

A Multimedia Messaging Platform for Content Delivering

Simo Gratschew, Janne Raitaniemi, Jorma Ylinen and Pekka Loula

Tampere University of Technology, Pohjoisranta 11 P.O Box 300, FIN-28101 Pori, Finland

simo.gratschew@tut.fi, janne.raitanemi@codebird.com, jorma.ylinen@tut.fi, pekka.loula@tut.fi

Abstract

Short messaging (SMS) is well known and widely used messaging system in mobile phones. Especially the European mobile users are interested in using short messaging communication and services. Short messaging is a text-based system where the user can send messages consisting of approx. 160-characters. The next evolution version of SMS is EMS (Enhanced Messaging Service), which is already being used in some mobile phone models. EMS supports melodies, pictures, sounds, animations and modified text in additions to plain text. Pictures and animations are based on black and white colors. MMS (Multimedia Messaging Service) follows SMS and EMS. The first MMS steps have already been taken in 2002, but the use of MMS will become wider in 2003 when operator interoperability problems will be solved and there are more mobile terminals available for end users on the markets. This study describes the MMS environment in general and the work to make a multimedia content distribution platform. This platform is basically server-based, however there are many things that should be taken into account from the mobile terminal point of view. If the terminal model does not support the multimedia content, the message may not be correctly received. Also, the needs of different kinds of end user groups should be taken into account.

Index Terms

EMS(Enhanced Messaging Service), HTTP(HyperText Transfer Protocol), MM(Multimedia Message), MMS(Multimedia Messaging Service), MMSE(MMS Environment), PDU(Protocol Data Unit), RMI(Remote Method Invocation), SMIL(Synchronized Multimedia Presentation Language), SMS(Short Message System), VAS(Value Added Service), WAP(Wireless Application Protocol), WSP(Wireless Session Protocol), XML(eXtensible Markup Language)

1. Introduction

The development of 3rd generation mobile networks and terminals will enable the provision of new services to mobile users [1]. These new services also include the

Multimedia Messaging Service. MMS is intended to provide a rich set of content to wireless messaging. It supports both sending and receiving of multimedia messages with MMS-enabled devices. Some devices may support only the reading of messages, while more advanced devices also support the sending of MMS content.

In both SMS and MMS, the message originating from a mobile terminal handset is sent and received by the intended recipient. However, there are some differences in the ways SMS and MMS messages are sent and forwarded. In SMS, a message is sent and forwarded directly to the recipient. MMS is a notification system where a recipient receives a notification of an arrived message from the network element and can download the message at that time or later.

2. MMS Architecture

Figure 1 shows that multimedia messaging may move across many different types of networks [2]. The basic connectivity between these networks is provided by the Internet protocol and by an associated set of messaging protocols to provide complete MMS to a user.

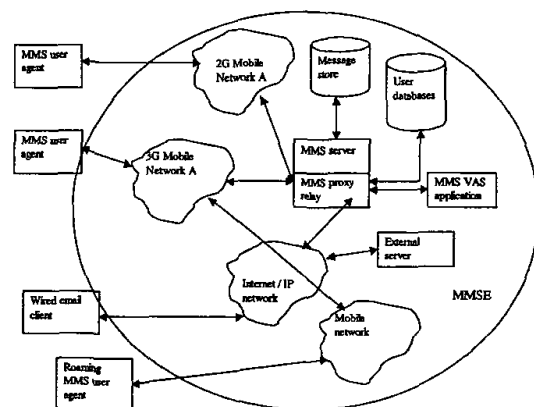


Figure 1. MMS Architectural Elements (MMSE) [2]

MMS user agent

The MMS user agent must provide functions to retrieve a multimedia message (MM) as well as terminal capability negotiation. The MMS user agent may also provide additional functions like

- multimedia message composition
- MM presentation
- MM submission
- presentation of notifications to the user
- signing of MM
- decryption and encryption
- storing of MM on a terminal
- handling of MMS related information
- management and presentation of multimedia mailbox content
- handling of external devices
- user profile management.

The MMS user agent must support a minimum set of media formats which consist of a plain text that contains characters in Unicode and a format for SMS interoperability, AMR for speech, baseline JPEG for still images, ITU-T H.263 for video and file format for dynamic media. MP4 ensures interoperability of video and associated speech/audio. Still image and video support applies to agents that support image and video.

There are also other codecs that are suggested to provide interoperability in the Internet, e.g. mp3, midi and AAC for audio, gif89a for image, MPEG4 visual simple profile level 0 and H.263 profile 3 level 10 for video formats.

MMS relay/server(Proxy-Relay)

The MMS Relay/Server (also called Proxy-Relay [3]) is the network element that interacts with an MMS user agent. It is responsible for sending and receiving MM. The MMS relay server also provides conversion to the MMS format for messages arriving from other systems and conversion of MMS messages leaving to other systems, e.g. Internet email or fax services. The MMS Relay/Server should also provide functions such as

- message content retrieval
- MM notification to user agent
- delivery reporting
- routing forward MMS
- address translation
- temporary storage
- ensuring that messages are not lost before they are successfully delivered

The MMS Relay/Server also includes many other functions that it should or can take care of like generating charging data records or negotiation of terminal capabilities.

It is possible to separate the MMS Server and the MMS Relay. In this case the MMS Relay provides access to the MMS Server, which is responsible for storage services.

MMS user databases

This element may consist of several entities that contain subscriber-related data like user profiles, user grouping, charging information, connection history, and capabilities of the user terminal.

Persistent Network based storage (MMBoxes)

MMBox is an optional feature. It is logically associated with MMS Relay/Server and can provide message related functions like storage, retrieval or removal.

External servers

Several other services can be included in the MMS network, e.g. Email server, SMS server or Fax. The MMS Relay/Server provides interoperability between the User Agent and these services.

MMS value added applications (MMS VAS)

An MMS VAS application works like extended user agents providing additional features that are not available from the user agent. [5]

2.1. Addressing

MMS must support the use of E-Mail addresses or MSISDN (E.164) or both. MMS may support the use of service provider specific addresses to address the recipient of an MM. MMS may also support short codes to address Value Added Services.

MMS connectivity across different networks (MMSEs) is provided based on Internet protocols. Therefore it is important that when MSISDN addresses are used they can be translated to routable addresses. Therefore each MMS network entity should be assigned a unique domain name, e.g. mms.servicea.net.

The sender MMS Relay/Servers should make this mapping before routing forward the message to the recipient's MMS Relay/Server.

MMS must support address hiding, i.e. anonymous messages where the sender's address is not shown to the recipient's MMS User Agent.

2.2. MMS Transaction model

The MMS service works by exchanging transactions between the user agent and the MMS Relay/Server. Transactions carry information and affect state changes on both devices. The transactions are not dependent on what

type of client the message is sent from or what client has received the message. The other end may be another MMS client, or a client of another type of wireless system, or perhaps an email server. Transaction flows occur basically on three occasions:

- A client sends an MM and receives a delivery notice
- A client retrieves an MM immediately
- A client retrieves an MM after a period of time.

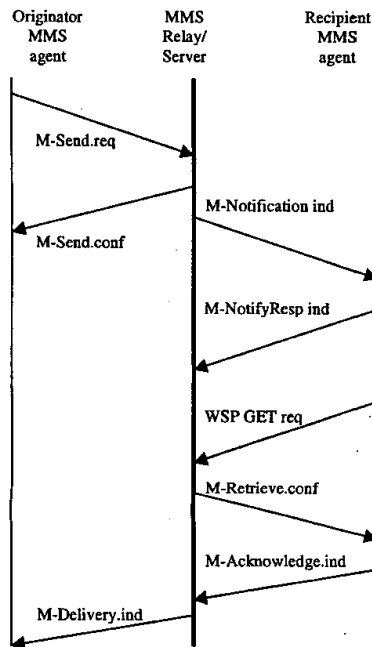


Figure 2. Example of MMS transaction flow [2]

An example of an MMS transaction flow is shown in Figure 2. A user agent sends a M.Send.req to the MMS Relay/Server using the WSP/HTTP POST method. The MMS Relay/Server sends the M-Notification.ind back to the user agent using the WAP PUSH method. The recipient MMS user agent retrieves the message by utilizing the WSP/HTTP GET method on the URI provided by MMS Relay/Server M-Notification.ind. The MMS Relay/Server returns M-Retrieve.conf to the recipient user agent. The recipient user agent may send M-Acknowledge.ind to indicate that the message really was received and finally the MMS Relay/Server may optionally send an M-Delivery.ind message to the originating client to report the delivery.

2.3. MMS Message Structure

The MMS protocol data units (PDUs) consist of MMS headers and the message body coded with MIME multipart technique. The body can contain any content type. The MIME multipart is used also in email systems to combine the email body and attachments together and is therefore compatible. The MMS headers part contains MMS specific information about the message, mainly how the message will be transferred from originator terminal to recipient. The content type of the PDUs is application/vnd.wap.mms-message.

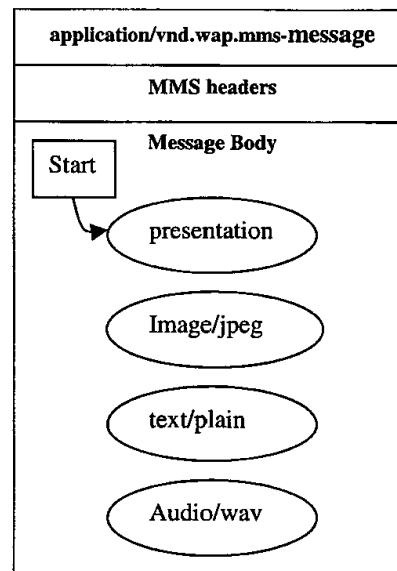


Figure 3. MMS data encapsulation model [4]

The message consists of an optional presentation part and multimedia objects in separate parts as shown in Figure 3. The order of the parts is not significant. The presentation part contains information on how the display and the loudspeakers of the user terminal will present the parts. If the presentation part is not present or the terminal does not support presentation it is up to the terminal software to present the message. The message body is used only when the message is sent or retrieved. The presentation technique can be e.g. WML (wireless markup language) or SMIL (synchronized multimedia presentation language), which is an implementation of XML (extensible markup language).

3. Content and service delivering

One big problem from the content and service provider point of view is interoperability. The problem consists of device-to-server interoperability and server-to-server interoperability. Device-to-server interoperability consists of the implementation of MM1-interface [2], the device SMIL [6] and content format support. The MM1-interface is used for MMS communication between the MMS user agent and the MMS Relay/Server (MMSC). The interface describes message flows as in Figure 2.

Server-to-server support includes MM4-interface and value added service-interface (MM7-interface) interoperability [2]. The MM4-interface describes interworking between different MMSEs. The MM7-interface is used to transfer multimedia messages between MMSC and VAS applications. Basically the MM7-interface is not a server-to-server interface, but from the VAS developer point of view it can be regarded as such. The VAS developer has to support different MMS centers, without changing the application itself. That is why the MM7-interface should be similar for MMSC producers and VAS developers. Standardization tries to achieve this similarity. The MM7-interface has not been standardized yet, but it should be ready within 2002. Figure 4 shows the most important interfaces in the MMS environment.

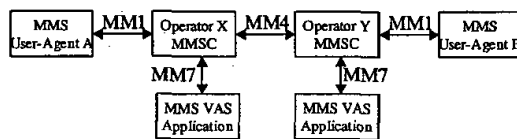


Figure 4. MM-interfaces [2]

The interoperability and device support problems were the reasons for planning the content delivering system:

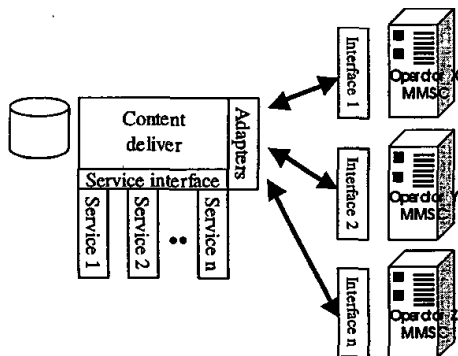


Figure 5. MMS content and service delivering system architecture

The whole system has been developed using the Java programming language and it is connected to the database using JDBC (Java Database Connection) technology. We have used the MySQL-database, but any database which has JDBC-interfaces could be used. The adapters are channels to different operator MMS centers. Because there are so many proprietary solutions in the MMSC MM7-interfaces, the adapters handle MM7-interface problems by providing one interface to all the operator centers. They also handle MM4-interface problems, because there are MM4-interoperability problems between the operator's MMS centers. MM4-interoperability problems are caused for example by differences in MMSC manufacturers' interface implementations. The content deliverer and service interface manage device interoperability issues. They handle issues like right service for a device, device SMIL-support, content format, and conversion. They also describe service interfaces, which can be done for example using RMI or HTTP. The content deliverer also handles user rights, transaction, message routing, content distribution from database, SMS routing, general service, and adapter management issues.

4. Case Web-camera picture to MMS

Using above described content delivering system we carried out a Web-camera case study. The web-camera was a normal camera which can be connected to network. In the case we made a service, which catches a specific Web-camera picture and then it makes an MM-message. The MM-message is delivered back to the MMS-terminal, which ordered the service. The following figure describes the case:

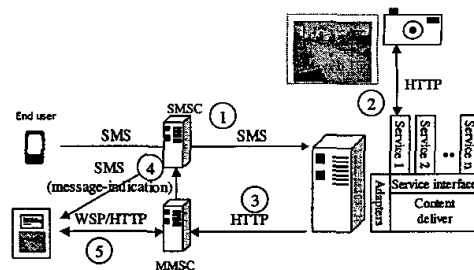


Figure 6. Case Web-camera service

1. At the beginning the end user orders a service by sending a short message (for example a text message WEBCAMERA EETU).
2. The specific message arrives from the SMS center to the content delivering system, which delivers the message to the right service. The service takes specific pictures and makes the MM-message. When the MM-message is ready it will be delivered to the content deliverer system.

3. The content deliverer system sends the message to the MMS center using operator interface over HTTP.
4. The MMS center sends a WAP-Push message indication over SMS to the specific device.
5. The device gets the message from the MMS center by using WSP/HTTP GET. First the GET-message goes to the WAP-gateway and then to the MMSC. In the future the WAP-gateway might be changed to an IP-based gateway [2].

Cases 4-5 are normal MMS delivering mechanisms in the operator network. Figure 7 below shows the case seen from the protocol stack point of view.

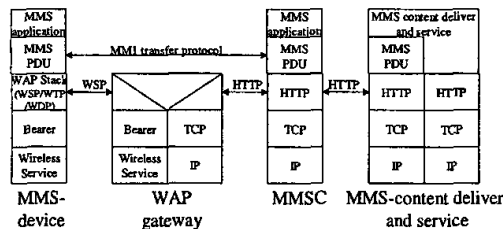


Figure 7. Case Web-camera service from the protocol point of view

The short message stack device is not included in this case, because it is well known and goes over mobile device control channels. After MM-message indication the MMS-device sends a WSP GET-message to MMSC. WSP-GET goes down to the WAP-stack to the selected bearer. The WAP-gateway gets the GET-message and converts it to the HTTP-GET. After that the HTTP-GET is delivered over TCP/IP to the MMSC. The MMSC handles the message and returns the selected message to the device through the WAP-gateway. The selected MM-message is delivered to the MMSC from the content delivering system over the HTTP. The Web-camera service gets the picture over the HTTP from the selected Web-camera.

5. Conclusion

The content delivering system is very easy to use, if it is possible to program the services. It gives a nice adapting mechanism to the operator server and easy-to-use interfaces to the services. There is, however, a tool for content uploading to the database, which makes the system easy to use also for content providers. The system also makes automatic content manipulation if necessary. For example a picture 640*480 pixels wide can be decreased to 128*96 pixels without weakening the sharpness. The content delivering system also contains SMIL-presentation issues, because it generates those SMIL-files automatically.

From the content and service provider point of view the MMS architecture and issues are too difficult to implement and use. There has to be a special system or tool that facilitates the use of complicated technology. The content or service providers are only interested in how easily they can deliver special contents and services to different mobile phone models. In the future the standardized MM7-interface will help the situation of the content and service provider, but some technical issues still have to be solved. The interoperability also makes content and service delivering easier, because there are not so many different variables anymore, which must be taken care off.

References

- [1] Sevanto Jarkko, "Multimedia Messaging Service for GPRS and UMTS", Proceedings of IEEE Wireless Communications and Networking Conference, vol.3, 1999, pp. 1422 -1426.
- [2] 3GPP, TS 23.140 V5.2.0 (2002-03), Multimedia messaging Service (MMS), Functional description, Stage 2 (Release 5).
- [3] WAP Forum, WAP-205-MMSArchOverview-20010425-a, Multimedia Messaging Service, Architecture Overview Specification, Version 25-April-2001.
- [4] WAP Forum, WAP-209-MMSEncapsulation-20020105-a, MMS Encapsulation Protocol, Version 05-Jan-2002.
- [5] WAP Forum, WAP-206-MMSCTR-20020115-a, Multimedia Messaging Service, Client Transactions Specification, Version 15-Jan-2002.
- [6] MMS Conformance document, version 2.0.0, February 2002, <http://www.forum.nokia.com>