MMMGate - Enabling Overall Multimedia Messaging

Manfred A. Bogen <Manfred.Bogen@gmd.de>
Arnold Krechel <Arnold.Krechel@gmd.de>

Keywords

Multimedia, Electronic Mail, MIME, X.400, BERKOM, Global Store, External References, Teleservices

Abstract

Electronic mail is one of the most popular value-added services in networking today. It is the basis for many multimedia applications, it belongs to the enabling technology for computer-supported co-operative work (CSCW) and it is a built-in feature of many desktop applications like text editors, word processors or presentation programs.

Up to now, messages and e-mail services are mostly text- and ASCII-based. With the availability and high user acceptance of multimedia-based services, the need for the transport of multimedia objects arises even in message-based applications.

In the BERKOM environment, a PTT-driven project to realise the communication infrastructure of Berlin, a superset of the X.400 protocol has been implemented to overcome deficiencies of present X.400 and the handling of large multimedia messages has been defined. GMD has been awarded by the DFN association being responsible for major mail components within the BERKOM environment to implement a multimedia mail gateway in this context.

In this paper the development of this gateway (MMMGate), is described. Major attention is paid to the handling of large multimedia messages, the management of partial messages and the co-operation with a global store. The experiences made and the planned embedding of the gateway into the national multimedia reference centre and into user-driven projects is presented finally.

1 Introduction

In this paper we assume a certain familiarity of the reader with e-mail protocols and formats like X.400, SMTP and MIME. Its structure is as follows: Section 1 provides the background why overall messaging and multimedia are needed today. Section 2 describes the MMMGate development. In section 3 we present our experiences and ideas about the future of MMMGate.

1.1 Why Multimedia?

More and more companies world-wide are organised following decentralisation principles giving more responsibility to individual groups (e.g. cost centres). Working groups and collaborative teams thereby need new communication and co-operation facilities which are more efficient, less expensive and more reliable than the traditional combination of telephone, fax and travelling. The goal is to improve the quality of work and the productivity of the teams and to strengthen the competitiveness of the company. Multimedia is/ are part of this process.

People get 80% of information by sight, 11% by hearing and 3.5% by smell (which is not supported by any multimedia format today). From this information they retain 20% of the information they see and 30% of the information they hear [1]. This means that people heavily rely on images in their information acquisition and that exchanging multimedia information is efficient. The price to pay are increased storage and bandwidth requirements (table 1).

Table 1: Document Lengths

Content	KiloBytes
text	30
text and graphics	200
stored video clips	> 1.000
interactive video clips	> 1.000/min

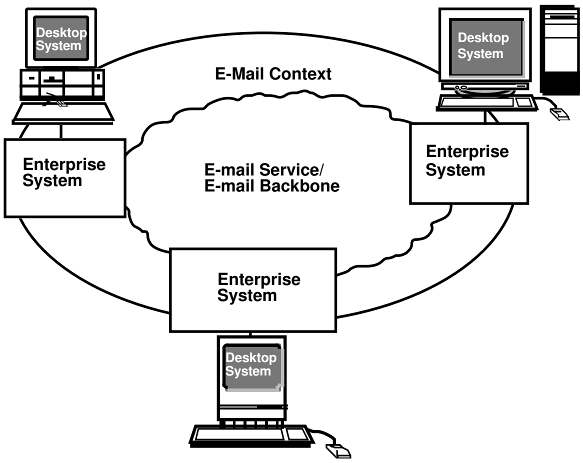
Multimedia can be classified as interactive and real-time on one side and message-oriented and asynchronous on the other side. Speaking about e-mail in the following we restrict ourselves to the latter category as message-oriented multimedia obviously is based on message transfers. These messages are called compound, hyper or multimedia documents as they not only contain the information itself but also structural information about the document. Multimedia documents can contain

- word-processed text,
- revisable graphics,
- images,
- fax,
- spreadsheets,
- CAD drawings,
- digitised voice,
- audio clips and
- video clips.

1.2 Overall Messaging

In the previous chapter we saw that the exchange of multimedia information and documents is a must today. This is especially true for electronic mail (e-mail) applications as e-mail is the most widely used Internet service today and is part of many CSCW applications being responsible for the transport and distribution of information. About 45 million users are estimated to use it. Its high acceptance is somewhere between telephone and fax machines. It is the basis for communication and co-operation world-wide. Electronic mail systems are available for notebooks, mainframes and everything in between.

Figure 1: Overall Messaging Model



We can classify e-mail systems into three different categories depending on their operation (figure 1):

- desktop systems,
- enterprise systems,
- backbone systems.

In the local area networks people use desktop and enterprise e-mail systems, in the wide area networks enterprise and backbone e-mail systems.

Most of the desktop e-mail systems like QuickMail, Microsoft Mail, Lotus cc:Mail, or NOVELL Global MHS use proprietary, non-standard e-mail protocols and formats internally. Thereby it is possible to offer a functionality and complexity for end users which is perfectly adopted to the proprietary architecture and system.

For the path through wide area networks they use directly, through gateways or through their enterprise e-mail systems (figure 1) SMTP and X.400 for the message transport. The same applies to CSCW software like Lotus Notes or Microsoft Exchange when they use embedded e-mail capabilities. For enterprise and backbone systems SMTP and X.400 is a must. By using SMTP and X.400 as open standards in the enterprise and backbone systems the e-mail backbone is kept simple.

In spite of the simplicity of the backbone, these systems must support multimedia body parts because otherwise the enhanced functionality of the desktop systems and information as such would be lost by passing the e-mail backbone.

The effective handling of large messages (see table 1) may not be important within an organisation in a local area network and individual solutions are acceptable here. But it becomes very important for world-wide enterprises in wide area network and for people using the e-mail backbone. Due to the simplicity principle open solutions have to be implemented here too.

[RFC1521] enhances the old format of ARPA Internet text messages [RFC822] without changing the related SMTP [RFC821], i.e. without implications on the message transfer

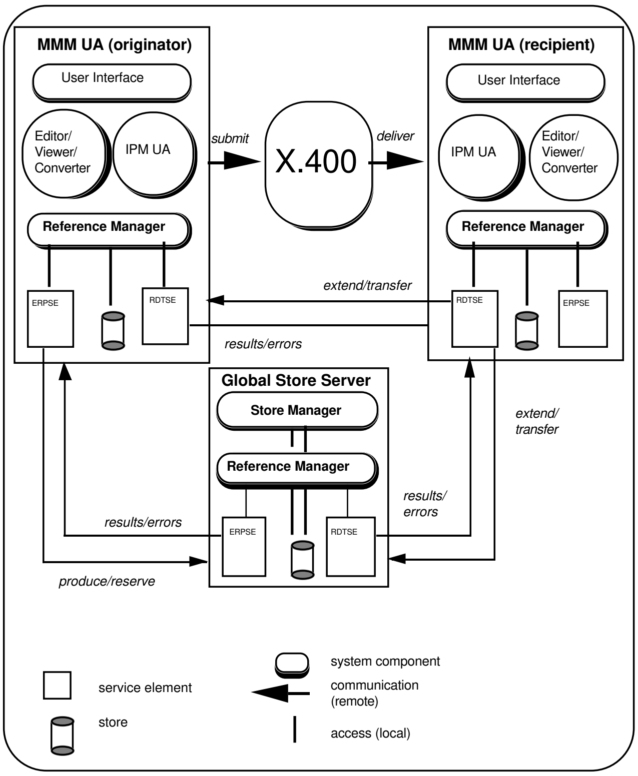
- to carry multimedia information effectively,
- "to permit the fragmented transmission of bodies that are thought to be too large to be passed through mail transport facilities,"
- to specify "large bodies by reference to an external data source."

These capabilities to handle large messages with multimedia body parts effectively are important for the quality of e-mail systems [2]. The question which arises immediately: does X.400 as second open standard in a simple e-mail backbone support similar capabilities? The answer is: [X.400-84,X.400-88] does not. The multimedia body types are supported in principle, however the precise definitions are still missing ("for further study").

1.3 The BERKOM Environment

These problems and deficiencies have been investigated and solved by the BERKOM people [3]. They invented X.400(BERKOM), a kind of multimedia-capable X.400-88 protocol, taking almost the same approach as the MIME people, i.e. without changing the underlying message transfer protocol (X.400-P1) but using the X.400-88 basic body part *externally-defined* for their extensions.

Figure 2: The BERKOM Multimedia-Mail Teleservice



The BERKOM Interpersonal Messaging Service (IPMS) is part of the BERKOM Multimedia Mail Teleservices (figure 2) and it supports (*externally-defined*) body parts for

- audio (G.711)
- audio-video (MPEG)
- compound documents (PostScript)
- external reference [DOR]
- image (IIF)
- link (Clear Text)
- binary (for anything not defined above)

Other *Basic Body Part Types* used in BERKOM are *ia5text* and *message*.

A mechanism (reference manager) was implemented to handle external references in message bodies via distinguished object references (DORs) [DOR] and to access a globally available store service, the global store server.[1] A special ISO file transfer protocol, Referenced Data Transfer [RDT], is used for all components of the external reference mechanism.

As long as the BERKOM people only communicated with partners in the BERKOM and native OSI world, everything worked fine. Finally it was decided to enable interoperability with the large Internet community.

2 The MMMGate Development

The DFN association, *Verein zur Förderung eines Deutschen Forschungsnetzes*, was a co-operation partner in the BERKOM project from the early years. They signed responsible for mail aspects and especially for a multimedia mail gateway (MMMGate) called "X.400(88-BERKOM)-X.400(88-MIME) gateway" in the beginning. MMMGate should offer the following functionality:

- mail gateway between X.400-88 and SMTP/MIME
- mail gateway between X.400(MIME) and X.400(BERKOM)
- conversion of multimedia messages with different body parts
- file transfer gateway between RDT and FTP
- handling of externally-defined body parts (BERKOM)
- handling and mapping of external references
- co-operation with the global store server

The plan was to base the development on a gateway specification written by a national expert group [4]. IC Release 2.0 with ISODE and PP should be the implementation platform for the mail part as PP already offered an X.400-88 to MIME gateway. The gateway should run under UNIX and the development should take 6 months.

[1] An external reference is similar to MIME a globally unique identification of the stored data (server, file, directory, access type ...).

The DFN association started a call for tender in December 1993. A GMD consortium, consisting of two GMD groups (the *FOKUS* institute in Berlin and the *Research Department for Network Engineering (NW)* in Sankt Augustin[2]) and the *Fraunhofer Institute for Computer Graphics* in Darmstadt, was awarded in March 1994. The project started in July 1994 and was finally completed in May 1995 after the approval of the results by the DFN association and the DFN multimedia teleservices reference centre at the Technical University of Dresden. The Fraunhofer Institute was responsible for the development of graphic converters, GMD's FOKUS institute in Berlin for the file transfer component and the NW group for the mail component and project management.

2.1 The MMMGate Design

At first, a new version of the gateway specification had to be written incorporating comments to the early specification, experiences in answering the call for tender and changes (new definitions) which had happened in the meantime in the BERKOM context. Apart from that, the specification looked quite different for us now that we had to implement it and we did it in parallel to the first tests and implementation steps.

2.1.1 The MMMGate Architecture

MMMGate was designed to consist of three components (figure 3):

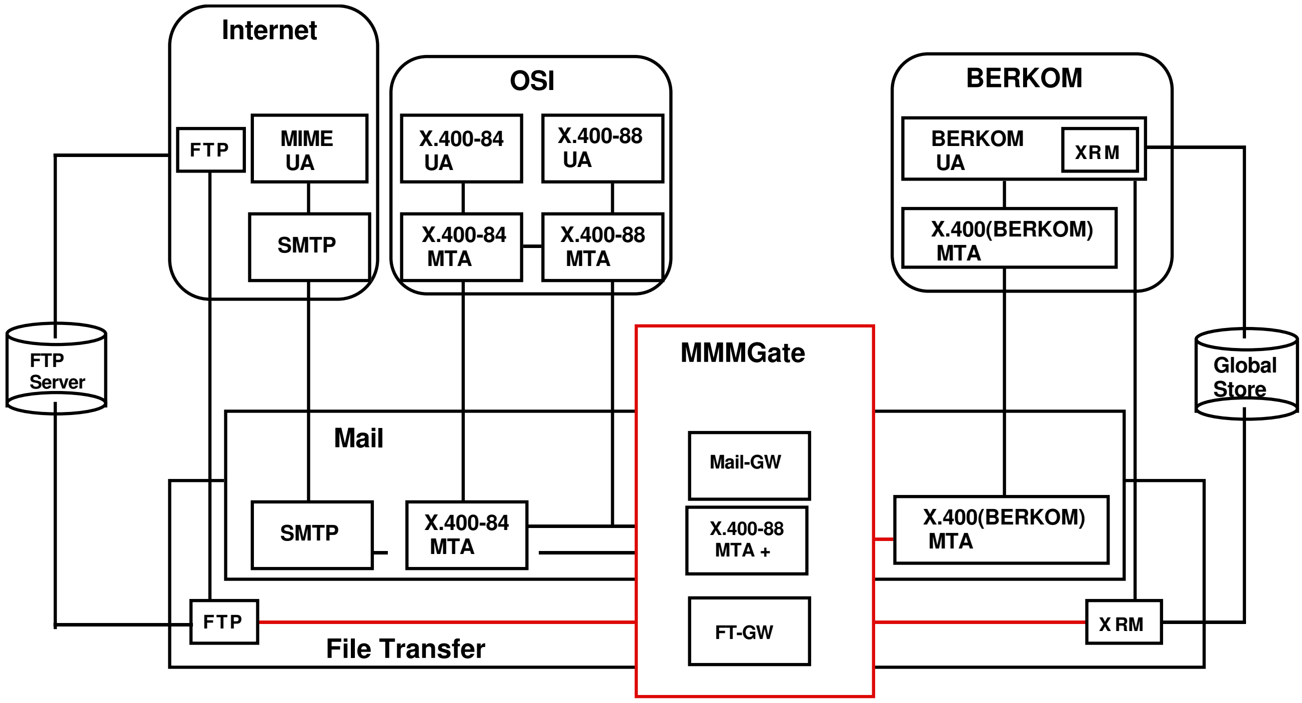
- an enhanced PP (X.400-88 MTA+) being responsible for the handling of X.400-84, X.400-88 and SMTP messages.
- a mail component (Mail-GW) being responsible for the conversion of individual body parts, the handling of the generic message structure and the processing of partial messages following [RFC1521].
- a file transfer component (FT-GW) offering gateway functionality between FTP and RDT and access to the global store.

Following the specification MMMGate should logically be located in between X.400 and X.400(BERKOM). We decided to respect this in our design strictly. The gateway from X.400 to X.400(BERKOM) should be called if and only if a message has to be sent to a BERKOM MTA (via X.400) and has arrived at the current MTA from a non-BERKOM SMTP or X.400 MTA. The other way round an X.400 message from BERKOM should pass the X.400(BERKOM) to X.400 gateway if and only if it has been accepted from a BERKOM MTA and has to be sent to a non-BERKOM MTA via SMTP or X.400. Both parts of the gateway

[2] We had been and still are responsible for the establishment and operation of DFN multiprotocol value-added services.

should operate on the complete P22 content of the messages.

Figure 3: MMMGate Architecture and Embedding



2.1.2 About Messages and Body Parts

Though most of the gateway's conversions could have been done on the level of body parts, following the PP philosophy, we detected that for internal reasons PP was not at all inclined to manage all conversations desired and we found two situations in which the gateway really needs the contents of the heading part of the P22 message:

- The MIME to X.400 gateway part gets confused if it detects *rfc987 heading extensions* according to [RFC1327] which do not correspond to the contents of the mail. Therefore these heading extensions have to be removed as soon as the operation of the gateway leads to deviations between heading extensions and the contents of the message.
- The partial message assembler (to be described in the next section) needs the whole of the partial message (heading and body) for the assembly of a message.

We finally wanted the gateway to operate primarily, i.e. by default on body part level eventually thus following more the philosophy of PP.

2.1.3 MIME Partial Messages

The treatment of MIME partial messages has not been included in the MIME to X.400 gateway of PP IC Release 2.0. According to the implementors of this gateway there are no plans to incorporate the handling of partial message in a future release. They consider partial messages to be a mistake and argue like this:

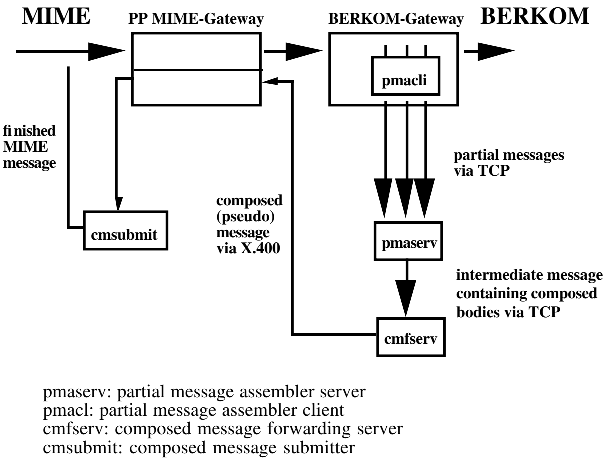
- How long should a gateway store the different parts of a message if not all of them have arrived? A day? A week? A month?
- Partial messages to a certain destination may arrive via different routes to the final recipient. It's only the final destination UA which may be sure to see all the different components of a partial message. Therefore it should be the task of the UA to assemble partial messages.
- UAs very often run on small systems and should not bear the burden of complex time and storage consuming operations like the reassembly of partial messages.

Though we saw their points as well we had to deal with the problem of partial messages at our gateway as the BERKOM community is purely X.400 located and will never have any knowledge on MIME partial messages.

The overall design of the partial message assembler is a client/server architecture (figure 4).

A MIME partial message content part will be mapped to a Generic MIME defined Extended Body Part (GMEBP) in X.400 by PP's MIME gateway according to [RFC1494,RFC1495]. If the BERKOM gateway detects such a body part in the P22 content of the message (using the content-type and content-parameters of the parameter part of the GMEBP (see paragraph 5.2 [RFC1494])) several steps will be performed.

Figure 4: Handling of Partial Messages



1. The set of P1 recipients for which the MTA is responsible is determined by *pmacli*. This information together with the message (P22 content) is sent via TCP to the partial message assembler *pmaserv*.

2. If the partial message has been delivered to *pmaserv*, it must be removed from PP. This is done by setting the responsibility bits for all P1 recipients to false implicitly.

3. Steps 1. to 2. are repeated until all the different components that sum up to the complete message have been received by *pmaserv* which is now able to assemble the message to an intermediate result.

The result of this assembly is an X.400 message with the following drawbacks:

a) The body of the message is a X.400-88 GMEBP consisting of the whole of the partial message as if having been gatewayed as number 1 of a partial message consisting of a single component (number=1,total=1 according to [RFC1521]).

b) The header mixing as described in [RFC1521] has not taken place and the content of the partial message has not been affected. The message will be sent/resubmitted to the set of P1 recipients as in 1.

The information on these P1 recipients is passed to the steps 4 and 5 which have to follow now.

4. The preliminary assembled message of step 3 is sent via TCP to *cmfserv* which is responsible for the further processing. The message is passed to PP in such a way that it will traverse the PP gateway from X.400 back to MIME.

5. The resulting MIME message looks now like a MIME partial message consisting of one single component. The header mixing will now take place so that the reassembled MIME message can be produced. *cmsubmit* resubmits the MIME message to the list of P1 recipients determined in step 3.

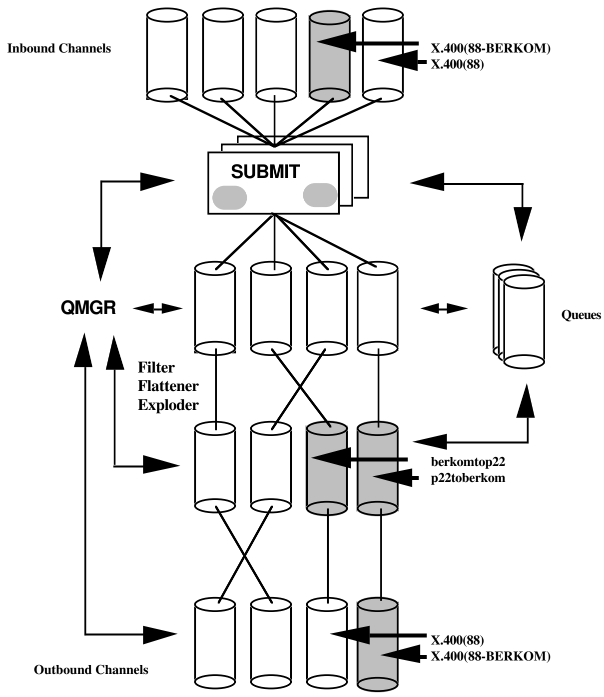
The advantages justifying the additional overhead to simpler solutions with respect to implementation effort and system resources are as follows:

- Only if partial messages arrive at the X.400 side and are destined for the BERKOM community the message will be reassembled.
- Incoming partial messages may arrive via X.400 or SMTP and thus may have taken different routes. Nevertheless they can be reassembled.
- If several instances of MMMGate are in operation *pmaserv* could be located on a dedicated system thus eliminating the different route problem completely.
- Simply as a matter of configuration of PP the implementation of the partial message assembler allows the assembly of partial messages for any other community.

2.2 The MMMGate Implementation

We had to modify PP's *SUBMIT* process slightly in order to make sure the behaviour of PP as indicated in figure 5.

Figure 5: Modification of PP for MMMGate



For an incoming X.400 message PP computes it's internal routing on a per recipient basis according to the encoded information types contained in the P1 envelope of the message and it's configuration. For each recipient the internal routing consists of a list of PP channels. These channels process the message in turn under control of PP's queue manager *QMGR*. To the list of channels computed by the *SUBMIT* process we had to add our gateway channel (from

X.400 to X.400(BERKOM) or vice versa depending on the concrete situation).

This could be achieved by a single subroutine call which had to be inserted in the PP sources. Along with this we assigned the queue id of the message to an additional global variable which is used by the FTP/RDT part of the gateway for the co-operation and communication of non-correlated processes operating on the same message.

Our modifications of PP could remain very small and should not lead to real problems when adapting the gateway to subsequent releases of PP.

2.3 The Functionality of MMMGate

In the following we highlight only some aspects of the MMMGate functionality.

2.3.1 Body Parts

Tables 2 to 4 show the functionality as far as the mapping of the different body parts is concerned.

Table 2: MIME to X.400(BERKOM)

MIME Body Parts [RFC1521]	**X.400(88) Body Parts [RFC1494/1495]**	**X.400(BERKOM) Body Parts [5]**
text/plain; charset=iso-8859-x	*EBP-General Text*	message with two body parts: *IA5Text* and *EBP-generaltext* (in link bp description *alternative*)
multipart/mixed	*message; Heading Extension mixed*	summary/description in link bp
multipart/digest	*message; Heading Extension digest*	summary/description in link bp
multipart/alternative	*message; Heading Extension alternative*	summary/description in link bp
multipart/parallel	*message; Heading Extension parallel*	summary/description in link bp
message/partial	*GMEBP; Heading Extension partial-message*	the message will be composed and mapped.

Table 3: X.400(BERKOM) to MIME

X.400(BERKOM) Body Parts [3]	**X.400(88-MIME) Body Parts [5]**	**MIME Body Parts [RFC1494/1495]**
BEBP-link	*ia5Text*	*text/plain*

Table 4: Reversible Mappings

MIME Body Parts [RFC1521]	**X.400(88-MIME) Body Parts [RFC1494/1495]**	**X.400(BERKOM) Body Parts [5]**
text/plain;charset=us-ascii	*ia5Text*	*ia5Text*
message/rfc822	*message BP*	*message* BP
message/external-body	*GMEBP*	*BEBP-External Reference*(DOR)
application/octet-stream	*bilaterally-defined*	*BEBP-Binary*
application/postscript	*MEBP-postscript*	*BEBP-document (postscript)*
image/gif	*MEBP-gif*	*BEBP-Image* (IIF - Colour look up table, uncompressed)
image/jpeg	*MEBP-jpeg*	*BEBP-Image* (IIF-greyscale or JPEG-coloured)
audio/basic	*GMEBP*	*BEBP-audio (G.711)*
video/mpeg	*GMEBP*	*BEBP-audio-video (mpeg)*
image/g3fax	*G3-facsimile*	*BEBP-Image*
application/x400-bp(g4-class1)	*g4-class1*	*BEBP-Image*

2.3.2 External References

On the message originator's request large messages or body parts are stored in a file server or the BERKOM global store and only an external reference pointing to the file store or the global store is passed through to the recipients of a message. It is their choice afterwards whether they want to fetch it or not. Figures 6 and 7 show the operation of the gateway in both directions, from MIME to BERKOM where a MIME extended reference (MEX) is translated to a BERKOM distinguished object reference (DOR) and from BERKOM to MIME.

Figure 6: External References at the Mail-GW (MIME2BERKOM)

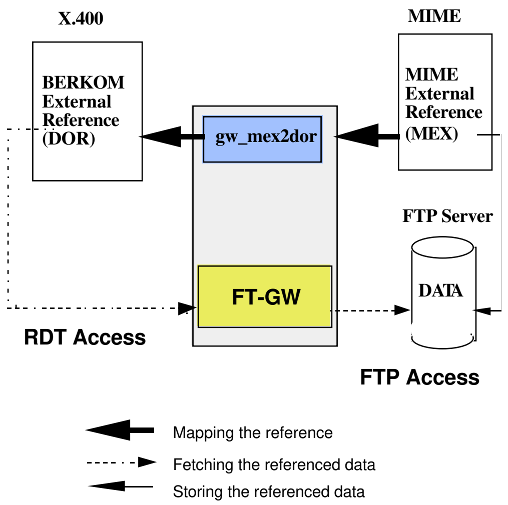


Figure 7: External References at the Mail-GW (BERKOM2MIME)

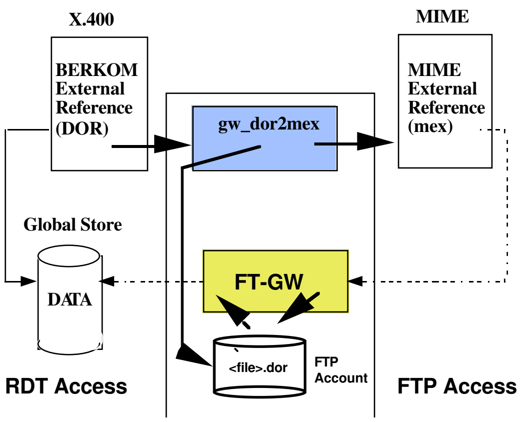


Figure 8 shows the operation of the FT-GW after a MIME external reference (MEX) was translated to a DOR: the recipient of the message in the BERKOM environment decided to fetch the referenced data. The external reference manager (XRM) opens as RDT-client a connection to the FT-GW. The XRM at the gateway takes all the data needed in the DOR and opens an FTP connection as FTP client to the target FTP server. From there the data are sent back to the gateway. They have to pass the gateway's mail component (Mail-GW) in order to assure the same mapping functionality as for messages. Then they are sent as BERKOM data to the BERKOM RDT-client having started the whole process.

Figure 8: External References at the FT-GW (X.400)

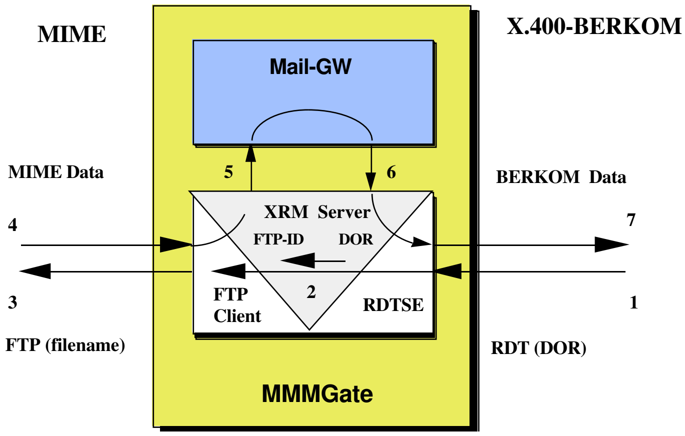
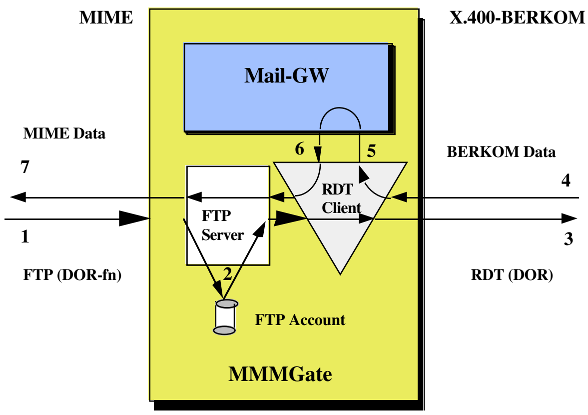


Figure 9 shows the operation of the FT-GW after a BERKOM DOR was translated to a MEX: the procedures are analogous.

Figure 9: External References at the FT-GW (MIME)



2.3.3 Conversion of the Message Structure

MIME allows to structure messages by the content types *multipart/mixed, multipart/alternative, multipart/digest, multipart/parallel*, and *message/rfc*. MMMGate transforms these body parts to a body part of type *message* and puts the information of the message structure into a *Heading Extension.*

The gateway component responsible for the conversion of the message structure uses this *Heading Extension* to check whether it is a body part of type *message/rfc822* or *multipart.* In the latter case or if a message contains a multipart message, a MIME Heading Extension, a BERKOM Link Part and a message summary (figure 10) are produced.

Figure 10: BERKOM Message Summary

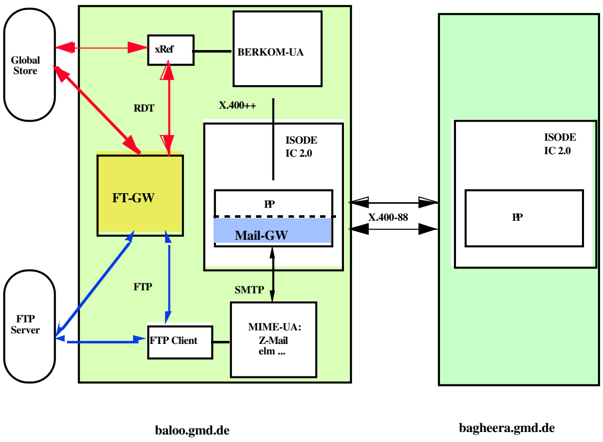
THIS IS A BODY PART GENERATED BY THE BERKOM MIME GATEWAY

Type now	MIME Type	Level	BP No
-	multipart/mixed	1	
ia5text	text/plain	2	3
audio_g711_data	audio/basic modified by GW	2	4
image_iif_data	image/gif modified by GW	2	5

2.4 Testing MMMGate

In order to have a clean test environment without identical components, especially ISODE and PP, influencing each other we built up a configuration with two different machines (figure 11). The BERKOM UA in particular could not be used as SMTP mail user agent.

Figure 11: MMMGate Test Configuration



After the gateway development itself we additionally performed some negative and stress testing with multimedia messages containing multipart/mixed with external references, GIF, JPEG, and PostScript body parts with the following results:

- MMMGate as a server is as robust as the underlying ISODE and PP software,
- the new parts can slow down the performance compared to an unmodified PP system for about 17%,
- we were able to pass 300-400 multimedia messages per hour through MMMGate,
- in handling hundreds of multimedia messages the gateway is able to consume up to 80% of the CPU power of a Sun Sparc Station 10/41 easily,
- we discovered some software bugs in these additional tests but no bottlenecks in our implementation.

3 Results and Experiences

During the project lifetime the following results have been delivered:

- a revised version of the first gateway specification [5],
- the gateway software as such (one compressed file) including a collection of small test tools,
- the combined installation manual and program description [6] which will be published as *GMD Arbeitspapier* soon,
- an assessment of the underlying PP software to be MIME-X.400-88-compliant[3],
- a wish list and alternative solutions for the future improvements of MMMGate,
- the final report, and
- some presentations on national and international level.

Apart from that we are planning an article in one of the next editions of the *DFN Mitteilungen*.

3.1 Lessons learnt

After the MMMGate development we know more, as expected, about multimedia messaging, software development in a group and the team work in a consortium.

3.1.1 Multimedia Messaging

There are still different interpretations of the standardised or well-accepted communication protocols. We found this especially related to X.208, the ASN.1 specification. The BERKOM people and the PP suppliers had different opinions about the encoding so that we had to take all of them into our consideration.

It's difficult to get all the examples needed for the tests as multimedia messages are hard to compose and as there are more standards and RFCs involved than expected.

We found several bugs i.e. deviations from the respective RFCs in the underlying software and reported them so that we expect improvement in the next official version (IC Release 2.1). The public-domain ISODE/PP software used had to be modified (RFC 1495). It is not really suited for a production environment. Configuration of PP is a pain which is confirmed by its supplier.

3.1.2 Software Development in a Group

As different parties were involved in the software development process on a single development machine we tried to introduce a UNIX group concept with individual access rights and individual development environments. The root password for the machine was only known to two project members. This failed especially towards the end of

[3] We doubted this in the beginning.

the project. We very often had the conflict between the people feeling hindered by the group concept and by not knowing the root password and people not wanting to maintain and administer a machine with system privileges for almost everybody. Our attempts to work with a professional software development environment failed due to the complexity of the basis software ISODE and PP.

The better (and more expensive) way seems to be to have stand-alone work stations for each software developer and to have a machine for the system integration and the overall tests maintained by the system administrators.

It is very difficult to base a software development on prototype components still under development. Whenever errors or undefined conditions occur it is not possible to locate the reason on easy terms. To have the sources for each software component involved is a must.

3.1.3 Communication, Co-operation and Responsibilities

Naturally the distribution of tasks has to be very precise with a clear understanding of the deliverables and the time schedule in the consortium. Control mechanisms have to be set up and agreed upon from the beginning.

The installation of the test configuration needed more time than expected. Our approach to install and configure all components by ourselves independent whether they belonged to the gateway development or not was wrong. It is a better way to let the people responsible for the respective components install them so that they can detect the problems directly.

As we participated in the work of the national group writing the first gateway specification [4] we restricted the number of meetings among the consortium members and with our contractor, the DFN association, during the project lifetime to

- one meeting with the contractor in Berlin,
- one meeting as project kick-off meeting in Sankt Augustin, and
- one workshop with the BERKOM people in Berlin.

The rest of the co-operation and communication among the consortium partners was done through the network and via telephone. In Sankt Augustin, we had an informal, internal project meeting once a week.

In spite of the higher travel costs, it is better to have monthly consortium meetings so that problems among the people in the consortium can be solved directly and face-to-face. This is not achievable through the network or with telephone only.

We finally found out that we needed four kinds of distribution lists:

a) local for the daily work,
b) company-internal for the synchronisation within GMD,
c) consortium-internal for the synchronisation within the consortium, and
d) a high-official one for the informal co-operation and communication between the consortium and the contractor.

3.2 Recommended Improvements

In our documentation we provided a rather long list of possible improvements on different levels. In the following only some issues are mentioned.

The MMMGate was developed on top of IC Release 2.0. We found problems in the internal PP routing of MIME messages to several recipients hindering a use of today's MMMGate for production purposes. We recommend to adopt it to IC Release 2.1.

From the experience with the public domain version of PP made in the establishment and operation of message-based value-added services and made in the development of MMMGate we strongly recommend to use the product version of PP as basis for the future work.

The BERKOM specification has been changed in the meantime, version 3.2 is available now. The implications on MMMGate have to be identified and necessary changes have to be implemented.

The handling of large messages can be improved in a way that MMMGate itself decides to put large messages into the global store and to send only an external reference to the recipient. This could be especially useful for composed messages being built out of partial messages. At present, this is only done by the originator of a message.

At last, to ease future developments, installation and maintenance we recommend to embed the MMMGate functionality into IC Release 2.1.

4 Conclusion

The target users of MMMGate are the people working in the DFN regional test beds (DFN-RTBs). At the end of 1994, the DFN association started under this name several application-oriented projects in different parts of Germany. The goal was to enable multimedia teleservices, new forms of co-operation and resource sharing on top of a high-speed networking infrastructure (34 and 155 Mbps) with ATM technology [7]. These people will be supported by the DFN multimedia teleservices reference centre at the Technical University Dresden where MMMGate has been installed for the first time outside of GMD.

At present, MMMGate is not in a state to be embedded into a multiprotocol e-mail backbone. To overcome this we have identified a list of issues to be worked on next. However, through the MMMGate development the possibility to interconnect SMTP/MIME, X.400 and the BERKOM environment via multimedia e-mail and to guarantee the interoperability of the e-mail systems involved with minimal loss of functionality was shown. The biggest part of the work has already been done.

Acknowledgements

The authors would like to thank the colleagues of the DFN office in Berlin, of the Fraunhofer Gesellschaft in Darmstadt, of GMD's FOKUS institute in Berlin and of GMD's Research Department for Network Engineering for their co-operation and support. Parts of this work has been funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF, formerly BMFT).

References

[1] Flückiger, F., "Understanding Multimedia Applications and Technologies," Proceedings of the Interop 93 Engineers Conference, Paris, 1993

[2] Juan Altmayer-Pizzorno, Manfred Bogen, Clemens Wermelskirchen, "E-Mail Quality as a Prerequisite for CSCW Applications," JENC6, to appear in: Proceedings of JENC6, Tel Aviv,

[3] De·Te·Berkom, "The BERKOM Multimedia-Mail Teleservice," Release 3.1, Technisches Zentrum, Berlin, Mai 1994

[4] M. Bogen et al, "X.400(88-BERKOM)-X.400(88-MIME) Gateway Specification, Technical report, DFN-Verein, Berlin, 1993 October

[5] C. Blum, M. Bogen, I. Brüning, V. Kobelt, A. Kraft, A. Krechel, C. Kumpf, H. Pusch, S. Zier, "MMMGate: Das Multimedia-Mail-Gateway," Spezifikation, GMD, Sankt Augustin, Arbeitspapiere der GMD Nr. 883

[6] Manfred Bogen, Thorsten Hennings, Volkmar Kobelt, Inke Kolb, Arnold Krechel, Herwart Pusch, Susanne Zier, "MMMGate: Das Multimedia-Mail-Gateway, Installationshandbuch und Programmbeschreibung," to appear as: GMD-Arbeitspapier, GMD, Sankt Augustin

[7] Kaufmann, P., "Start der Regionalen Testbeds," DFN Mitteilungen, Heft 36, November 1994, pp. 10-12, DFN-Verein, Berlin, 1994

[RFC821] Postel, J., "Simple Mail Transfer Protocol." 1982 August

[RFC822] Crocker, D., "Standard for the format of ARPA Internet text messages." 1982 August 13

[RFC1327] Hardcastle-Kille, S., "Mapping between X.400(1988) / ISO 10021 and RFC 822." 1992 May

[RFC1494] Alvestrand, H.; Thompson, S., "Equivalences between 1988 X.400 and RFC-822 Message Bodies." 1993 August

[RFC1495] Alvestrand, H.; Kille, S.; Miles, R.; Rose, M.; Thompson, S., "Mapping between X.400 and RFC-822 Message Bodies." 1993 August;

[RFC1521] Borenstein, N.; Freed, N., "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies." 1993 September;

[X.400-84] CCITT. Data communication networks: message handling systems. Recommendations X.400-X.430, Malaga-Torremolinos, October 1984

[X.400-88] CCITT. Data communication networks: message handling systems. Recommendations X.400-X.430, Melbourne, November 1988

[DOR] ISO/IEC, "ISO/IEC DIS 10031-2: Information Technology - Text and Office Systems - Distributed-Office-Application model (DOAM) - Part 2: Distinguished Object Reference (DOR) and associated procedures," 1991

[RDT] ISO/IEC, "ISO/IEC IS 10740: Information Technology - Text and Office Systems - Referenced Data Transfer (RDT)," 1992

Author Information

Manfred Bogen has been active in the area of group communication, X.400 development, and X.400 standardisation as expert of the German PTT since 1983. In 1987 he became head of the VaS group being responsible for the provision of value-added services. He studied computer science at the University of Bonn and is co-author of two books about X.400 and distributed group communication. At present, he is the convenor of the TERENA working group on quality management for networking (WG-QMN) and a member of the TERENA Technical Committee and the Internet Society.

Arnold Krechel received his master's degree in mathematics from the University of Bonn, Germany in 1985. He worked in the area of parallelization of numerical algorithms till 1992. Since then he is engaged in the networking area, mainly concerning X.400 and its flavours. He is responsible for the operation, customisation and integration of the X.400 software at GMD, he performed the successful transition from EAN to PP in 1992 and he is an active contributor to the development of the ISODE and PP packages.

The authors can be reached at GMD - German National Research Center for Information Technology, D-53754 Sankt Augustin.