Delivery of Video Mail on the World Wide Web

Schubert FOO & Siu Cheung HUI
Division of Software Systems, School of Applied Science
Nanyang Technological University, Singapore

Summary

This study is initiated to examine the feasibility of providing multimedia mail, in particular video mail, on the existing World-Wide-Web platform. In this instance, the Web browser will act as the user agent of the mailing system. With this, users will be able to compose, send and read video mail directly on the Web. This paper examines a number of possible system architectures that can be employed for the development of Video Mail. Based on the following key issues: portability of system, flexibility of system, Internet standard support, security and authentication, and message compression and storage, a final architecture was selected to build Video Mail. The approach taken is that the system will be interfaced to the Web using a standard HyperText Transfer Protocol (HTTP) server and a standard Web browser. This is achieved through the use of Common Gateway Interface (CGI) programs and forms which reside in a standard HTTP server to provide the functionality of video mail, and a separate mail server for the storage and management of messages. This will cause the HTTP server to behave as if it is an electronic mail system.

Video Mail has been successfully implemented at the School of Applied Science, Nanyang Technological University, Singapore. The development work has been carried out on a Sun SPARC5 workstation and the Typhoon relational database system was used to construct the mail server. All software codes were written in C or C++. A video capturing application used in conjunction with a SunVideo card has been developed for video and audio recording. The paper also addresses implementation issues pertaining to the use of CGI programs, mail server development, and video and audio management.

1. Introduction

Internet has gained widespread popularity with the advent of the World-Wide-Web (Web for short) technology in 1992. Growth in Internet has been exponential since then as Web browsers such as NCSA's Mosaic [1], Netscape's Navigator [2] and Microsoft's Internet Explorer [3] have enabled non-technical users to surf the net with a minimal amount of computer knowledge. Original Internet facilities such as ftp, electronic mailing, News, Gopher and so on have also become more accessible and more widely used. Although many multimedia mail systems [4-9] are developed and marketed, Internet's Web-based electronic mailing is still confined to ASCII-text only mail today. With the popularity of the World-Wide-Web and its ability to manipulate multimedia information, there is a widespread interest in combining the functionality of the Web with that of electronic mail.

This study is initiated to examine the feasibility of developing an integrated multimedia mail system (in particular video mail) on the existing World-Wide-Web platform. In this instance, the Web browser will act as the user agent of the mailing system. With this, users will be able to compose, send and read video mail directly on the Web.

2. The World-Wide-Web and Related Protocols

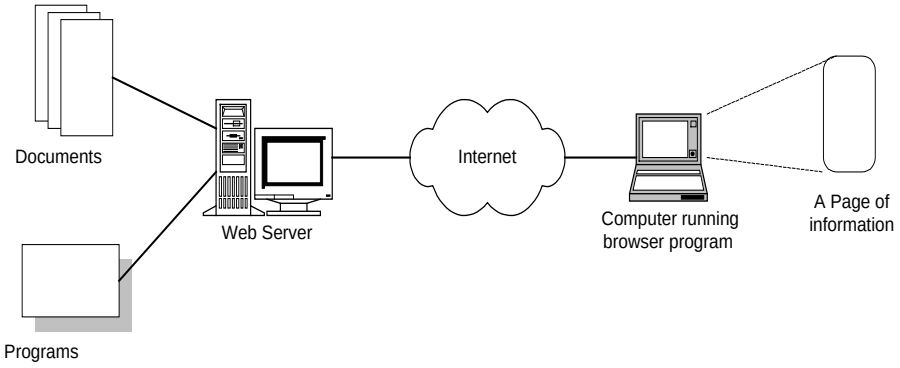

Figure1. The World Wide Web Architecture

The World Wide Web is a wide area hypermedia information retrieval initiative aimed to give universal access to a large universe of documents. It is organized as a set of HyperText Transfer Protocol (HTTP) [10] servers designed specially for rapid distribution of

hypermedia documents as shown in Figure 1. HTTP connections over the Internet are implemented using the TCP/IP [11] protocol. When a user wants to access the Web, a Web browser program (such as Mosaic, Netscape, or Internet Explorer) must be activated at the client machine. The browser will access the user-defined web site through the Internet. A web server at that web site will retrieve the information requested. The web server could be a NCSA server [12] or HyperWave server [13] which is a second generation server. The information is then transmitted back to the browser for display to the user. The documents retrieved can contain hyperlinks that point to other nodes. Nodes can take the form of another section of the same document or to another new document. When a hyperlink is activated (usually by clicking the mouse button), the system extracts its destination node and loads the corresponding document as the current document. Users navigate around the hyperspace by moving from link to link, or by specifying direct links.

The HyperText Markup Language (HTML) [14] is a hypermedia language used to construct WWW documents. It is designed to specify the logical organization and formatting of text documents, with extensions to include inline images, audio, video clips, fill-in forms and hyperlinks to other HTML documents and other Internet resources (such as files, ftp, USENET, telnet). Each WWW resource has a unique address known as the Uniform Resource Locator (URL) [15].

The capabilities of a Web server can be extended by allowing developers to develop their own set of server-specific Application Programming Interfaces (APIs). Common Gateway Interface (CGI) [16], the Internet standard interface for invoking server-based scripts or compiled programs at the requests of clients is currently the most common way of providing clients interactive and dynamic content and make the browser come æaliveÆ. CGI programs are special in that they do not receive their arguments from the command line but from special variables known as CGI variables that are generally passed to the program as environment variables.

3. Related Work

This section examines a number of categories of related work starting with multimedia mail developments. Subsequently, it examines existing Web browsers that provide mailing capabilities and eventually systems that can deliver real time multimedia functionality on the Web.

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. However, many of existing multimedia mail systems such as Andrew

Message system [4], Diamond system [5], Montage System [6], Microsoft/MSMail [7] and NeXT/MediaView [8] are homogeneous systems which require the same make of computers to be used in order to exploit multimedia capabilities. They are proprietary in nature and have been successful in small local user communities. A prototype of a new generation of multimedia mail system, MEMail [9], has been developed by the authors which can function over an heterogeneous environment.

MEmail 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 makes use of mail and media servers and object-oriented databases to process, store and manage the different forms of media data. It uses MIME [17] and SMTP [18] for the mail format and mail transport respectively. A production rule-based system allows users to filter incoming messages and keep only the ôrelevantö messages. It contains a query and retrieval capability for managing kept messages. The reader can refer to [9] for a complete description of the MEmail system. This system has provided important considerations that have been factored in the design of Video Mail.

Currently, one of the best well-known and used Web browsers, Netscape Navigator 2.0 [2] exhibits ASCII text-only mail capability to allow users to exchange electronic text messages. In this system, the browser is initially used to view Web pages. When mailing is required, the browser will activate its built-in mail reader for mail composition. When ready to be delivered, it will communicate with an Internet mail server using an Internet mail protocol (SMTP/POP3 [19]). This means that users will be able to exchange messages on many existing Internet mail servers. It supports hyper-links within mail messages so that the sender is able to include links to Web pages in the message. This adds a whole dimension to the way electronic messages behave. Being a text-based mail facility has inhibited it from including multimedia in the message. Inclusion of multimedia is currently achieved through the use of attachments. In order to process the attached media, it must first be extracted, stored and subsequently read with an appropriate reader.

Vosiac, or *Video Mosaic* [20], is a Web browser that incorporates real time video and audio into standard hypertext pages and which are displayed in place. It extends the architecture of the Web to encompass the dynamic real time information space of video and audio. The system is developed by modifying the Mosaic Web browser developed by NCSA[1] to incorporate the on-the-fly decoding of audio and video. The syntax of HTML was extended to allow the integration of audio and video into standard HTML pages. It uses a proprietary protocol VDP, an augmented Real Time Protocol [21], to handle real time video over the Web. The authors claim that such a protocol reduces inter-frame jitters by

dynamically adapting to the client CPU load and network congestion. Hence, video and audio transfers can take place in real time without file retrieval latency.

VDP is an asymmetric protocol connecting a client and a video server. Two channels are associated with each VDP connection. The first is an unreliable data transmission channel used for transmitting data and uncritical feedback information. The other one is a reliable channel to send control information from the client to the server.

The main advantage of *Vosaic* is an integrated multimedia WWW browser that supports real time playback of video and audio data on-demand with no quality of service guarantees. However, the system requires the specialized browser and the extended server in order to support the real time transfer of video and audio data. In addition, *Vosaic* only provides asymmetrical transmission of video and audio data. No provision is available for the capturing and transmission of the data from the client to the server. Therefore, it is not totally suitable for the symmetric nature of Video Mail. Furthermore, as Video Mail is asynchronous is nature, there is no requirement for a real-time protocol unless real-time playback on-the-fly is required.

4. System Design and Application

4.1 Key Issues in System Design

This work aims to extend the current World-Wide-Web architecture to incorporate integrated electronic mailing with multimedia capability (video, in particular) with the consideration of the following key issues:

Portability of system. As the Internet is a heterogeneous environment that interconnects a wide range of computer platforms, the system designed should be able to cater this heterogeneity and be adaptable to as many platforms as possible

Flexibility of system. The World-Wide-Web is a rapidly moving environment with new tools and standards which are continuously evolving [22] (for example, Java, JavaScript, Secure HTTP through Secure Sockets Layer (SSL), LiveMedia, Live3D, Virtual Reality Modeling Language (VRML), and so on). The new system should be easily modifiable to cater for future technological changes.

Internet standard support. The frenetic growth of the Internet and related applications owe a large part to open Internet standards. At such, there is a need for the system to be designed to follow closely to these existing standards to ensure interoperability.

Security and Authentication. As in all electronic mail systems, security and authentication is necessary to ensure that mail is secure and that it reaches the recipients safely. HTTP, which is a connectionless protocol, poses a special problem since there is difficulty determining who are the legitimate users. For this reason, Secure Sockets Layer (SSL) has been developed to support commercial and sensitive transactions. In addition, HTTP does provide mechanisms to protect the accessibility of documents to users.

Message Compression and Storage. As the system has to handle video information that requires substantial storage of its image and audio data, some form of data compression and efficient storage and management of these messages is necessary.

4.2 System Architecture

A number of different system architectures which allow the extension of the current Web system to incorporate Video Mail were derived and gauged along the various issues identified in the previous section. A final system architecture was chosen based on the analysis on the key issues and the authors Æexperience in developingMEmail.

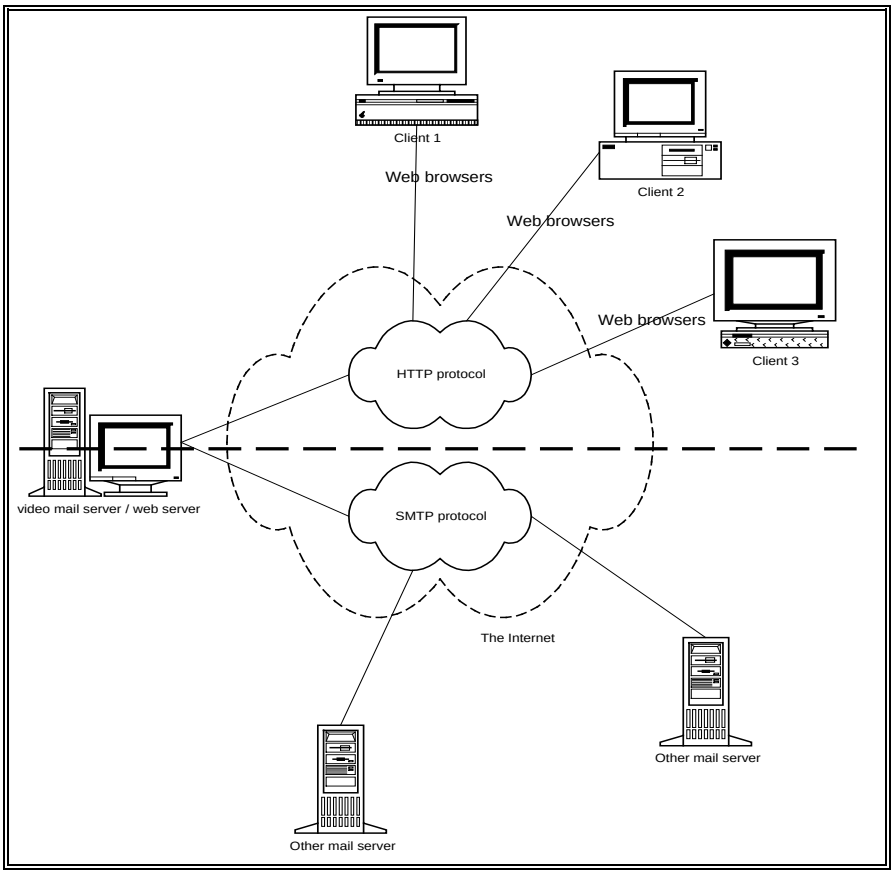

Figure 2. Video Mail Architecture: Integrated Approach

The first design as shown in Figure 2 adopts an integrated approach to combine the mail server which is responsible for mail storage, and incoming and outgoing mail processing together with the Web server as one complete unit. The standard SMTP protocol is used for inter-server communication. The advantage of this design is the combination of both functionalities into a single service. This means that there would not be a need to create a series of CGI programs but a larger program that resides on the server. From the system administrator's viewpoint, building the system this way is cleaner and more well-defined as there is the need to deal with one single program rather then a number of CGI programs. In addition, designing the system this way means that additional programs can be interfaced directly to the server without following the CGI protocol. This can result in the more efficient use of resources.

However, the design does have its disadvantages. This design does not exhibit much flexibility in the system. If changes are to be made, the whole system have to be recompiled again. Without the CGI support, the system cannot be used for other purposes without knowledge on how the server was written. For example, if there was a need to include a

newsgroup capability to the system, the developer will have to understand how the system was written before embarking into designing and incorporating the new capability. Another inherent difficulty is the nature of the HTTP protocol that is connectionless in nature while an electronic mail system is a connection-oriented system.

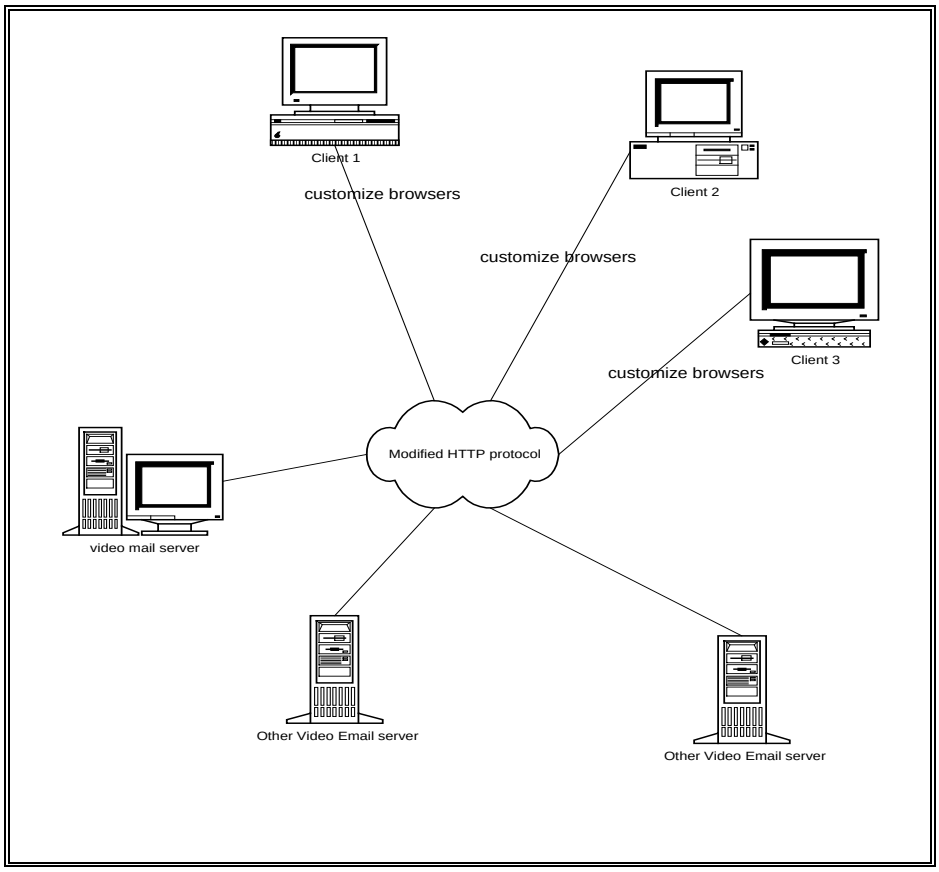

Figure 3. Video Mail Architecture: Modified HTTP Approach

In the alternative design shown in Figure 3, the SMTP protocol for inter-server communication is not used, but a modified version of the HTTP protocol is used instead. This modification is necessary to convert the HTTP protocol into a connected-oriented protocol. A new class of browsers must also be written to communicate with the mail server using the new modified HTTP protocol. This browser can keep track of what the user is doing at any point of time and call up necessary services in the server. Another feature of this design is elimination of the need for the SMTP protocol. At such, the system is no longer limited by the 7-bit data restriction of SMTP. The system can make use of the

modified HTTP protocol directly which will have features to cater for multimedia transfer and result in reduced programming efforts.

As in the first design, the second design also suffers from inflexibility. It introduces a new protocol that does not conform to any standard as yet. This is likely to result in usersÆ buy-in and interoperability problems. At such, it may encounter difficulties in communicating with other electronic mail systems unless a gateway is used. In addition, both designs require the implementation of user authentication and security features that are complex issues by themselves.

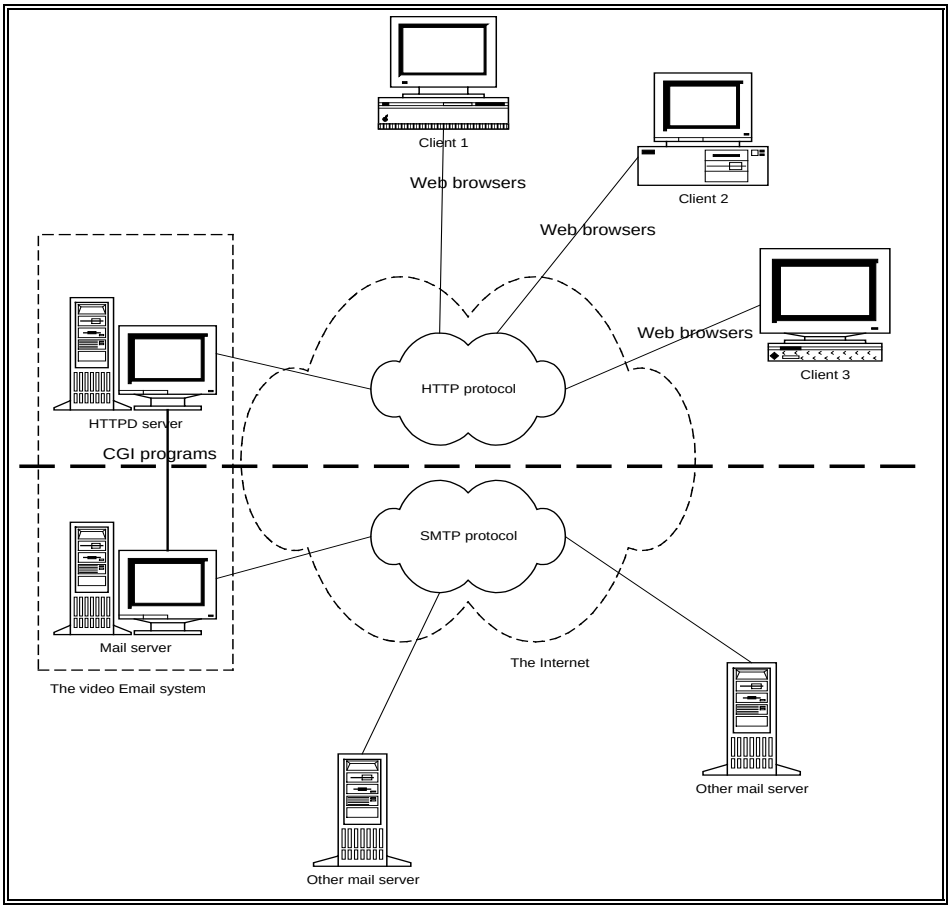

Figure 4. Video Mail Architecture : Proposed Architecture

The final design of Video Mail, which was eventually adopted, divides the architecture into two sections as shown in Figure 4. The upper section deals with the interaction between the client and the server while the lower section deals with inter-server communication. The upper section deals with how the user interacts with the system to

retrieve a mail. The approach taken is that the system will be interfaced to the Web using a standard HTTP server and a standard Web browser. Residing in the HTTP server is a set of CGI programs that will link the HTTP server to the mail server and cause the HTTP server to behave as if it is an electronic mail system. On the client machine, there will be an external program to act as a mail reader to view the video.

The lower section shows how the mail server interacts with other mail servers on the Internet. The design is basically the same as a standard Internet electronic mail system. In such a system, the functionality of the mail server is to store and transfer messages. The difference between this and other mailing systems is that it interacts with the user agent via a set of CGI programs.

This architecture exhibits portability and flexibility by allowing existing systems (Web servers and browsers) to be used without any modifications. It conforms to Internet standards thereby allowing the seamless exchange of messages between other SMTP servers. The system is modular as changes to the system are confined to the affected CGI programs. As the HTTP server is linked to the mail server by a set of CGI programs, it becomes extensible and adaptable to any Web browsers. In addition, new CGI programs can be written to add functionality to the system. Since the system is based on standard HTTP protocols, it can make use of its existing built-in authentication and security features directly in Video Mail.

This final design is not without drawbacks. In order to overcome the connectionless nature of the HTTP protocol, a fixed set of pre-defined CGI programs must be executed in order to handle and keep track of usersÆactions. If one of these CGI programs is missing or wrongly placed, the system will not be able to function. In addition, the system is more difficult to maintain due to the large number of small CGI programs in the system.

4.3 Functional Components

A first working prototype Video Mail based on the final design architecture has been successfully implemented at the School of Applied Science, Nanyang Technological University. The home page of Video Mail is shown in Figure 5. The development work has been carried out on a Sun SPARC5 workstation and the *Typhoon* relational database system [23] was used in the construction of the mail server. All software codes were written in C or C++. A video capturing application used in conjunction with a SunVideo card is developed for recording the video and audio inputs.

Figure 5. Home Page of Video Mail

The home page shows the main menu of Video Mail and a list of functional operations. Progression graphs are used to keep track of the sequence of operations of the CGI programs (see next section) for every operation and cause the system to behave as an integrated system. Four main operations are supported in Video Mail:

New Mail Notification. This is used to notify users of new incoming messages. A hierarchical tree structure is used for folder management and support. Different icons are used to denote folders and mail. New folders are interactively created by users. Hyperlinks are used for folder navigation. Such a system facilitates structured mail organization.

Mail Reader. This is used for message management and browsing. As in folder management, navigation around mail messages is achieved through hyperlinks. HTML commands can be embedded in messages so that they can in turn contain hyperlinks pointing

to other Web pages and sites. A separate icon is used to denote the presence of a video clip in the mail. Clicking on this button will activate an external viewer program at the client machine to view the video clip. Facilities for replying and deleting existing messages are supported. Figure 6 shows the mail reader for browsing messages and viewing video.

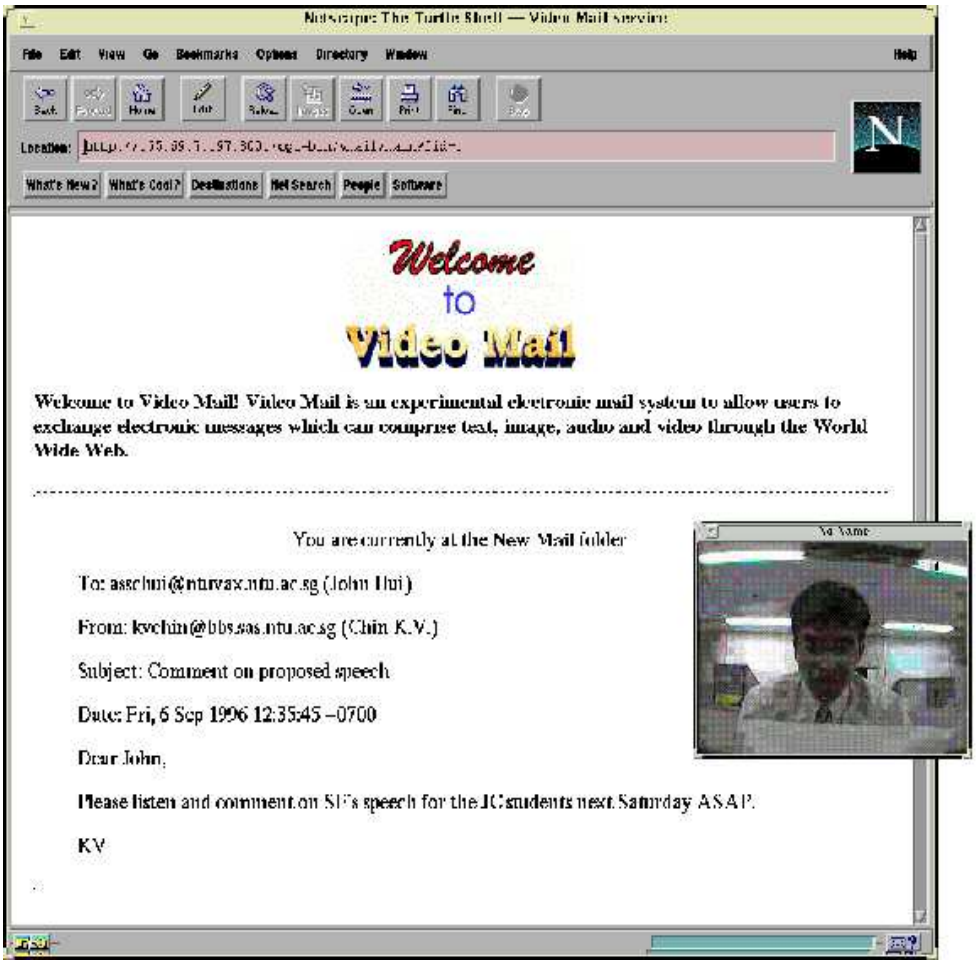

Figure 6. Mail Reader

Mail Authoring. This is used for message composition and delivery. A CGI-form is displayed for the user to input the message content. HTML commands can be included as part of the message. A checkbox is provided for the user to attach a video clip together with the message. The video clip must first be created, if it does not already exist, using a video capturing system. When a message is ready to be sent, the mail server will check for the existence of a video attachment and corresponding transfer the video from the client machine to the mail server, carry out the necessary processing and route the complete message to its destination. Images can be inserted into the mail body through the use of hyperlinks. This

implies that the user would have access to some Web server to store these images before it can be used. This follows HTML conventions for inserting images in HTML documents.

General Utilities. A change password utility exists for password management and an address book utility exists and acts as a personalized address book for users to store a list of their most-commonly used email addresses.

4.4 Implementation Issues

CGI Programs. In the use of CGI programs in Video Mail, the important consideration is the order in which these programs are presented to the user such that it behaves as a single integrated program. As an illustration, consider the situation of the user logging into the system and sending an electronic mail as shown in Figure 7. In this figure, the shadowed-rectangle represents a CGI program and the round rectangle represents the output page on the Web browser

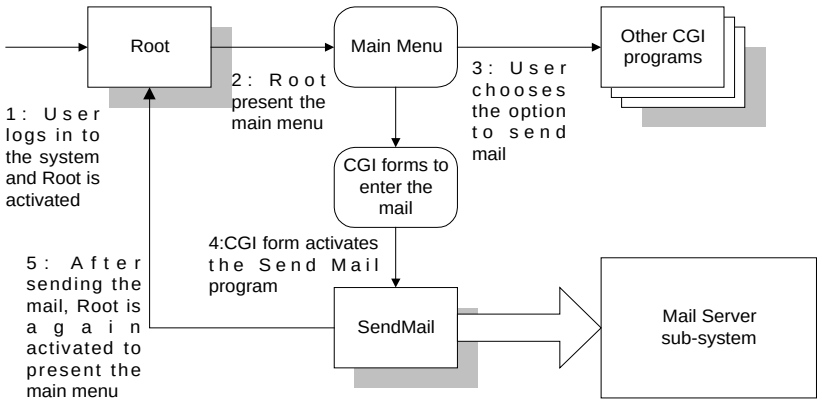

Figure 7. Logging in Video Mail to Send an Email

When a user first logs in to the system, the HTTP server will activate a CGI program (say Root) which will present the user with a list of options of the system. These options are hyperlinks that point to other programs or pages. When the user selects the SendMail option, it activates a CGI-form for the user to enter the text message and optionally, attach a video. When the Submit option is chosen at the end of mail composition, it activates the SendMail

program to transfer the message to the mail server. It subsequently activates Root to present the user with a list of options again.

Therefore, in the design of the CGI programs, there is a need to identify the sequence of actions that can take place and derive the state-diagrams of these actions. With these identified and the various outcomes known, it becomes a matter of defining the various CGI programs to carry out the various tasks.

Mail Server. The mail server as shown in Figure 8 comprises three components for message storage and processing: *Receive Folder* area which holds all incoming mail for the recipients of the mail server, *Send Folder* area which holds all outgoing mail created by users of the system, and the *Mail Storage* facility to store mail messages and provide support for different types of mail data. The CGI programs interface directly with the mail storage for the retrieval of messages from the mail storage and pass outgoing messages to the send folder.

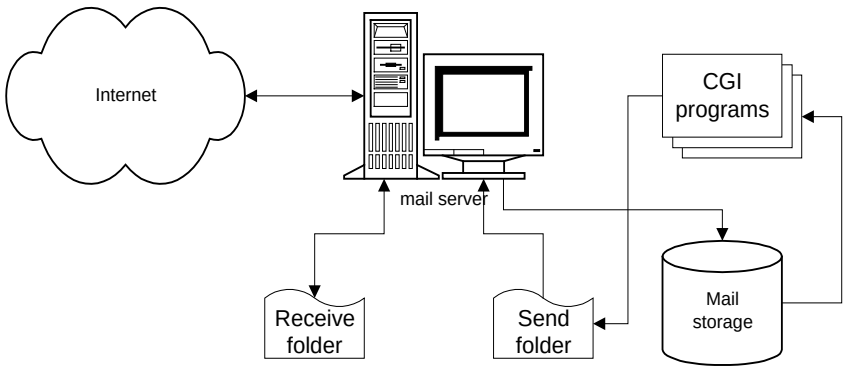

Figure 8. Mail Server

A mail daemon (which is basically the mail server program) interacts with the receive folder and inserts the corresponding data into the mail storage. Outgoing messages, depending on internal or external recipients, are automatically routed to the receive folder or sent out directly using SMTP respectively. All outgoing external messages conform to the MIME format. In addition, the daemon also processes all incoming messages via SMTP and stores them into the Receive Folder area. This mail server architecture is flexible and allows extensibility to support other standard Internet mail servers using SMTP/POP3 protocols. In this instance, the daemon must be extended to support the POP3 protocol to download messages to the video mail server.

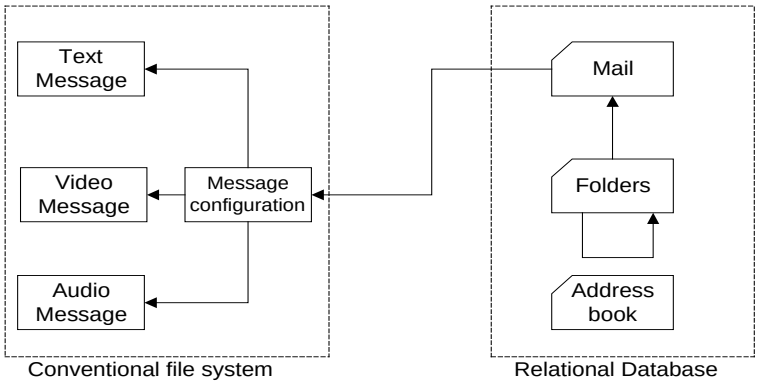

Figure 9. Mail Storage in Video Mail

In the mail storage design shown in Figure 9, a conventional file system is used to store the message body while a relational database is used for storing folder information, mail pointers (to the message configuration file) and address book of all users. The message content can include text, video, and audio. The purpose of the message configuration file is to keep track of a multi-part multimedia message whose individual components are stored as conventional files. In addition, the file contains other information including the sender name, time routing information and recipients' names. Such a design facilitates sharing of information so that only a single copy of the message is kept by the server and shared by all recipients.

Video and Audio Management. In order to design the video functionality in Video Mail, three issues must be addressed. The first is to decide how the individual video and audio bitstreams are to be constructed and decoded. The second is to determine how these individual compressed streams are encoded into a suitable form for inclusion into a mail for transmission. The final issue relates to how these individual streams are to be synchronized during playout of the video.

Although many existing compression and encoding algorithms exist for video and audio, we have analyzed the alternatives and chosen to compress both video and audio signals into a single stream defined by the MPEG-1 standard [24] based on the criteria of availability, efficiency, nature of application, quality of the reconstructed signal, acceptance and recognition. The compressed streams are multiplexed together with time dependencies as a single stream. This time dependency information is required for the synchronization of

video playback. The single stream, although more computationally demanding, exhibits two advantages. The first advantage stems from having a single stream that will reduce the interface complexity with the mail server. The second advantage is that real-time decoding of streams is more easily achievable for a single stream since there is no need to wait for the successful arrivals of separate data.

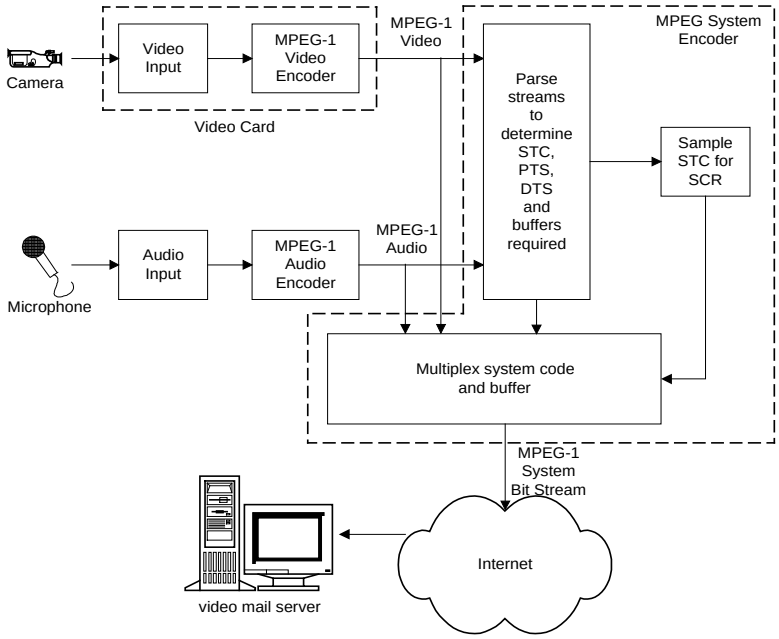

Figure 10. Video Composition Process

Figure 10 shows the process of video composition into a single bit stream (file) prior to transmission to the mail server. The video and audio input is captured using a camera and microphone. The video signals are directly converted into a MPEG-1 video stream from the video card while the audio signals are captured in PCM format and subsequently fed into the MPEG-1 audio encoder to yield MPEG-1 audio format. The compressed video stream uses the following frame sequence: I B B P B B P B B I. (I frames are intra-frame coded with JPEG compression; P frames are predictive codes with respect to a past picture; B frames are bidirectionally predictive coded [25]). The choice for this pattern is aimed at seeking an optimal trade-off between the amount of compression and the granularity of random access points. A larger number of B and P-frames will result in better compression ratios while the number of I-frames determines the number of random access points. The I frame is needed by all P and B frames in order to be decoded. This frame sequence will allow for more

gradual degradation of the playback in case of insufficient resources (such as bandwidth or computation powers). Audio is captured at the lowest rate allowed by MPEG-1 of 32 kHz in order to reduce bandwidth requirement of the mail.

These two streams are passed to the MPEG-1 system encoder which will process the streams to determine the Presentation Time-stamps (PTS) and Decoding Time-stamps (DTS). The system time-clock (STC) and the buffers required to decode the streams (STD) are also obtained. Following this, the STC is sampled to get the system clock reference (SCR) values. Finally all information are encoded into a system stream (file) together with the multiplexed audio and video streams in accordance to the MPEG-1 system format. This stream is the final form used for transmission.

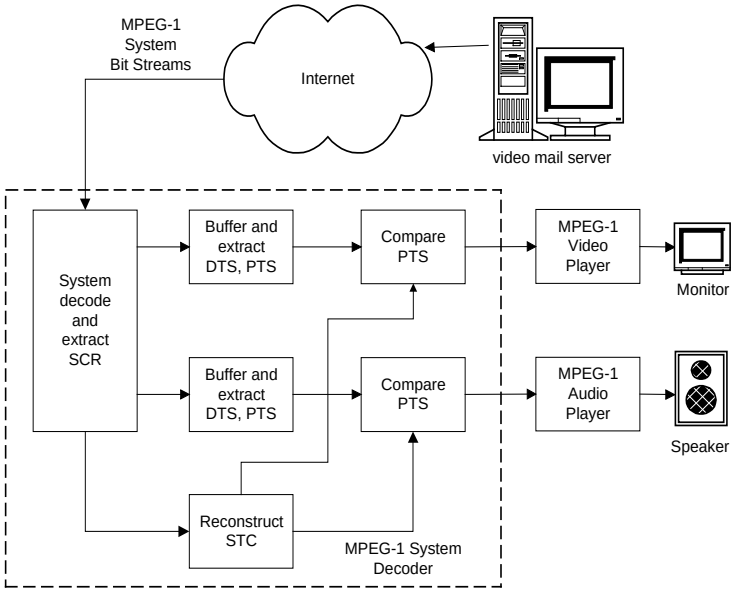

Figure 11. Video Playback Process

Figure 11 shows the reverse process of playing back a MPEG-1 system stream via a decoder. In the decoder, the various streams are buffered for handing over to the respective stream decoders. The PTS and DTS are extracted. The PTS is compared with the reconstructed STC to determine the time to pass the buffer over to the individual MPEG stream player. Finally, the audio and video streams are decoded and played back on the speaker and monitor respectively.

5. Conclusions

This work has successfully demonstrated the feasibility of extending the current World-Wide-Web environment to incorporate the handling of multimedia mail, namely, Video Mail. This has been achieved by using a set of CGI programs to provide the basic functionality of the mail system. Apart from using Video Mail as a stand-alone application, the system can also be incorporated into Internet-based co-operative groupware systems such as virtual meeting systems and workflow management systems.

As this is the first working prototype, there exists much possibility for future enhancements and added functionality. As continuing work, we are examining the provision for on-the-fly video playback on Web browsers in order to reduce the transfer latency for continuous media files. Such extensions can only be applied to browsers that support incremental displaying of documents with the ability to stream data via external ôhelper applicationsö (such as NetscapeÆs Navigator). In addition, the use of Java [26] for the implementation of Video Mail is currently under investigation.

References

1. NCSA, *NCSA Mosaic WWW Browser*. <URL: http://www.ccs.org/winsock/mosaic.html>
2. Netscape Communications, Inc. *Netscape Navigator Gold 2.02 WWW Browser*. <URL : http://home.netscape.com/comprod/products/navigator/version_2.0/index.html>
3. Microsoft Corporation, *Internet Explorer*. <URL: http://www.microsoft.com/ie/ie.html>
4. N. S. Borenstein, C. F. Everhart, J. Rosenberg & A. Stoller 1989. Architectural Issues in the Andrew Message System. *Message Handling Systems and Distributed Applications*, North Holland: E. Stefferud, O. J. Jacobsen & P. Schicker Eds. 471-487.
5. R.H. Thomas *et. al.* 1985. Diamond: A Multimedia Message System built on a Distributed Architecture. *IEEE Computer*, 65-78.
6. W. KeithEdwards 1990. The Design and Implementation of the MONTAGE Multimedia Mail System. *IEEE Computer*, 47-57.
7. Microsoft Corporation 1989. *Microsoft Mail User's Guide.* Apple Macintosh Series, Redmond, Washington.
8. R. L. Phillips 1991. MediaView: A General Multimedia Digital Publication System. *Communications of the ACM*. **34(7)**, 75-83.
9. S. Foo, S. C. Hui, C.J. Daroy 1996. MEmail - A Heterogeneous Multimedia Electronic Mail System, *International Journal of Information Technology*, **2(1)**, 55-77, World Scientific Publishing Company.
10. World Wide Web Consortium. *HyperText Transfer Protocol*. <URL: http://www.w3.org/hypertext/WWW/Protocols>
11. D. Comer & D. Stevens 1991. *Internetworking with TCP/IP*, Prentice Hall, Englewood Cliffs, NJ.
12. University of Illinois. *NCSA HTTPd Server*. <URL: http://hoohoo.ncsa,uius.edu/docs/overview.html>
13. H. Maurer 1996. *HyperWave - The Next Generation Web Solution*, Addison-Wesley.
14. World Wide Web Consortium. *Hypertext Markup Language*. <URL: http://www.w3.org/hypertext/WWW/Markup>
15. World Wide Web Consortium. *Uniform Resource Locators*. <URL: http://www.w3.org/hypertext/Addressing/URL>
16. World Wide Web Consortium. *Common Gateway Interface*. <URL: http://www.w3.org/hypertext/WWW/Overview.html>

17. N. S. Borenstein & N. Freed 1992. MIME - Multipurpose Internet Mail Extensions: Mechanism for Specifying and Describing the Format of Internet Message Bodies. *Network Working Group Request for Comments: 1341*.

18. J. B. Postel 1982. *Simple Mail Transfer Protocol*, Internet Request for Comment (RFC) 821.

19. M. Rose 1993. Post Office Protocol, Version 3. Internet Request for Comment (RFC) 1225.

20. Z. Chen, S. Tan, R. H. Campbell & Y. Li. Real Time Video and Audio in the World Wide Web. <URL : http://choices.cs.uiuc.edu/Papers/New/Vosaic/vosaic.html>

21. H. Schulzrinne & S. Casneer 1993. RTP: A Transport Protocol for Real Time Applications, Interent Draft.

22. Netscape Communications Corporation, *The Internet Application: A White Paper*, <URL: //home3.netscape.com/comprod/server_central/tech_docs/oif.html>

23. T. B. Pedersen. typhoon 1.10.3 relational database management system, <URL: http://hpux.cs.utah.edu/hpux/Users/typhoon-1.10.3.html>

24. ISO/IEC International Standard 11172-1, ôInformation Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/sö, in 3 parts: Part 1 : System, Part 2: Video, Part 3: Audio.

25. D. E. Gall 1991. MPEG: A Video Compression Standard for Multimedia Applications, *Communications of the ACM*, **34(4)**, 46-58.

26. P. Wayner 1997. *Java and JavaScript Programming*, AP Professional, Boston.