Imagine sitting at your desk at home and checking your electronic mail. An audio-visual clip of your family members hundreds of miles away pops up with warm greetings and family news. This scenario lies within the capability of the integrated multimedia mailing system, called VistaMail, developed at the University of Illinois at Urbana-Champaign. VistaMail provides smooth and seamless transitions among the creation, transmission, reception, and display of e-mail messages composed of audio, video, and text. It integrates the entire process—from the creation of audio, video, and text by the sender to its final synchronized presentation to the receiver. While other mailing systems might support video and audio media as attachments, they do not offer the same integrated capability.

VistaMail messages represent a fundamental shift in how people perceive e-mail messages. Video and audio are no longer considered special additions to a text-only e-mail message. Instead, the combination of text, audio, and video constitute the e-mail message. People often find it difficult to gauge the tone of a text-only e-mail message. Many people have used emoticons (smilies) to convey their emotions. Audio and video can convey true emotions accurately without emoticons. The integrated nature of VistaMail brings a more personal touch to e-mail messages.

This article presents the design and implementation challenges that arose during the development of VistaMail. We explore issues concerning media processing, mailing, and management functions within the system architecture, services, and graphical user interface. Finally, we discuss user experiences with the system based on our observations.

Related work

Text-based e-mail systems have become commonplace in our computing environments. These systems build on the X.420 standard or the Simple Mail Transport Protocol (SMTP) and Multipurpose Internet Mail Extensions (MIME) standards. Although the X.420 standard was expanded to support videotex and telefax, it does not include any specification for multimedia message types such as audio and video. In contrast, MIME can attach message types such as audio, video, and images to the body of text e-mail messages sent by SMTP.

Support for composition and playback of multimedia messages requires extensions to traditional text-based systems. Most current e-mail systems need external applications so that users can view the multimedia components. As audio and video messages become more common, increased support for these message types will become available.

Many of the systems described below are external players or recorders, which cannot send messages themselves. As a result, the process of sending and receiving multimedia messages becomes difficult and inconvenient. For example, a video or audio message must first be recorded with the external application and then saved to the hard drive. An existing e-mail package then attaches and sends the saved file. This inconvenience may discourage users from using such systems.

Current e-mail systems have traditionally taken a decoupled approach to including non-text components in mail messages. This has let e-mail package developers focus on the text and transport of messages, while letting separate applications handle specific attachments. This approach might work for a range of attachment types. However, if audio and video are included as an integral part of the communication, then the capture and playback mechanism should be incorporated into the e-mail system design. The audio and video components are no longer considered separate from the text and should not be handled that way by the mail browser.

Current work that supports audio and video in e-mail user agents generally falls into one of four categories, as follows:
**Multimedia E-mail**

For more information about the commercial e-mail systems mentioned in this article, please visit the following Web sites:

- **Howdy! ME-Mail** by Magicbit
- **See-mail** by Realmedia
  - http://www.realmedia.com
- **PureVoice Player-Recorder** by Qualcomm
  - http://www.eudora.com/purevoice
- **VideoLink Mail** by SmithMicro Software
- **Voice E-mail** by Bonzi Software
- **Video Express E-mail** by ImageMind Software
  - http://www.imagemind.com
- **BravoMail** by Softlink
  - http://www.bravomail.com
- **Vmail 1.5** by NetActivity
  - http://www.netactivity.com

### Capture and playback tools with no mailing capability

Several applications, separate from the e-mail user agent, record and/or play back audio and video messages. These applications create multimedia files that require an existing e-mail system to transport the messages as MIME attachments. Examples include Magicbit’s Howdy, Realmedia’s See-mail, Qualcomm’s Pure Voice Player-Recorder, SmithMicro’s VideoLink Mail, and Bonzi Software’s Voice E-mail (see the sidebar “Multimedia E-mail”).

### Streaming tools

ImageMind’s Video Express E-mail supports the composition of multimedia messages with streaming technology for delivery. This system records audio and video messages and sends them to a central message store. The receiver uses an existing text-based mailing system to receive an executable attachment, which initiates the streaming process for the playback of messages.

### Presentation tools

Several researchers have developed applications for the synchronized presentation of various multimedia forms. MHEGAM8 integrates an MHEG (Multimedia and Hypermedia Experts Group) editor into the mailing systems to let a user compose and transfer multimedia messages consisting of prerecorded components. The MHEG editor allows the creation of hypermedia documents with proper synchronization among audio, video, graphics, images, and text. Softlink’s BravoMail offers composition and playback capabilities for audio, video, and animation. The system provides a tool to create the media forms that are packaged into a single executable file. An existing e-mail system is required for transporting the executable, since no direct support for message delivery is available.

### Integrated capture and mailing tools without playback

NetActivity’s Vmail 1.5 integrates recording and transport capabilities into one application. However, while it lets users record and send QuickTime format video messages, it does not receive messages. An existing e-mail application is necessary to receive the message, in addition to a QuickTime movie player to view the attachment.

In contrast to the systems above, VistaMail fully integrates the capability to record, play back, send, and receive audio and video messages into a single system. It also includes the standard features found in most text-based e-mail applications.

### Design of VistaMail

We designed the VistaMail e-mail system so that current users of text-based e-mail systems could accept it quickly and easily. Many of the design decisions were made so that users wouldn’t have to change their knowledge base greatly to use the system. By integrating capture and viewing services directly into the mailing system, the system architecture and protocol let users easily incorporate audio and video in their messages. We designed the graphical user interface to...
expand the electronic message concept to contain audio and video in addition to text. Next we describe the design decisions for the system architecture, services, and graphical user interface.

System architecture and protocols

The VistaMail system architecture consists of user agents, message stores, and the message transfer system (MTS). We divide the user agents on the sender and receiver sides into functional components (see Figure 1). It’s helpful to view these objects in a layered fashion because each component depends on the component above or below it for information. On the sender side, these components consist of the

- Capture component
- Integration component
- Sending component

On the receiver side, the components consist of the

- Display component
- Assembly component
- Retrieval component

Figure 1 depicts the interaction of the sender and receiver user agents with the MTS and message stores for delivering and receiving VistaMail messages. Below we discuss the responsibilities of each functional component in detail.

The sender side. The Capture component receives direct input from the user. It primarily captures the media components that make up the e-mail message in real time. For example, if a message contains audio and video, the Capture component must communicate directly with the microphone and digital camera, synchronizing the recording of the audio and video streams. Once the Capture component has captured all the message components, it passes these components down to the Integration component. The Integration component integrates the separate message components and adds important header information to create a single VistaMail message object. This message object is finally passed down to the Sending component, which establishes communication with the standard MTS.

The receiver side. After the MTS deposits the VistaMail message object into the message store on the receiver side, the Retrieval component extracts the object from the message store. The Retrieval component then disassembles the message object back into the header information, text component, and multimedia components. The system then places multimedia components into an extended message store (EMS), separate from the standard message store of the MTS, which still contains the text components (see Figure 2). The header information, which includes references to the message’s specific audio and video components in the EMS, is then passed to the Assembly component. Using these header references to obtain the multimedia components in the EMS, the Assembly component—when directed by the user—assembles the separate message components and passes them to the Display component, which then plays the message in a synchronized session. Figure 3 illustrates the complete process.

Services

A completely integrated multimedia messaging system must provide a minimum set of composition, playback, and management services to compete with current easy-to-use text-based e-mail.
systems. In addition to providing all the basic services of text-based e-mail systems, VistaMail also offers multimedia capabilities. While many of the related message systems discussed in the previous section might offer a subset of VistaMail’s services, none of them offer all the necessary services. Below, we define the required minimum set of services and discuss how these services are delegated to the user agent’s functional components.

**Composition services.** The composition services, shown in Table 1, are implemented by the components found in the sender side user agent (see Figure 1). In addition to providing basic text services such as forward and reply capabilities, text importing, signature inclusion, and file attachments, the Capture component also can capture and preview video and audio streams. The process of creating the VistaMail message object in the Integration component must be flexible enough to allow any combination of text, audio, and video.

**Playback and viewing services.** The components in the receiver side user agent (see Figure 1) provide the playback and viewing services, shown in Table 2. Actual reception of messages occurs in the Retrieval component. Once the Retrieval component places the video and audio message parts into the EMS, the Assembly component gathers these parts and passes them to the Display component simultaneously, causing the text and video to automatically appear on the screen. The integration of playback services into the Display component distinguishes the VistaMail system from related works. We describe our adaptive synchronization protocol for video and audio playback in the section “Playback and synchronization.”

**Mail management and additional services.** The mail management and additional services in Table 3 make up the core services, which aren’t mapped to specific components in the user agent. These services must be provided by any e-mail system, whether or not it integrates with multimedia services.

**Graphical user interface.**

The graphical user interface (GUI) design of a new multimedia e-mail system must build on traditional user interfaces for current users to accept it. Most e-mail systems consist of two main windows—one for browsing and the other for composing messages. Below, we describe the functionality of a minimum set of graphical components essential to these windows.
The main browsing window. The main browsing window divides into four functional regions, which let the user manage, select, and view messages. Below, we briefly describe each area.

1. Command area. The command area, located at the top of the browsing window, consists of various buttons that let users navigate around the system. The functionality of such buttons typically include:
   - Compose a message
   - View next message
   - View previous message
   - Delete and undelete the current message
   - Reply to and forward the current message

2. Selection area. The selection area displays the list of message headers. Generally, each header in the list includes the sender, date, and subject of the message. Users can change the message headers displayed in the selection area by moving to different mail folders.

3. Text area. The text area contains the selected message's text component. Once a user selects a message header, the text automatically appears in the text area.

4. Video area. The video area contains the video screen, VCR controls, and volume control for the selected message's video and audio components. Once a user selects a message, the video component automatically appears on the video screen. This area and the media capture area in the composition window are notable additions to an e-mail system's GUI.

The composition window. The composition window consists of a text and media capture area. These areas let users compose integrated messages.

1. Text area. A multimedia e-mail system must still retain the ability to send text. Text often contains information that cannot be properly conveyed through video and audio. In addition, some occasions call for text-only messages. The text area of a multimedia e-mail system is the same as that found in text-based e-mail systems.

2. Media capture area. The media capture area lets the user record video streams, audio streams, or both. This area contains a video screen and VCR controls to record messages. The user has the flexibility of composing audio only or video and audio messages (with or without text). Additionally, the user can preview and re-record these media components.

Implementation of VistaMail

We developed the VistaMail system on Sun Ultra machines running the Solaris 2.5.1 operating system. VistaMail was installed on a cluster of these workstations in the Computer Science Department at the University of Illinois connected by a 10-Mbps Ethernet. The message stores and EMS were located on disks mounted via a network file system (NFS). Video messages were captured with Sun Video MPEG hardware cards using the Sun Video software application programming interface (API). We captured and played back the audio using the internal audio device in μ-law format. A modified version of Berkeley's mpeg2play MPEG software decoder handled video playback. With this implementation, we aimed to explore the integration complexity and provide compatibility with current e-mail systems.

Compatibility and standard protocols

We envision the VistaMail system as an alternative to popular e-mail systems like Netscape, Eudora, and Pine. For compatibility with other e-mail systems, we implemented VistaMail using several standard protocols commonly used by these systems. This compatibility lets other e-mail systems read text e-mail sent by VistaMail and vice versa. Below we briefly describe some of the common protocols used in the implementation.

SMTP. SMTP transports messages within the MTS. It uses simple text commands to communicate with a local message transfer agent (MTA), which in turn contacts the remote MTA and transfers the mail message. The mail message consists of the message header followed by the message body. The message header includes standard fields such as "From," "To," and "Subject." These headers let the receiving application identify the message to the user. The message body includes the text portion and any MIME attachments.
MIME. MIME is a specification format developed for including non-text components within standard Internet mail sent through SMTP. When a MIME attachment is appended to a message, the message header must contain a field identifying the top-level media type and subtype. A media type identifies the data in the attachment. The subtype gives a more exact description so that the data can be presented properly. When users add more than one MIME attachment to a message, the top-level media type indicates the existence of multiple parts. Each of these parts is preceded by the appropriate media type/subtype identifying the attachment.

**Base64 encoding.** Standard SMTP only allows for the transport of ASCII characters. Since multimedia components contain binary data, they must be encoded into ASCII characters before they can be transported. A generally used encoding technique is Base64. Every three binary bytes (24 bits) divides into four 6-bit parts. Each 6-bit value serves as an index for the Base64 alphabet, a table that holds ASCII characters for index values 0 to 63. The reverse procedure decodes Base64 to binary.

**Implementation of system architecture.**

In this section, we describe interesting implementation issues concerning the VistaMail system architecture.

**The sender side.** We previously described creating a VistaMail message object with the Integration component. This message object consists of a modified SMTP header, the text, video, audio, and synchronization components as MIME attachments. (Note that other attachments not related to the VistaMail message object, but included as part of the user’s message, will also be attached as MIME attachments.) These components attach to a top-level media type or subtype of “multipart/mixed” and specify a unique message boundary, which can be used by any mail application to find the attached components as defined by the MIME standard. Each of the multimedia components attaches to an “application” media type. Three subtypes identify each specific component. Figure 4 depicts these subtypes, labeled “x-vmail-audio,” “x-vmail-video,” and “x-vmail-sync” for the audio, video, and synchronization components, respectively. (As we’ll discuss in the section “Receiving messages,” audio and video are recorded as separate streams. The synchronization component contains information necessary for the proper synchronized playback of the audio and video streams.) This implementation combines the responsibilities of the Integration and Sending components into a single procedure by sending the VistaMail message parts to the MTA in the order specified by the VistaMail message object. The MTA in our implementation is the local sendmail daemon, which uses a reliable transmission-control and Internet protocol (TCP/IP) to send the message to the receiver.

Several alternative methods can be used for the MIME content’s format. For example, the top-level media type might be “multipart/parallel.” This would signify that the components should be rendered together. However, the message can contain two distinct parts. The first part, called the unified message, includes the text, audio, and video that should be rendered together. The second part contains any additional attachments. Note that a “multipart/parallel” content-type would not provide this distinction in parts.

Another alternative would be to attach the body parts as “multipart/mixed” and specify “video/mpeg” and “audio/basic” for the content type of the video and audio components, respectively. However, this approach would only work if the order of the parts adhered to a given rule. If no specific ordering convention was established, you could not differentiate between the unified audio/video/text parts and extra attachments. An ordering convention would make the purpose of the attachments unambiguous and allow the content type to more accurately describe the component.

**The receiver side.** Recall that the Assembly component passes the VistaMail message object’s head-
er information to the Display component, which extracts the audio and video components from the EMS when directed by the user. This header information consists of the standard fields found in the SMTP header and several new fields specific to the VistaMail system. Specifically, when a mail message contains any audio or video component, the system adds a new field labeled "X-Vmail-ID" followed by a unique identifier to the VistaMail header (see Figure 4). The Display component uses this field to extract the audio or video components from EMS for synchronized playback. To identify the compression schemes of the video and audio components, the system also adds the fields labeled "X-Vmail-Audio-Type" and "X-Vmail-Video-Type" to the header. For example, μ-law-encoded audio files would have "au" as their X-Vmail-Audio-Type and MPEG-encoded video files would have "mpeg" as their X-Vmail-Video-Type.

Media server. We explored two different implementations of the MTS. The first implementation (Figure 5a) sends the video and audio components using a new protocol. While the text component still transmits through SMTP, the system sends the video and audio components to a special media server located in the same network as the recipient’s mail server. Rather than having the video and audio components extracted from the standard message store by the Assembly component, this server deposits these multimedia components directly into the EMS. In addition to resolving the capacity problems of the message stores, this means VistaMail no longer needs to perform the potentially costly task of decoding the multimedia components (see the “Receiving messages” section for further discussion). The server implementation simply opens up a connection and transfers the separate files. A more sophisticated server may make specific accommodations for the media types, such as laying out the audio and video sequentially on disk. While this approach improves performance, the use of a nonstandard protocol can hinder the acceptance of the system.

The second implementation (Figure 5b) sends the VistaMail multimedia components as MIME attachments via SMTP as a text e-mail message. This implementation lets VistaMail read and send messages to other e-mail systems. The drawback to this implementation, however, is that the accumulated VistaMail messages on the receiver side may exceed the capacity or specified limits of the message stores. Because the video and audio components of a VistaMail message may be quite large (we discuss this issue further in the “Observations and experiences” section), several VistaMail messages may rapidly fill or exceed a disk quota for the message stores before the Assembly component can separate these multimedia components. In a multiuser setting, for example, this setup may be problematic for a system administrator in charge of allocating disk space for users. However, we adopted this approach based on the support of standard transmission protocols.
The first approach can actually lead to a hybrid solution, which would require changes in the MTA to support transmitting the multimedia components separately. However, this topic requires further research.

**Implementation of services**

Many of the services described are self-explanatory. Below we discuss issues pertaining to those services that require additional explanation.

**Receiving messages.** Because VistaMail uses a software video decoder, any machine—with or without cameras and video cards—can play back messages. This makes VistaMail flexible for site adoption. Only a few sending machines might possess digital cameras, and the rest can be used for receiving messages.

Another issue of receiving VistaMail messages involves extracting multimedia components from the message stores. (Note that this discussion only pertains to the implementation of VistaMail without the server.) The first time a user reads or selects a VistaMail message, the multimedia components are parsed out of the message stores, Base64-decoded, and placed into the EMS by the Assembly component. Subsequent reads of the message therefore don’t require the time for decoding. When a message that contains video is selected, the first frame of the video automatically displays on the video screen. Extracting and decoding messages only once lets the initial frame of the video appear immediately, therefore providing a better response for the user.

**Flexible composition.** VistaMail can be used regardless if a machine has digital cameras and microphones. If the machine has no multimedia capabilities, VistaMail simply becomes another text-based e-mail system. When used with video and audio devices, however, VistaMail lets users send e-mail messages using any combination of text, video, and audio. Users can compose the separate components in any order desired.

**Playback and synchronization.** The available resources limit VistaMail’s real-time capture capabilities. Specifically, the Sun Video cards can only compress video into MPEG-1 format in real time, without audio. Additionally, the MPEG software decoder does not decode audio. If the resources are available, a decoder that supports MPEG system streams can be used. However, separate streams allow the retrieval and playback of only the audio component (which is generally much smaller in size) from the EMS when system performance is limited. For these reasons, audio and video are recorded as separate components.

Because VistaMail uses separate video and audio components, it needs information to synchronize the two separate streams. The system generates a synchronization file on the sender side, as video is recorded. Relative time stamps, indicating when a frame should be displayed, are associated with every video frame and stored in the synchronization file. (Audio is recorded at a constant bit rate and therefore does not require any explicit synchronization information.) As Figure 4 shows, the system sends this synchronization file as an extra component of the message and stores it in the EMS. Therefore, complete VistaMail messages contain at least four components: text, audio, video, and synchronization information.

Since VistaMail videos are intended to capture the sender’s face, lip synchronization proves essential. VistaMail uses a simple adaptive synchronization scheme to achieve this lip synchronization. The system synchronizes the video and audio streams against the real-time clock time axis, not each other. While the audio device will play back the audio stream in real time, the playback of video must be controlled according to the information in the synchronization file. Specifically, if a video frame cannot be displayed between the times indicated in the synchronization file for that specific frame and the next frame, the system drops it. Note that VistaMail uses MPEG files, which consist of I (intracoded), P (predictive), and B (bidirectional) frames. Only P and B frames are dropped because I frames are essential for decoding subsequent frames properly. However, dropping some P and B frames can also cause distortion in the video, since certain macroblocks will be incorrectly decoded without the data provided by a previously dropped P or B frame. Figure 6 illustrates the adaptive protocol. Higher user load, LAN traffic, and processor speed may affect the jitter and synchronization skew of the VistaMail video messages.

**Implementation of the graphical user interface**

The graphical user interface should give users the feel of creating one unified message with audio, video, and text. To provide such an integrated perception, we placed a persistent video area in the VistaMail system’s browsing and composition windows. This area contrasts with a “pop-up” style video window approach used in other mailing systems, which appears whenever the user...
composes or views a video message. We believe that a "pop-up" style window detracts from the unified nature of the message. Next we discuss further issues pertaining to the two specific windows.

**Main browsing window.** The main browsing window is the implementation of the Display component in the VistaMail design (see Figure 7). The selection area displays some of the header information provided to the Display component by the Assembly component. When a user selects a message, the text automatically appears in the text area, and the first frame of the video (if there is video for that particular message) automatically appears in the video area. The user employs the VCR buttons in the video area to maneuver through the video message. This automatic video display gives the user the sensation of a single integrated message.

**Composition window.** The composition window also provides a persistent video area (see Figure 8). The user engages the digital camera and begins recording to compose a video message. Pressing the record button without engaging the digital camera creates an audio-only message. The preview button lets the user view or listen to a message before sending it to the receiver.

**Observations and experiences**

VistaMail was made publicly available to the student labs of the Computer Science Department at the University of Illinois. This multiuser environment provided valuable feedback on the acceptance and usability of a multimedia e-mail system in an academic environment.

**Technical issues**

Here we’ll discuss some of the technical issues we observed during the deployment of VistaView.

**LAN environment.** Disk quotas are an important concern when using VistaMail in multiuser LAN environments. In the student clusters at the University of Illinois, for example, each student can have approximately 5 Mbytes of disk space. Because VistaMail sends and receives 320 by 240 MPEG files recorded at about 25 frames per second, these disk quotas may be easily exceeded. If, for example, users received only 10 VistaMail messages daily, each containing a 10-second video, the 8 Mbytes (such a video averages 800 Kbytes) of disk space needed to store the messages would exceed current student quotas. To deal with this issue, we placed the EMS outside the users’ home directories on a large, separately mounted disk partition. This partition provided ample disk space for many VistaMail messages. While decreasing the picture size and frame rate would reduce these space constraints, this space issue is only temporary. With the exponential growth of disk sizes, disk quotas in multiuser environments will become less of an issue.

As we described in the section “Playback and synchronization,” VistaMail implements a simple adaptation protocol for the playback of video and audio. To achieve immediate video playback,
VistaMail doesn’t use any buffering techniques when reading video data over NFS. Because NFS provides enough bandwidth for the video sizes used, buffering was not necessary when bringing the multimedia components from the EMS to the local machine. The jitter and synchronization skews measured in the acceptable ranges when the machines were lightly loaded. (Steinmetz and Nahrstedt showed that people perceive video as “out-of-sync” when synchronization skews exceed 160 milliseconds.) However, we experienced increased performance degradation as system load on the local machine increased. This behavior can be improved when using a reservation-based and QoS-aware CPU server.

Home environment. In single-user PC home environments, gigabyte hard drives are already becoming commonplace, and the disk space required is no longer a concern. However, the waiting time to download large multimedia files becomes an issue for home users who ordinarily download and view e-mail over relatively slow modem connections. As network technology progresses, these time constraints will become less of a concern. In the meantime, better download performance can be achieved by

- Reducing frame size, quality, and rate. For instance, a 2-minute 160 by 120 MPEG file recorded at 6 frames per second averages only 400 Kbytes.

- Incorporating streaming protocols for delivering media streams between the EMS residing at the service provider site and the user agent at the user site. Start latency would reduce to the amount of time needed to initiate the streaming process. However, streaming introduces additional problems such as scalability and user acceptance of special streaming service requirements.

- Employing negotiation protocols to determine user bandwidth and adapting video characteristics based on the available bandwidth. The user could specify parameters to filter messages or provide audio only for large messages.

Acceptance issues

Integrated multimedia systems such as VistaMail have several advantages, for example, the ability to move away from the English-based communication system. Integrated audio lets individuals communicate in their native languages. Users who aren’t comfortable with typing or can’t type (for example, because of carpal-tunnel syndrome) also benefit from using audio and video.

The current implementation didn’t include explicit security features. Video in e-mail may provide some measure of security, since it would be more difficult to falsify a video message. Explicit security measures would be a possible extension to the mail system. We don’t believe that the lack of security features deterred users from accepting the system.

However, in a multiuser environment such as a student lab where workstations are close to each other, little privacy exists among students. The fear of being overheard prevents many students from listening to or composing audio-video messages. Therefore, VistaMail mostly benefits an office or home where the environment implicitly enforces privacy.

Even outside of a multiuser setting, however, our experiences indicate that the transition from text e-mail to multimedia e-mail may not be easy. E-mail users have become accustomed to using text as a communication medium. Additionally, individuals tend to be camera shy and feel awkward talking to inanimate objects. Thus many hesitate to use video to communicate. We feel that acceptance of a multimedia e-mail system will depend on consumer behavior and changing perceptions of e-mail.

Conclusions

According to our studies and observations, the acceptance of a multimedia e-mail system will depend on

- Perceptual quality of received multimedia e-mail messages (control of resources to achieve bounded jitter and skews)

- Network technology progress to increase bandwidth to the home environment

- Disk technology progress to allow storage of large multimedia messages

- Privacy when composing and receiving audio-video messages

- Easy use of the integrated system
The increase in the number of systems becoming available to facilitate composition and playback of audio and video messages illustrates the increased interest in multimedia e-mail. With the continued progress of technology, we believe that this trend will continue, and video e-mail systems will become commonplace in the future.

Acknowledgments

This work was supported by the National Science Foundation’s Computer and Information Science and Engineering (CISE) Infrastructure grant (NSF CDA96-24396) and the Career grant (NSF CCR96-23867). We also thank the reviewers of this article for providing helpful suggestions.

References


Christopher Hess is a PhD candidate in the Computer Science Department at the University of Illinois at Urbana-Champaign. He received BS and MS degrees in mechanical engineering from Tufts University, Medford, Massachusetts and an MS degree in computer science from the University of Illinois. He has recently been involved in streamed media and video conferencing research. His interests include object-oriented design, multimedia applications, operating systems, and networking.

Daniel Lin is a law student at the University of Chicago. He received a BA from Amherst College, Amherst, Massachusetts and an MS from the University of Illinois at Urbana-Champaign in computer science. Most recently, he served as the policy fellow for the US Public Policy Office for the Association for Computing Machinery. His interests include multimedia systems, computer ethics, and issues concerning the relationship between technology and law.

Klara Nahrstedt is an assistant professor at the University of Illinois at Urbana-Champaign in the Computer Science Department. She received her BA in mathematics from Humboldt University, Berlin, in 1984, and MS in numerical analysis from the same university in 1985. In 1995 she received her PhD from the University of Pennsylvania in the Department of Computer and Information Science. Her research interests focus on multimedia services, protocols, end-point architectures for multimedia, QoS, and resource management in real-time multimedia distributed systems. She co-authored Multimedia: Computing, Communications and Applications (Prentice Hall, 1995) and received the Early NSF Career Grant. She is a member of the ACM and IEEE Computer Society.

Readers may contact the authors at the Computer Science Dept., University of Illinois at Urbana-Champaign, 1304 W. Springfield, Urbana, IL 61801, e-mail {ckhess, klara}@cs.uiuc.edu or djlin@acm.org.