic network of NTU. ME-mail attempts to incorporate the key success factors identified in the previous section in its design and implementation.

![Network Architecture of ME-mail in NTU](image)
MEmail, developed using an object-oriented approach, is a client-server based system that allows email users to store, manage, retrieve and exchange multimedia messages (which can comprise text, graphics, images, video and audio). MEmail's system architecture is shown in Figs. 3 and 4. Clients can either be Unix workstations or PCs. MEmail makes use of mail and media servers and object-oriented databases to process, store and manage the different forms of media data. A production-rule based system is employed to filter incoming messages so that the system will only deliver what is deemed "relevant" to the user, while discarding all junk messages.

MEmail is integrated with a typical query and retrieval facility found in traditional IR systems. Users are allowed to query on the known fields of the mail (such as From, Subject, Folder, Date and so on) as well as the contents and description of each media object in the message. The provision of this facility is made easy since MEmail is built using databases which in itself encompasses powerful query and retrieval capabilities. MEmail supports relevance feedback, so that if users can identify a list of relevant messages, the system will make use of information contained in these messages to identify new related messages and present a re-ranked message list to the user.

4.1. MEmail design issues

The key success factors identified in Section 3 were considered and factored in the design and implementation of MEmail. These are elaborated below:

Heterogeneity Support

MEmail is designed specifically to handle a heterogeneous network environment where different makes of computers can co-exist together. The exchange of multimedia messages is complex in such an environment because of the differences in hardware and data
formats used in these systems. In using the client-server architecture for the design, there are three possible approaches for clients to access the mail server: off-line method, on-line method and disconnected method.

In the off-line method (e.g. uucp, POP-3), client applications periodically connects to the server, downloads all messages to client machine and deletes these messages from the servers. In the on-line method (e.g. NFS, IMAP4), the client manipulates data directly on the server machine. The client must remain connected to the server throughout the whole login session. No mail data is kept on the client site and the client retrieves all the data from the server as it is needed. The disconnected method (e.g. PCMAIL(RFC1056)) is a combination of both earlier approaches. In this model, the client downloads some of the messages from the server, manipulates them on the client machine and then at some later time updates the changes. The server does all the synchronization and repository functions of the whole system. However, none of these three approaches (without any modifications) are suitable for multimedia messages because having one server to act as both message storage and multimedia storage can cause severe overloading at the server and hence cause system degradation.

MEmail adopts the disconnected method in the design but instead of using a single server, uses more than one server: one for message handling and storage and different servers for media storage and processing (Fig. 4). If necessary, these servers can reside on different sites to reduce the load at a single site. Such a design also facilitates nicely in handling the heterogeneity problem since the media servers in addition to their storage function, can be used for handling translations and transformations to resolve incompatibilities of hardware and data formats between different client machines.

In MEmail, different media servers are used to handle different media types. An email message created in the sender's machine will have its constituents of various media stored in the respectively media servers in its original form. Before sending the information to the recipient, the media servers will first check for the hardware support in the recipient's machine. If this is present, a subsequent check on hardware and data format incompatibility is carried out. Configuration files maintained by the user at the recipients' machines provide the necessary information of hardware support and data format for the media servers. Appropriate translations and transformation are then applied as necessary to the recipient's machine requirements at the media servers before they are sent to the recipient. Although, this approach of handling heterogeneity requires more processing, it nevertheless, preserves the original format of the various media types in the originator's message. The other alternative approach of converting all media data into some internal standard format representation was considered but not implemented in MEmail, as the preservation of the original form of message outweighs the benefits of reduced processing requirements of the latter approach.

**Interconnectivity**

The interconnectivity issue is unfortunately a problem without a prospective short-term solution. Obviously the optimal solution is to have all the email systems' community to
agree, identify and use a de-facto standard for data representation and method for transporting multimedia electronic messages.

In considering the interconnectivity issue in MEmail, the assumption is made that a such form of consent and co-operation is unlikely to happen so quickly, so that messages will still have to be translated between mailing systems. Thus, the "bottoms-up" approach has been adopted in MEmail with the aim to use a minimum number of translators by promoting the usage of a de-facto standard data format representation. At such, translators will still be needed for interchanging the data format from the proposed standard to each other until such time a de facto standard emerges.

The proposed standard mail representation format of MEmail is MIME [11], an extension developed for SMTP[12] mail systems to include multi-part messages. The choice for this combination is twofold. First and more importantly, is the flexibility and extendibility to enable multiple (and new types of media) objects to be stored in the same single message. Second, the Internet (which uses SMTP as its MTA and TCP/IP as its TM) community, with an ever-increasing number of new users each day, is far too large an email community to ignore. For every different mailing system which MEmail wishes to communicate, there will be a corresponding pair of translators needed - to convert to and from each other's format.

*Extendibility to support new media types*

The issue of extendibility is best approached by recognising that different media types are essentially different objects with different attributes. It becomes therefore, a natural choice to use an object-oriented approach in designing MEmail since these objects together with its operations can be associated with different classes. These classes can then be organised into class hierarchies, thereby allowing generalisation and aggregation to occur.

MEmail is able to support the integration of a new media object to the system by defining a new metaclass object (instance of class) with its attributes, operations and functions to perform these operations. MIME fits into this design by allowing these objects to be combined into one same file, so that the constituents of each multimedia message can be compacted together prior being sent to the recipient. It also permits a metamail program to be used to determine how to display the various multimedia objects.

*Media Storage and Support*

The media servers, in addition to being used to solve the heterogeneity problem, also serve another purpose as being responsible for media storage and support. The accountability for the overall storage of messages is achieved in the system by the architectural design where each media server is responsible for managing its own media type. The main advantage of such an approach is to ensure that only one copy of each unique media object is kept and that no duplication exists. This makes it not only possible for the system to store only a single copy of a multimedia message that can be shared by...
many users, but also that these objects can be shared among different messages that contain them. Such a scheme together with data compression techniques such as MPEG and JPEG will reduce the data storage requirements substantially.

**Message Filtering**

In order to minimise junk and maximise relevant mail reaching the recipient, some form of an automatic filtering mechanism is desirable. There are essentially two main approaches used in filtering: *keyword filtering technique* [13] and *production rule based system* [14].

In keyword filtering, each message is to be sent is represented by a series of 'keywords'. These keywords are then matched against an index in the recipient's filtering mechanism that will either accept or reject the document or message (for examples: see [14] -[17]). Two problems are inherent in this technique. The first is the difficulty in selecting keywords to represent the subject content of the messages. The second is that the use of only pre-defined descriptors to represent the message content is a severe limitation to the filtering system.

Production rule based filtering makes use of the fact that email messages are semi-structured messages so that rules can be formulated to process and filter these messages (for examples: see [14],[18]). A semi-structured message is a message containing a known set of fields, but with some fields containing unstructured text or non-textual information, such as audio clips or images.

The structure of the rules is very similar to those used in logic study and rule-based expert systems. Rules are represented in the form of IF (condition is TRUE) THEN (prescribed action) syntax. Multiple conditions when specified are "ANDed" together to obtain the result. Boolean combinations are possible within fields so that compound rules can be formulated. Each user has his or her user profile to store all the pre-defined rules. The user can monitor and check to see if the messages received are those of interest or little relevance. New rules or changes to existing rules can be made periodically to ensure proper filtering. This technique exploits the structure of semi-structured messages by allowing the user to define superior filtering conditions compared to the keyword filtering technique. Although most rules can be intuitively defined, its main disadvantage lies in the difficulty for novice users to formulate the rules correctly and in the syntax required.

MEmail adopts the production rule based approach for message filtering. Due to the added constituents in the data model for multimedia mail, the user will have more filtering options. For instance, the user can delete all incoming mail that contains video clips if such a facility is not present at the client machine. Alternately, the filtering can pick up mail containing only audio clips. In this way, MEmail can be configured to act solely as a voice mail system.
Mail Query and Retrieval

Most existing multimedia mail systems are either folder or file-system based which limits its IR potential. The use of a database system to construct mail systems is only demonstrated by Kent et al [19]. The main advantages of using database systems for building email systems include those of data organisation, data integrity and security, and built-in powerful query and retrieval engines.

MEmail supports both filing systems and object-oriented databases leading to enhanced IR characteristics. Users have different information needs and interests that change over time such that it would not be possible to prescribe an "optimal" IR model to satisfy their changing demands.

MEmail uses a modified best match search based on the probabilistic model for information query and retrieval. The best match search will result in a ranked output so that documents that are most "similar" to the query are presented in order to the user. In order for a measure of this similarity to be computed, some form of weight assignment is made to the original message and query words.

In addition to normal query and retrieval capabilities, MEmail will allow, after a sorted list of output is obtained, the user to perform a relevance feedback search. In using relevance feedback, the user will first select the most relevant messages from the sorted list and invoke the system to look for similar messages to those selected. The system will then re-weight the search terms and re-order the messages. Unchecked messages that are similar to the relevant documents, but are earlier given lower weights, would now be brought up to the head of the output list. This approach to IR, adopted in MEmail, provides the flexibility to the users to control the search criteria and messages retrieved.

User Interface Design

MEmail's UI design is one based on considerations for both novices and experienced users of mailing systems. For example, users can choose between a text and graphical user interface to define the production rule filtering definition in their user profile. Users who prefer the "point and click" paradigm will be able to select the latter interface which provides more prompting and requires minimum typing. In addition, the desirable human factors in design identified in Section 2 were closely adhered in designing the layout and functionality of the user interface. Whenever possible, the whole interface screen would be used to present groups of related information to the users, with menus, commands and status areas of the interface remaining consistent between interfaces.
4.2. MEEmail functional components and implementation details

Fig. 5 shows the functional components of MEEmail. The Mail server consists of two specialised components with each performing certain message processing functions:

- **Mail manager** that mediates all the processing for message handling for each subnet system (such as notification of new mail, error reporting in sending out mail, etc.).
- **Filter manager** that intercepts incoming messages and performs predefined actions based on the rules defined in the user profile.

The Media server also contains two components:

- **Object manager** that provides the main storage, modelling, management and retrieval support of mail messages.
- **Operation manager** which intermediates the display and translation of different media objects across the heterogeneous environment.

Each client subsystem (the UA of MEEmail) contains five components:

- **Control panel** that is responsible for the management of various kinds of user interactions.
- **Authoring tool** for message browsing, composition and destination definition.
- **Query tool** that is used for query formulation and message retrieval.
- **Filter tool** that is used for user profile definition or modification.
- **Local object manager** that functions similarly with the server’s object manager but stores and processes messages that are currently edited or created by the client.

MEmail’s **control panel**, which is front end user-interface of MEmail, is shown in Figs. 6a and 6b (for Unix and PC clients respectively). It is divided into a number of areas: **Toolbox area** (for different message handling functions such as compose, send, reply, forward, extract, move, print and delete mail), **Menubar area** (which uses a click and drag paradigm to perform operations similar to Toolbox area plus other utilities such as folder management, address book management, and help information), **query and retrieval area**, and **editor area** (for message composition and editing).

The **authoring tool** employs the select/copy/cut/paste paradigm plus WYSIWIG concept to allow users to have an exact feel of what is been created or edited. In this manner, multimedia objects can be combined and integrated easily in the message. Users are allowed to attach other messages to the current one being composed. These attachments are represented as icons in the editing area. Message formatting is achieved with the grabbing, dragging and dropping paradigm.

![Fig. 6a. MEmail Control Panel for Unix Workstation Client](image-url)
The query tool is an integral part of the control panel to allow immediate user access to carry out message query and retrieval. Messages can be queried through a combination of the following fields: FOLDERS, FROM, DATE, SUBJECT, TEXT content and CONTENT-TYPE. The ORed operator is intrinsically implied within fields except for the SUBJECT and TEXT content where both AND and OR relations can be used within query words. The normal relational operators for DATE (=, <, >, <=, >=) are supported. The TEXT content allows the various media descriptors to be queried in addition to the SUBJECT query. In addition, a search on the CONTENT-TYPE can also be performed. Thus, a user can pose a query to search for all messages that contain images.

To allow the best match search to be performed, the SUBJECT and TEXT contents of each new message must first be indexed. Indexing starts with the elimination of all numbers, stopwords followed by stemming. Root words resulting from stemming are inserted into an inverted index for subsequent query operations. The inverse document frequency weight is computed for each stemmed word in the message and subsequently used for the computation of the message relevance.

The output from the query is a retrieved message list in ranked order. Alongside the left of each message listing is a "signature" area to allow multimedia information (such as an image of the sender, or an audio clip recorded by the sender) to be displayed and activated, if necessary. This is an example of another small but contributory effort of ME-mail to enhance the effectiveness of email systems. If a signature is not present, a default image is automatically displayed by the system in the signature area. Users can double click on the message listing to display the message in the authoring area for
browsing. Users can also select messages from the message list to carry out a re-ranking of messages based on relevance feedback.

The filter tool allows the user to define or modify the filtering rules in user profile by calling up either of the two available editors as shown in Figs. 7a and 7b. Users can apply filters to the header fields: FROM, SUBJECT, KEYWORD and DATE. Wildcard characters and pattern matching allow more general conditions to be defined. In addition, Boolean operators are also provided to allow more complex rules to be formulated by combining fields. The normal relational operators for DATE are supported. The different types of action supported by MEmail including PRINTing, filing messages to defined FOLDERs, automatic REPLYing of messages, FORWARDing of messages and DELET-ing the incoming message.

The defined set of rules that represents the user's interest at that moment is saved into a user profile and stored in the server database. Rules are first syntactically and semantically checked before they are converted from their defined infix form into postfix form. These rules are verified and tested for rule conflicts that must be subsequently resolved before being stored in the user profile. Security and privacy are maintained for all user profiles in the server. The same user who has access to different servers can have different user profiles in different servers.
MEmail utilises a layered extendible architecture that serves as the interface between mail-enabled applications and system services providers as shown in Fig. 8. Although, this architecture is unnecessarily more complicated for satisfying MEmail's requirements, it is nonetheless developed with future extendibility in mind. Basically, the architecture is divided into two layers:

- **MEmail Application Programming Interface (API) layer** that caters for the functions directly involved with any mail-enabled applications. These APIs are implemented as class libraries to provide a set of generic multimedia and mail services.

- **Service support layer** that includes the storage support, transport support and media support. This layer facilitates MEmail APIs to directly link to the services provided by the system.

### 5. Current Status and Future Work

The prototype of MEmail is completed and currently undergoing end users' evaluation. MEmail offers storage management and transportation of messages across different computer platforms within NTU. The Mail and Media Servers are implemented on SUN Sparc workstations. The Versant object-oriented database system is used as the Object Manager. All the major functional components of the UA (i.e., client machines) are implemented on both Unix workstations and PCs. This research has demonstrated the successful exchange of multimedia messages between Unix workstations and PCs that have significant hardware and data format incompatibilities.

Users' preliminary informal indicators reveal that most of the filtering rules are predominately based on information of senders or the groups to which the senders...
belong. Keyword filters are also used to tap into general research interests and used as a means to keep users informed of the developments of other research groups within NTU. Filters have shown their usefulness in discarding “junk” messages. However, users are still cautious to ensure that genuine messages are not discarded accidentally due to poor rules definition.

The query mechanism provides a simple but powerful means to allow the user to recall kept mail in various folders. The availability of pattern matching, wildcard facility together with the provision of relevance feedback enhances the query formulation and retrieval.

The layout of the main user interface facilitates ease of navigation to initiate various functions to perform different tasks. The integral query and retrieval area in the control panel allow immediate access to such a facility. The retrieved message area also serves as the new mail notification area. Users can click on the message listing to read new mail or browse old mail. The consistency in layout and operations helps reduce the learning curve of new users to the system.

Further work falls into two categories. First, is a need to accumulate more user experience to provide a means to further confirm the functionality of the system, the added effectiveness provided by the filtering and querying facilities, and to incorporate improvements as a result of users’ feedback.

Second, more study into the IR model used for query and retrieval can be carried out. A set of separate controlled tests to allow recall and precision measures to be obtained would prove useful in evaluating the IR model and ideas used in the filter and query tool. Some preliminary work on fuzzy retrieval methods utilising a fuzzy thesaurus for IR retrieval has already been explored. In future, the IR model can be upgraded to allow even more sophisticated filter and query facilities to be incorporated into the system.

6. Conclusions

In this paper, we have identified the key success factors of a new generation of multimedia email and demonstrated how these are realised in the design and implementation of ME mail. In summary:

- ME mail is capable of functioning over a heterogeneous environment;
- ME mail is able to provide a good message handling and storage mechanism for multimedia email by making the system accountable for the overall handling and storing of messages for better repository utilization. ME mail achieves the storage of only one single copy of a shared message with the object-oriented approach in design.
- ME mail can be configured to handle new media types (such as tables, formulae, etc.) by making use of MIME to handle such extensions by incorporating them as attachments to the mail. These attachments can be identified and as a result, the
system is able to spawn external processes or programs to manipulate and process these new media types.

- MEmail can present only relevant and important mail to the users while discarding other forms of junk mail. This is made possible by the filtering mechanism that will intercept new incoming mail and compare them with the rules defined in the user profile and to take the predefined action as a result of a condition match.
- MEmail provides a set of resource management tools to aid the management, querying, browsing and retrieval of messages. The query tool provides the facility to help users locate and retrieve messages within the mailing system effectively. As the number of kept mail grow, so will the usefulness of having such a facility.
- MEmail provides a "good" user interface that has been designed for both novices and experienced users in mind to allow easy access and use of the system's facilities.

A layered approach and the use of the object-oriented paradigm for data modelling and database support have been utilised in the design and implementation of MEmail's architecture. Such an architecture exhibits the advantages of independence, maintainability, flexibility and extendibility. It allows improvements in information query and retrieval techniques to the incorporated into the system with greater ease. In addition, this architecture also forms a basic platform for other message-enabled applications to be built (e.g. CSCW applications, computational email).

Acknowledgments

The authors would like to thank the following people who have helped in one way or another in the development and testing of MEmail: Wu Hua Horng, Tan Poh Kean and Loke Teng See.

References


