MHEGAM—A Multimedia Messaging System

Brigitte Kervella and Valérie Gay
Université de Picardie Jules Verne

MHEGAM (MHEG-1 Advanced Mail) is a complete multimedia messaging system for the creation, exchange, and restitution of multimedia messages that express spatial and temporal synchronization among their components. MHEGAM can be based on the standard messaging systems X.420 or MIME. We present the multimedia extensions MHEGAM-X.420 and MHEGAM-MIME and discuss the multimedia message format and architecture components for both systems.

The explosion of the Internet and increased use of electronic mail have generated wide commercial interest in multimedia messaging systems. Multimedia messaging can facilitate the communication of multimedia information among users. In addition, it can be used with multimedia applications such as distance learning and multimedia conferencing.

Most systems proposing multimedia message interchange do not allow the expression of spatial and temporal synchronization among message components. Existing messaging systems' multimedia messages still contain a limited set of media types and present body parts sequentially to the recipient. To improve multimedia messaging systems, therefore, we must find ways to support creation of formatted messages that include synchronization features among the body parts.

This article presents such a solution: a complete multimedia messaging system called MHEGAM (MHEG-1 Advanced Mail). This system supports the creation, exchange, and playout of messages containing many different data types including audio, video, text, graphics, and pictures. Users may also express multimedia synchronization between the message components. In addition, MHEGAM can handle the processing of new data types.

The development of multimedia mail requires the support of standards that allow the exchange of multimedia messages in heterogeneous environments. This pertains to both message transfer and multimedia information format. To ensure openness, MHEGAM takes an original approach by using the MHEG-1 standard to specify the message's elaborated multimedia content and a standard messaging system for message exchange. MHEGAM is based either on X.420 or on MIME (Multipurpose Internet Mail Extensions), two standard messaging systems. We analyze and compare these and present their extensions, which we call MHEGAM-X.420 and MHEGAM-MIME. MHEGAM does not require modification of the message transfer system (MTS), which guarantees interoperability and portability.

Comparing X.400 and MIME

The X.420 and MIME architectures use the same message handling system but have evolved in two different ways. The X.420 messaging system, mainly used in the commercial domain, facilitates interpersonal message transfer.[1] (Other X.400 system types include the Voice Messaging System (VMGS) for voice messages and the Interapplication Messaging System (IAMS) for direct information exchange between applications.) MIME, a multimedia extension of the Internet messaging system,[2,3] specifies mechanisms to include and code a wide range of data types without changing anything in the message transfer. It is now widely used in commercial and academic environments.

Messaging systems are not always compatible. For a long time, X.400 was the only standard messaging system able to federate heterogeneous messaging systems. The Internet phenomenon changed this. However, X.400 offers more services than the Internet messaging system. For example, in X.400, a probe gives information concerning a message's readability by the recipient before being sent; also, messages can be forwarded selectively and automatically to other recipients. Moreover, this series provides such services as electronic data interchange and directory service.

The X.400 messaging system encounters compatibility problems among its standard versions (for example, X.400-84, X.400-88, and X.400-92). MIME may also have various versions, but it has defined a conformity concept to avoid compatibility problems.

The Internet messaging system goes beyond X.420 in defining and handling multimedia body part types. A MIME message may include text, audio, video, pictures, and other data types, whereas X.420 permits only the exchange of text

1070-986X/97/$10.00 © 1997 IEEE

and other formats such as teletex, videotex, and encrypted data. Table 1 shows the limits of X.400 in handling multimedia information compared to MIME.

MIME can also reference a file or application data, and it can group information of different types. The recipient can choose a preferred format. This mechanism also permits splitting a long message to ensure its transfer; the message is recombined before being presented to the recipient.

Nevertheless, an X.400 message may contain voice data, computer application data, and other messages. X.400 is a complex messaging system: it is reliable and complete, and it offers various services useful to federate messaging systems. It is expensive, though, a major disadvantage for users. MIME is more widely accepted and better suited for multimedia but offers fewer services overall.

Table 1 reveals that neither messaging system can create formatted multimedia messages that express space and time relationships between different body parts. MIME cannot really execute presentation in parallel even if it provides a multipart/parallel body part type. This is the main feature missing from the multimedia messaging systems that puts X.420 and MIME at the same level. The next section provides a solution to this problem for both standard messaging systems.

Table 1. Comparison of X.420 and MIME body part types.

Body Part	X.420 Type	MIME Type
Text	Ia5-text	Text
Fax	G3-facsimile and g4-class1	—
Audio	VMGS System (X.440)	Audio
Picture	—	Image
Video	—	Video
Message	Message	Message
Multiple data	—	Multipart
Application data	IAMS System	Application
Referenced data	—	Message/external-body
Formatted message	—	—

MHEGAM

In developing MHEGAM, we sought to offer users the ability to create and exchange multimedia formatted messages. A formatted message may contain all kinds of body part types, such as text, audio, and video, that can be synchronized in time and space. A complete multimedia messaging system must accommodate new data types and be usable worldwide, and thus must be based on standards. In MHEGAM, multimedia body parts are based on MHEG-1 to express spatial and temporal synchronization between different media types. This new body part type does not change the message transfer system; MHEGAM, in order to be standards-based, is based on X.420 or MIME.

MHEG-1 overview

The ISO SC29/WG12 working group carries out the MHEG standardization (for a list of sources, see the sidebar "MHEG Details"). In developing MHEGAM, we used MHEG-1 to specify elaborated multimedia parts. This standard represents and encodes multimedia and hypermedia information, and concentrates on object interchange more than on the production or execution process. This has led, sometimes, to misunderstandings of MHEG-1's concepts and mechanisms.

Fundamentally, the MHEG-1 standard specification seeks to provide a generic and extensible framework for multimedia and hypermedia object encoding without application- or platform-specific assumptions for multimedia object production, handling, and execution. Because MHEG-1-based interchange is independent from the production environment, the resulting objects can be downloaded and used on different runtime systems.

MHEG-1 is object-oriented; that is, it defines a set of classes to encode multimedia application behavior for presentation that focuses on layout and logical relationships between multimedia objects. An MHEG-1 object can be composed of parts based on other standards such as JPEG for

MHEG Details

MHEG technology is further detailed in the following ISO documents:

ISO-13522(1): "MHEG: Information Technology—Coding of Multimedia and Hypermedia Information, Part 1: MHEG Object Representation, Base Notation (ASN.1)," IS version, 1996, http://www.demon.co.uk/tcasey/wg12.html.

ISO-13522(3): "MHEG: Information Technology—Coding of Multimedia and Hypermedia Information, Part 3: MHEG Script Interchange Representation," DIS version, 1996.

ISO-13522(4): "MHEG: Information Technology—Coding of Multimedia and Hypermedia Information, Part 4: MHEG Registration Procedure," IS version, 1996.

ISO-13522(5): "MHEG: Information Technology—Coding of Multimedia and Hypermedia Information, Part 5: Support for Base-Level Interactive Applications," DIS version, 1996.

ISO-13522(6): "MHEG: Information Technology—Coding of Multimedia and Hypermedia Information, Part 6: Support for Enhanced Interactive Applications," DIS version, 1996.

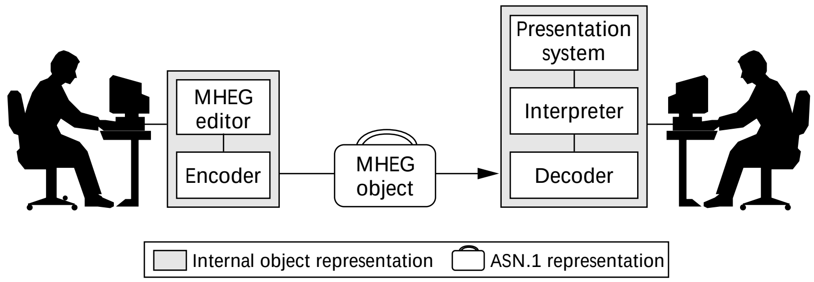

Figure 1. MHEG-1 runtime system.

pictures and MPEG for video. To describe the time and space synchronization between object parts, MHEG-1 introduces mechanisms that fulfill the needs of most multimedia and hypermedia presentations and documents. These include generic mechanisms to define interactive behavior and presentation synchronization between separate object parts.

For MHEG object presentation, a minimal MHEG runtime system provides an entity for decoding ASN.1 data structures and an entity called an MHEG engine that parses and interprets MHEG objects. The MHEG engine also communicates with local presentation facilities, responding to application- or user-initiated events in a specified way (by pushbutton, for example). MHEG objects are created by an MHEG editor and coded in ASN.1 by an encoder before being transmitted. Upon their reception, the decoder translates the objects into an internal data representation, and the interpreter and presentation system then present them to the user.

Figure 1 shows this configuration.

Acronyms

ASN.1	Abstract Syntax Notation Number One
CCETT	Centre Commun d'Etudes de Télédiffusion et Télécommunications
CD	Committee draft
Davic	Digital Audio Visual Council
DBMS	Database management system
DIS	Draft International Standard
DOR	Distinguished Object Reference
DS	Directory service
EDIMS	Electronic Data Interchange Messaging System
EMS	Extended Message Store
EUA	Extended User Agent
FTAM	File Transfer Access And Management
HyTime	Hypermedia/Time-Based Structuring Language
IAMS	Inter-Application Messaging System
IANA	Internet Assigned Numbers Authority
IS	International Standard
ISO	International Standardization Organization
ISO SC29 WG12	Subcommittee and working group studying MHEG standards
JPEG	Joint Photographic Expert Group
MHEG	Multimedia and Hypermedia Expert Group
MHEGAM	MHEG-1 Advanced Mail
MIME	Multipurpose Internet Mail Extensions
MPEG	Moving Picture Expert Group
MS	Message Store
MTS	Message transfer system
RDT	Referenced Data Transfer
SGML	Standardized General Markup Language
SMTP	Simple Mail Transport Protocol
SQL	Structured Query Language
UA	User Agent
UI	User interface
VMGS	Voice Messaging System

MHEGAM characteristics

Some MHEG-1 tool kits have been developed as part of CCETT projects to support MHEG-1 standard development. The one we use here, MHEGAM, is based on the DIS version and has been developed by CCETT in cooperation with Euclid. It contains an MHEG-1 editor, an interpreter, and a program allowing MHEG object presentation. It runs under Windows 3.1. The MHEG-1 editor has a graphical interface that offers a layout view corresponding to the spatial composition, a timeline view that represents object behavior in time, and a library view that displays every object and data being edited. The user need not be familiar with MHEG concepts to use the MHEG-1 editor. Additional information about MHEG-1 appears elsewhere.[4]

MHEGAM employs new techniques for creating and exchanging multimedia messages. It also uses MHEG-based delivery mechanisms.

Creation and exchange of multimedia messages. MHEGAM lets users create a message that includes body parts of the types accepted by the messaging system on which it is based. It also permits an elaborated multimedia message encompassing spatial and temporal synchronization using the MHEG-1 standard. The user creates an MHEG object using an MHEG editor and integrates it in an extended body part, an MHEG body part. A multimedia message integrating an MHEG body part may also contain classical body parts. Figure 2 illustrates an example.

In this example, Brigitte sends a multimedia message to answer Elena's message. Brigitte's message contains a message body part, one MHEG body part, and a text body part. The message body part corresponds to the message Elena sent. The text body part is a short answer and the MHEG body part is a multimedia document, a report synchronized with English subtitles. The other body parts are created with the mail tools on which MHEGAM is based. An alternative body part can replace any body part that cannot be presented on the recipient's system.

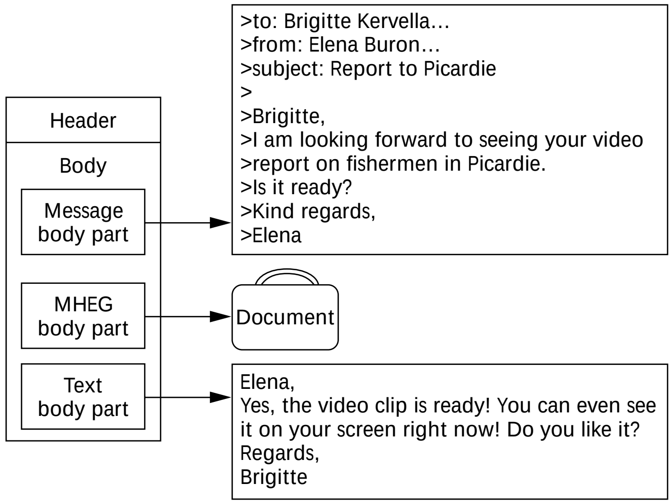

Figure 2. An example of a MHEGAM message.

We create the MHEG object using the MHEG editor and, if necessary, specific editors such as a video editor. The characteristics of the MHEG object are translated into composite object attributes and action or link objects; objects are then encoded in ASN.1. The MHEG object can then be integrated in the message as an MHEG body part, as described in the MHEGAM-X.420 and MHEGAM-MIME sections below. Figure 3 illustrates the generation of an MHEG object.

While in theory neither messaging system standard limits message size, in practice the capabilities of message transfer agents limit the message size to around 2 Mbytes—even if some agents have higher capabilities. If a message exceeds this size, a reference is created using the distinguished object reference (DOR) standard and is automatically sent instead of the data itself. DOR defines some references on objects;[5] in this case the referenced data is stored for the recipient to retrieve later. Once the object or its reference is integrated with the message, transfer can occur as for any other message. To avoid a double coding (with possible incompatibility) during the transfer, it is necessary to prohibit implicit conversions.

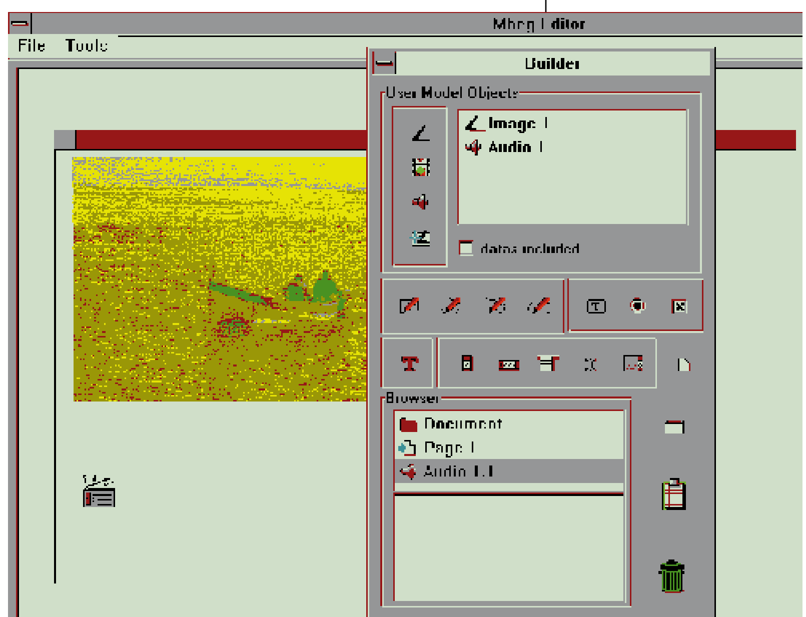

Figure 3. Creating an MHEG object using the MHEG Editor.

Multimedia message delivery. At delivery time, the recipient system displays our multimedia message like any other message except for the MHEG body part. Its presentation occurs in a new window using an MHEG-1 tool kit. Before being presented, the MHEG body part must be decoded by the ASN.1 decoder, processed by the interpreter, and played out. If the received message contains a reference, the messaging system asks the recipient whether she wants to retrieve the referenced object. If so, another mechanism automatically retrieves the object.

When a problem prevents a body part's presentation, two solutions are possible. Either the system presents an alternative body part (if included in the message), or it presents an alert window to warn the sender that the message has only been partially received.

MHEGAM choices

Table 2 (on the next page) summarizes the differences between X.420 and MIME concerning multimedia data. MIME supports all (or nearly all) data types, but both messaging systems lack formatted multimedia message capabilities. Therefore, an MHEG object containing formatted multimedia data is introduced. The user creates this MHEG object with the MHEG editor; it is coded in ASN.1 before being integrated into the MHEG body part.

X.420 does not support picture, sound, video, and multiple body parts, for which we must use an MHEG body part. On the other hand, voice and application data may be replaced by a standard: a voice body part defined in VMGS and an IAM body part defined in IAMS.[6,7] X.420 processes the reaction to a new media type in a preventative way using a probe; MIME does this in a curative way using an alternative body part.

X.420 lets the user send a probe to determine whether the message can be sent to the recipient. A probe, which is a message without its content, indicates message size and data type contained. The recipient messaging system automatically notifies the sender whether the message can be processed in its environment. If so, the message may be sent without risk of rejection. MIME permits sending an alternative body part in case the main body part cannot be received. We chose to add this mechanism to MHEGAM-X.420, defined as an alternative body part.

In standard messaging systems, the message cannot exceed 2 Mbytes; this significantly constrains multimedia message exchange and thus multimedia mail system development. However, for messages containing video, for instance, MHEGAM may send a reference instead of the video, which lets the recipient retrieve it using another means. MIME proposes this, but X.420 does not; for the latter we thus defined a new body part type—a DOR body part. It follows the DOR standard, which defines object references. Some tools to create and handle these references are also defined using the DOR standard.

MHEGAM-X.420

The X.420 standard allows the transfer of very few data types (text and message). Some extended body parts—the file transfer and voice body parts—have been added to the X.420 standard in an externally defined body part using a macro. This macro permits defining three new body part types to integrate MHEG, object reference, and alternative body parts. For MHEG body parts, it is better not to allow the conversion of the message during its transfer; a header field can provide that.

Our messaging system contains different body parts. The X.420 standard provides ia5-text and message body parts. The X.400 standard also provides two extended body parts, Voice and IAM, that are standardized in VMGS and IAMS. MHEGAM adds three other extended body parts: MHEG and DOR body parts for multimedia and referenced data, and the alternative body part for parts the recipient cannot receive (as discussed above). In contrast to MIME, this alternative body part contains a single component.

The DOR body part contains a reference indicating data type, referenced data's physical location, communication service used, and quality-of-service parameters. The sender can replace a body part with its reference; the body part or file is then stored on a local server and transferred to the recipient on demand using a data transfer service such as Referenced Data Transfer (RDT) or File Transfer Access and Management (FTAM). Using the DOR body part limits the message size to 2 Mbytes (practical limit of the message transfer agent's capabilities). In this case, the biggest body parts are referenced until the message size is acceptable; this mechanism may be automatic.

Table 2. MHEGAM body part types.

Body Part	X.420 Support	MIME Support
Text	Already supported + fax	
Audio	VMGS (Voice body part)	
Picture, video, multiple data	New feature: MHEG body part	Already supported
Application data	IAMS (IAM body part)	
Referenced data	New feature: DOR body part	
Formatted message	New feature: MHEG body part	
New type of media	Probe	Not supported
	New feature: Alternative body part	Already supported: Multipart/alternative

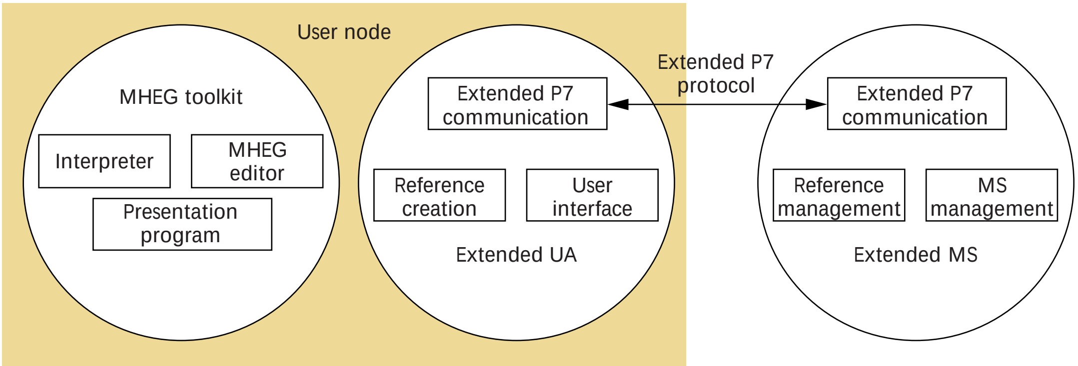

Figure 4. Architecture of the complete MHEGAM system.

MHEG-1 provides another alternative: the ability to reference data using the SGML standard. The reference, instead of the data itself, is defined and included in the object. So for the MHEG body parts, huge data may be referenced directly into MHEG objects as soon as the user creates them.

Architecture

The MHEGAM-X.420 system architecture is based on the client-server extended messaging system service model described elsewhere.[8] It takes into account modifications made in 1992, 1994, and 1996. This model corresponds to an extension of the Message Store (MS), the User Agent (UA), and the P7 protocol. It contains an Extended MS (EMS) and Extended UA (EUA). The EMS acts both as a server for an EUA and as a client in relation to the MTS. EMS functions are provided to its EUA through the standard X.420 MS ports.[9] The only functional differences exist in the operations transported through these ports.

Figure 4 illustrates the complete system architecture. It shows the three components of this architecture—the EUA, the EMS, and the Extended P7 protocol—and their distribution.

Extended User Agent. The EUA is located on the user's system. It contains the user interface, the extended P7 communication module, and the reference creation module. The interface realizes a multimedia messaging system user interface and can automatically download the part of the interface processing it requires from the server as soon as it becomes active. The extended P7 communication module permits communication between the EUA and the EMS.

The reference creation module collects information needed to create the reference and transfers it to the reference management located on the EMS. Then it waits for the answer to its reference creation request. If accepted, the module puts the received reference instead of the object into the message. Upon receipt of a DOR body part, this module provides the user with the information required to retrieve the body part content.

The user node also contains the MHEG-1 tool kit for multimedia message creation. These tools include the MHEG editor, the presentation/runtime program, and the interpreter briefly presented earlier in the section "MHEG-1 overview."

Extended Message Store. The EMS implements the facilities offered to the user. It contains the standard MS facilities plus the MHEGAM-X.420 facilities concerning referenced objects. For the multimedia messaging system, the specific service is registered to create an object reference and store this object in the EMS (due to its memory capabilities).

The EMS consists of three main elements: the extended P7 communication module, reference management, and message store management. The extended P7 communication module allows communication between the EUA and the EMS. The reference management checks whether a reference can be created when it receives a reference creation request from the EUA. When there is enough space and the operation is allowed, reference management stores the object, creates the reference using the DOR standard, and transmits it to the EUA. To avoid storage problems, a storage authorized delay is defined for each object; the objects can be destroyed by reference management once the delay expires. Message store management is a standardized MS.

Extended P7 protocol. We defined an extended P7 protocol that facilitates exchange between the user and the multimedia messaging system. This extends the P7 capabilities in part by using

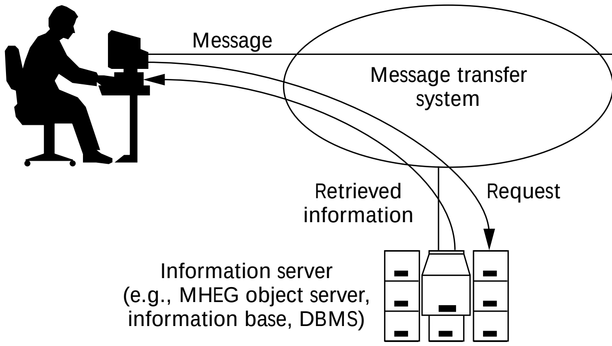

Figure 5. Multimedia messaging system: extended services.

the bind and unbind operations between the UA and the MS. At binding time, the EUA requires negotiation of services. The service provided can be with or without the MHEGAM extension.

Additional modifications concern the message's object reference part. The reference requires three operations:

1. The *register* operation allows reference management to create a reference using the DOR standard and store the referenced object in the EMS. This reference is returned to the EMS user to be included in the message.

2. The *navigate* operation returns the list of referenced objects contained in a directory.

3. The *fetch* operation lets users retrieve objects on the EMS using their name and location.

We chose this design to guarantee maximum interoperability among systems using the X.400 specifications.

MHEGAM-MIME

The MHEGAM messaging system based on MIME requires few modifications to provide a complete multimedia messaging system.

As mentioned, most media types can be exchanged via MIME, which allows the exchange of messages or application data and permits referencing to a file. It also permits sending a message containing different body parts and defining alternative body parts. Several alternative body parts can be provided for presentation in case of a problem. However, this messaging system cannot exchange formatted multimedia messages that define spatial and temporal relationships between components. As with MHEGAM-X.420, we use an MHEG object to define such a message.

Integrating an MHEG object requires extending MIME to the MHEG data type. We do this in MIME using subtypes. The application type is used for data that does not correspond to the other predefined types and especially for data that must be processed by an application. An MHEG object, for example, must be decoded and interpreted before being presented to the recipient. To do so, a new subtype called MHEG may be linked to the application type; it must be registered to the IANA organization before use. We could also define a private subtype called X-MHEG, but we desire a multimedia messaging system that is as open as possible. As with MHEGAM-X.420, it is better not to allow the conversion of the message during its transfer; a header field can do that.

Architecture

MIME architecture is similar to the Internet messaging system architecture; it does not change the Internet message transfer system.[10] The SMTP protocol ensures this transfer by providing a store and forward service for messages.

Conclusion

MHEGAM lets users create, exchange, and receive formatted multimedia messages that may contain spatial and temporal synchronization of their components. Because this multimedia messaging system is based on the X.420 and MIME standards, it can be used worldwide and across many different platforms. MHEGAM can also be integrated in other multimedia applications such as distance learning and multimedia conferencing. For example, it can be used in a multimedia conferencing service to send the conference minutes to participants or to retrieve slides from an information server.

Using MHEGAM: An example

Figure 5 shows an example of extended service via our multimedia messaging system. The user on the left prepares an MHEGAM message using information found in a remote database by first sending a request to the information server, which is in charge of satisfying the request and returning the result. The message mode is well adapted to request large volumes of information, especially if the user wants to avoid high connection costs. This could be done by sending an SQL query encapsulated in an MHEGAM message to a specified address corresponding to, for example, a database management system (DBMS) or an MHEG object server having a messaging system interface. When the message is ready, it is sent to the user on the right. This example illustrates how a multimedia messaging system can be used in a multimedia distributed environment with other applications.

A postscript

Two years after the end of this study, MIME has become more popular than X.420. MHEG-1, which became a standard, has lost some of its economic interest compared to SGML and HyTime. In fact, the Internet's rapid development favors HyTime and Web-related tools, formats, and mechanisms. Originally, MHEG-2 proposed an alternative representation of MHEG-1 written in SGML. If the work had been completed and had not been canceled from the WG12 project list, MHEG-2 might have spurred the integration and use of MHEG on the Web.

In the MHEG standards family, MHEG-5 is now more popular than MHEG-1, probably thanks to its use in Davic. However, the solution we propose here is still up-to-date and valid. Indeed, MHEG-1 suffices for a multimedia messaging system, the MHEG-1 tool kit developed in parallel with the standard providing a useful foundation. We used the tool kit based on the DIS version of the standard in this study (no fundamental difference exists between the DIS and IS versions). The detailed specification of MHEGAM exists elsewhere,[11] where the abstract service and operations are described in ASN.1. MM

References

1. ISO 10021(7) and ITU-TS X.420, "Message Handling Systems—Information Processing Systems—Text Communication—MOTIS—Interpersonal Messaging System," Oct. 1991, http://www.iso.ch/catf/cat.html.
2. N.S. Borenstein, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies," RFC 1521, Network Working Group, Sept. 1993.
3. D. Crocker, "Standard for the Format of ARPA Internet Text Messages," RFC 822, Network Information Center, Aug. 1982.
4. T. Meyer-Boudnik and W. Effelsberg, "MHEG Explained," *IEEE MultiMedia*, Vol. 2, No. 1, Spring 1995, pp. 26-38.
5. ISO-10031(2), "Information Technology—Text and Office Systems—Distributed-Office-Applications Model, Part 2: Distinguished-Object-Reference and Associated Procedures," Dec. 1990.
6. ISO-10021(16), "Message Handling Systems—Information Technology—Text Communication—MOTIS—Inter-Application Messaging: Service Definition," Feb. 1994.
7. ITU-TS X.440, "Message Handling Systems: Voice Messaging System," Oct. 1991.
8. L. Duchien, V. Gay, and E. Horlait, "Integration of a Multimedia Filestore with X.400," *Proc. ICCC 92*, R. Parodi Eds., Genova, Italy, Oct. 1992, pp. 365-370.
9. ISO-10021(5) and ITU-TS-X.413, "Message Handling Systems—Information Technology—Text Communication—Motis—Message Store: Abstract-Service Definition, Amendment 1: Message Store Extensions and Message Store Logs (Part 1, 5, 6, 7)," Dec. 1993.
10. J.B. Postel, "Simple Mail Transfer Protocol," Network Information Center, RFC 821, Aug. 1982.
11. B. Kervella, *MHEGAM: A Complete Multimedia Messaging System Based on X.400/MIME and MHEG*, PhD thesis, MASI Laboratory, Paris 6 University, Paris, June 1995.

Valérie Gay is an associate professor at Université de Picardie Jules Verne in Amiens, France. She received her PhD degree in computer systems from the Université de Paris 6 in May 1991. Currently, she supervises a team on open distributed object-oriented platforms in Laboratoire de Recherche en Informatique d'Amiens (LaRIA). Her research activities include quality-of-service, multimedia, and distributed application design.

Brigitte Kervella joined the LaRIA laboratory in September 1996. She is an associate professor. She received her PhD degree in computer systems from the Université de Paris 6 in June 1995. She worked on multimedia messaging systems and participated in the French AFNOR standardization group on the X.400 messaging system. Currently, her research activities include object-oriented distributed platforms and multimedia information.

Contact Kervella at LaRIA, Université de Picardie Jules Verne, 5, rue du Moulin Neuf, 80000 Amiens, France, e-mail Brigitte.kervella@laria.u-picardie.fr.