Companion Website

mobile messaging technologies and services

Second Edition

SMS, EMS and MMS

Gwenaël Le Bodic

WILEY

message_

MOBILE MESSAGING
TECHNOLOGIES AND SERVICES
SMS, EMS and MMS

Second Edition

Gwenaël Le Bodic
Vodafone

John Wiley & Sons, Ltd

MOBILE MESSAGING

TECHNOLOGIES AND SERVICES

MOBILE MESSAGING
TECHNOLOGIES AND SERVICES
SMS, EMS and MMS

Second Edition

Gwenaël Le Bodic
Vodafone

John Wiley & Sons, Ltd

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-470-01143-2

Typeset in 10/12pt Times by Thomson Press (India) Limited, New Delhi.
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire.
This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

Be liberal in what you accept, and conservative in what you send

Jonathan B. Postel
RFC-1122 (originates in RFC-760)

Contents

Preface

Is SMS already history? Definitely not if you consider the high SMS traffic volumes of today's mobile networks! SMS will certainly represent one of the major milestones in the history of mobile telephony. With SMS, users have forged their own dialect to cope with service limitations, composed their own communication groups or communities, and are enjoying new channels of interactions. Any GSM handset has SMS capabilities and if each GSM subscriber sends a message at the same time then more than 1 billion messages would fly over the radio waves of mobile networks worldwide. From an engineering perspective, technologies for SMS have reached a mature stage and no more extensions of SMS are being considered in standardization forums. Much focus is now given to the emerging Multimedia Messaging Service (MMS). The deployment of MMS only started a few years ago and MMS is already gaining wide support from the mobile industry with a fast growing handset penetration rate and worldwide operator support. MMS underlying technologies are still in an ongoing maturation process, and user experience with today's phones has already greatly improved compared with the one of early implementations. MMS has benefited from the introduction of a new generation of handsets with integrated multimedia capabilities such as color screens and built-in still and video cameras but also from the introduction of packet-based transmission in mobile networks. MMS opens the door to new business opportunities and is believed to be well positioned as the appropriate distribution channel for commercial contents (music downloads, alerts, news, etc.). The future will tell if MMS will follow SMS in becoming a true success story.

The first edition of this book was published in late 2002. It covered SMS, the Enhanced Messaging Service (EMS), and MMS. At that time, SMS was already a very successful service and MMS was emerging. Following the growing interest in MMS, I published a second book dedicated to MMS in late 2003. The second edition of this book builds up from the two previous books. All chapters have been completely revised according to the most recent developments in standardization, but also according to my own experiences, specifying embedded messaging solutions for a manufacturer of mobile devices and in designing MMS solutions for a large group of operators.

The first chapter of this book introduces the evolution of mobile telephony from its origins with the deployment of first generation systems, followed by the introduction of second generation systems supporting digital communications and packet-based transmissions. Emerging third generation systems are also described along with the latest developments in the standardization of techniques for digital rights management. Chapter 2 proposes to demystify the "too often" obscure structures and procedures of standardization organizations. It is of key importance to understand how these organizations produce the necessary messaging standards in order to design interoperable commercial solutions. Chapter 3 is dedicated to the Short Message Service. Firstly, it describes major use cases and quickly progresses into the technical details of the service. Chapter 4 focuses on the standard application-level extension of SMS known as the Enhanced Messaging Service. It explains how to create rich-media content and how to distribute this content over SMS as a transport bearer. Chapters 5 and 6 are entirely dedicated to the Multimedia Messaging Service. Chapter 5 explains the service use cases, the overall architecture, and describes how multimedia messages can be designed. Chapter 6 focuses on protocol aspects, presenting the technical realization of each of the MMS interfaces. A set of appendices complement the contents of all chapters and a comprehensive index has been compiled for this book to represent a practical reference companion for solution architects, telecommunication engineers, standardization practitioners, instructors, and students.

I must admit that one of the primary reasons for writing books is that it represented for me a very good opportunity for pretending not to have enough time for washing dishes by hand, hovering the flat, and tidying up my desk. My wife, Marie-Amélie, recently discovered the trick and it became a real challenge to finish this book according to the agreed timelines, while being obliged to do the hand-washing of dishes at the same time. We recently purchased a second-hand dish-washer. This really improves our daily living. I have now realized that I do not need to write books anymore to pretend not to have enough time for washing dishes. I may still consider writing articles from time to time, a good reason for pretending not to have the time to clean the table and put dirty dishes in the beloved dish-washer.

I would like to gratefully acknowledge the time and effort of many people who reviewed the content of this book. The book has benefited from constructive comments from experts involved in various MMS activities (standardization bodies, mobile network operators, handset manufacturers, and third party application developers). In particular, I am thankful to Eskil Åhlin, Stéphane Augui, Philippe Bellordre, Luis Carroll, Dave Chen, François Courau, Philippe Delaloy, Cyril Fenard, Peter Freitag, Arthur Gidlow, Pierre Grenaille, Ian Harris, Michael Ishizue, Hervé Languille, Josef Laumen, Marie-Amélie Le Bodic, Arnaud Le Roy, Bernd Mielke, Ngoc Tanh Ly, Jérome Marcon, Thibaut Mienville, Thomas Picard, Jean-Luc Ricoeur, Friedhelm Rodermund, Andreas Schmidt, José Soares, Frank Timphus, Frédéric Villain, Paul Vincent, and Wilfried Zeise.

I would also like to acknowledge all readers of the first edition and in particular those who provided feedback. I have used this valuable feedback, whenever possible, to improve the accuracy and readability of this second edition.

The team at John Wiley & Sons, Ltd involved in the production of this book, provided excellent support and guidance. Particularly, I am grateful to Mark Hammond and Sarah Hinton for their continuous support during the entire process.

In addition, I am thankful to Alcatel Business Systems, Bijitec, Siemens AG, and Sony Ericsson for providing illustrations for this book.

The bibliography lists a number of standards that are useful for exploring further topics introduced in this book. Pointers to these standards and other useful resources are available from this book's companion website at:

http://www.lebodic.net/mms_resources.htm

Gwenaël Le Bodic, PhD

About the Author

Gwenaël Le Bodic is a messaging architect for the Vodafone group, located in Germany. In the scope of his activities for Vodafone, he is involved in the design of messaging solutions for large multi-operator environments. He also contributes to the development of system interworking to enable the exchange of multimedia messages between operators.

Previously, Gwenaël Le Bodic was a messaging and standardization expert for Alcatel's mobile phone division, located in France. His activities for Alcatel included participating and contributing to the development of messaging services and technologies in the scope of the 3GPP and OMA standardization forums. He has been responsible for the design of the software architecture of the embedded multimedia messaging solution for Alcatel's first two MMS phones.

A certified engineer in computer sciences, Gwenaël Le Bodic obtained a PhD in mobile communications from the University of Strathclyde, Glasgow. He is the author of many research publications in the field of mobile communications. He wrote the first edition of this book *Mobile Messaging, Technologies and Services* (John Wiley & Sons, Ltd, November 2002) and also a book focusing on MMS, *Multimedia Messaging Service, an engineering approach to MMS* (John Wiley & Sons, Ltd, October 2003).

Gwenaël Le Bodic can be contacted at gwenael@lebodic.net

Typographic conventions

In this book, the following typographic conventions are used:

Typeface or symbol	**Meaning/used for**	**Example**
Courier	Refers to a system command, name of protocol operation, meta language tag, or any computer output.	The two most common object identifiers that are used are the Content-ID and the Content-Location.
<courier>	Serves as a placeholder for variable text that can be replaced as appropriate in the context of use.	Fields <to-address> and <from-address> can take two forms.
[<courier>]	Serves as a placeholder for optional variable text that can be replaced as appropriate in the context of use.	[<from-address>]
[Times]	Refers to a document (e.g., standard, book, article) listed in the References section.	[3GPP-23.140]
Italic	Emphasizes a new word or term of significance.	The application segments the message into several pieces called *message segments.*
●	For description of protocol data units, this symbol indicates that the corresponding parameter is mandatory.	
○	For description of protocol data units, this symbol indicates that the corresponding parameter is optional.	
©	For description of protocol data units, this symbol indicates that the corresponding parameter appears conditionally in the unit.	
0xA9	Represents a hexadecimal value prefixed by "0x."	"0x1A" represents the hexadecimal value 1A (decimal value 26).

1

Basic Concepts

This chapter outlines the basic concepts of mobile communications systems and presents the required background information necessary for a clear understanding of this book. First, an overview of the evolution of mobile communications systems is provided. This encompasses the introduction of first generation analog systems supporting only voice communications to the recent deployment of third generation systems supporting voice and multimedia services. The Global System for Mobile Communication, commonly known as GSM, has been a major breakthrough in the domain of mobile communications. Elements composing a typical GSM network are presented. Another important milestone is the introduction of the General Packet Radio Service (GPRS) allowing the support of packet-based communications in evolved GSM networks. The architecture of a GPRS network is presented. Recently deployed are Universal Mobile Telecommunications Systems (UMTS). These systems support advanced multimedia services requiring high data rates. UMTS services and supporting technologies are also introduced in this chapter. Additionally, the Wireless Application Protocol (WAP) is described. WAP is an enabling technology for developing services such as browsing and multimedia messaging. An overview of latest digital rights management methods is also provided. The last section of this chapter provides pointers to books and reference articles for anybody wishing to further explore the topics covered in this chapter.

1.1 Generations of Mobile Communications Networks

In France, in 1956, a very basic mobile telephony network was implemented with vacuum electronic tubes and electron-mechanical logic circuitry. These devices used for wireless communications had to be carried in car boots. In these early days of mobile telephony, service access was far from being ubiquitous and was reserved for a very limited portion of the population. Since the introduction of this experimental network, mobile communications technologies benefited from major breakthroughs commonly categorized in three generations. In the 1980s, *first generation (1G) mobile systems* arrived in Nordic countries. These first generation systems were characterized by analog wireless communications and limited support for user mobility.

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2

Digital communications technology was introduced with *second generation (2G) mobile systems* in the 1990s. Second generation systems are characterized by the provision of better quality voice services available to the mass market. Second generation systems benefited from the cellular concept in which scarce radio resources are used simultaneously by several mobile users without interference. The best known 2G system is the Global System for Mobile Communication (GSM) with the billionth GSM user connected in the first quarter of 2004. Other major 2G systems include cdmaOne (based on CDMA technology), with users in the Americas and Asia, and Japanese Personal Data Cellular (PDC) with the iMode technology for mobile Internet.

Early 2004, first *third generation (3G) mobile systems* have been deployed in several European countries. With 3G systems, various wireless technologies converge with Internet technologies. Third generation services encompass a wide range of multimedia and cost-effective services with support for worldwide user mobility. The migration to 3G systems is facilitated by the introduction of intermediary evolved 2G systems, also known as *2.5G systems*.

1.2 Telecommunications Context: Standardization and Regulation

In the telecommunications environment, *Standard Development Organizations* (SDOs) provide the necessary framework for the development of standards. These standards are technical documents[1] defining or identifying the technologies enabling the realization of telecommunication network technologies and services. The prime objective of SDOs is to develop and maintain widely accepted standards allowing the introduction of attractive services over interoperable networks. The actors that are involved in the standardization process are network operators, manufacturers, and third party organizations such as content providers, equipment testers, and regulatory authorities. One of the main objectives of telecommunications regulation authorities is to ensure that the telecommunications environment is organized in a sufficiently competitive environment and that the quality of service offered to subscribers is satisfactory.

In the early days of mobile communications, various regional SDOs developed specifications for network technologies and services independently. This led to the development of heterogeneous networks where interoperability was seldom ensured. The lack of interoperability of first generation mobile systems prevented the expansion of a global international mobile network that would have certainly greatly improved user experience. With second and third generations systems, major SDOs decided to gather their efforts in order to ensure that mobile communication networks will appropriately interoperate in various regions of the world. In 1998, such an effort was initiated by several SDOs including ARIB (Japan), ETSI (Europe), TTA (Korea), TTC (Japan), and T1 (USA). The initiative was named the Third Generation Partnership Project (3GPP). The 3GPP standardization process is presented in Chapter 2.

[1] Technical documents are also known as technical specifications, reports, or recommendations.

1.3 Global System for Mobile Communication

Before the introduction of the *Global System for Mobile Communication* (GSM), mobile networks implemented in different countries were usually incompatible. This incompatibility made impracticable the roaming of mobile users across international borders. In order to get around this system incompatibility, the Conférence Européenne des Postes et Télécommunications (CEPT) created the Groupe Spécial Mobile[1] committee in 1982. The main task of the committee was to standardize a pan-European cellular public communication network in the 900 MHz radio band. In 1989, the European Telecommunications Standard Institute (ETSI) took over the responsibility for the maintenance and evolution of GSM specifications. In 2000, this responsibility was transferred to 3GPP. The initiative was so successful that networks compliant with the GSM standard have now been developed worldwide. Variations of the GSM specification have been standardized for the 1800 and 1900 MHz bands and are known as DCS 1800 and PCS 1900, respectively. In March 2004, the GSM association[2] reported a total number of 1046.8 million subscribers distributed over 207 countries.

A GSM network is characterized by digital voice communications and support of low-rate data services. The GSM air interface is based on Time Division Multiple Access (TDMA). With TDMA, a radio band is shared by multiple subscribers by allocating one or more timeslots on given radio carriers to each subscriber. With GSM, the transfer of data can be carried out over circuit-switched connections. For these data communications, bit rates up to 14.4 Kbps can be achieved on single-slot connections. The single-slot configuration is called Circuit Switched Data (CSD). Higher bit rates up to 57.6 Kbps can be attained by allocating more than one slot for a data connection. This multi-slot configuration is called High Speed CSD (HSCSD).

One of the most popular GSM services is the Short Message Service (SMS). This service allows SMS subscribers to exchange short text messages. An in-depth description of this service is provided in Chapter 3. An application-level extension of SMS in the form of the Enhanced Messaging Service (EMS) is presented in Chapter 4.

1.3.1 Cellular Concept

Radio bands available for wireless communications in mobile networks represent very scarce resources. In order to efficiently use these resources, GSM networks are based on the *cellular concept*. With this concept, the same radio resources (characterized by a frequency band and a timeslot) can be utilized simultaneously by several subscribers without interference if they are separated by a minimum distance. The minimum distance between two subscribers depends on the way radio waves propagate in the environment where the two subscribers are located (e.g., presence of buildings, etc.). In a GSM network, the smaller the cells, the higher is the frequency reuse factor, as shown in Figure 1.1.

In a GSM network, a fixed base station transceiver manages the radio communications for all mobile stations located in a cell. Each geometrical cell in Figure 1.1 represents the radio coverage of one single base station.

[1] The name Groupe Spécial Mobile was later translated to Special Mobile Group (SMG).
[2] http://www.gsmworld.com

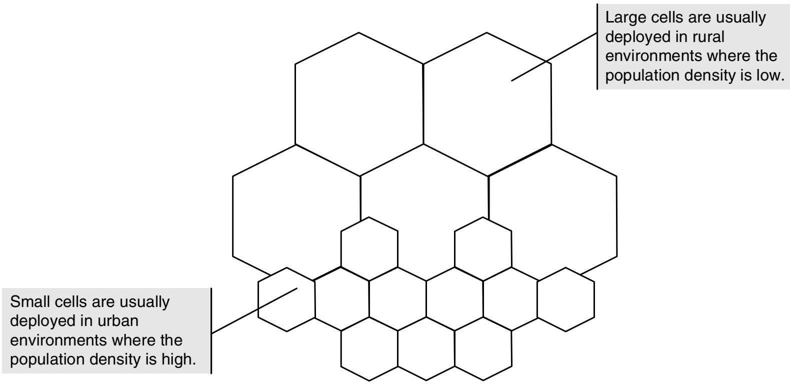


Figure 1.1 Cellular concept

1.3.2 GSM Architecture

The main elements of the GSM architecture [3GPP-23.002] are shown in Figure 1.2. The GSM network is composed of three subsystems: the *Base Station Subsystem* (BSS), the *Network Subsystem* (NSS), and the *Operation Subsystem* (OSS). The OSS implements functions that allow the administration of the mobile network. For the sake of clarity, elements of the OSS are not represented in the GSM architecture shown in Figure 1.2. Elements of the BSS and NSS are further described in the following sections.

1.3.3 Mobile Station

The *Mobile Station* (MS) is a device that transmits and receives radio signals within a cell site. A mobile station can be a basic mobile handset, as shown in Figure 1.3, or a more complex Personal Digital Assistant (PDA). Mobile handset capabilities include voice communications, messaging features, and phone book management. In addition to these basic capabilities, a PDA is usually shipped with an Internet microbrowser and an advanced Personal Information Manager (PIM) for managing contacts and calendaring/scheduling entries. When the user is moving (i.e., while driving), network control of MS connections is switched over from cell site to cell site to support MS mobility. This process is called *handover*.

The mobile station is composed of the *Mobile Equipment* (ME) and the *Subscriber Identity Module* (SIM). The unique *International Mobile Equipment Identity* (IMEI) stored in the ME identifies uniquely the device when attached to the mobile network.

The SIM is usually provided by the network operator to the subscriber in the form of a smart card. The microchip is often taken out of the smart card and directly inserted into a dedicated slot in the mobile equipment. A SIM microchip is shown in Figure 1.4.

Today's mobile stations can be connected to an external device such as a PDA or a personal computer. Such an external device is named a *Terminal Equipment* (TE) in the GSM architecture.

A short message is typically stored in the mobile station. Most handsets have SIM storage capacities. High-end products sometimes complement the SIM storage capacity with

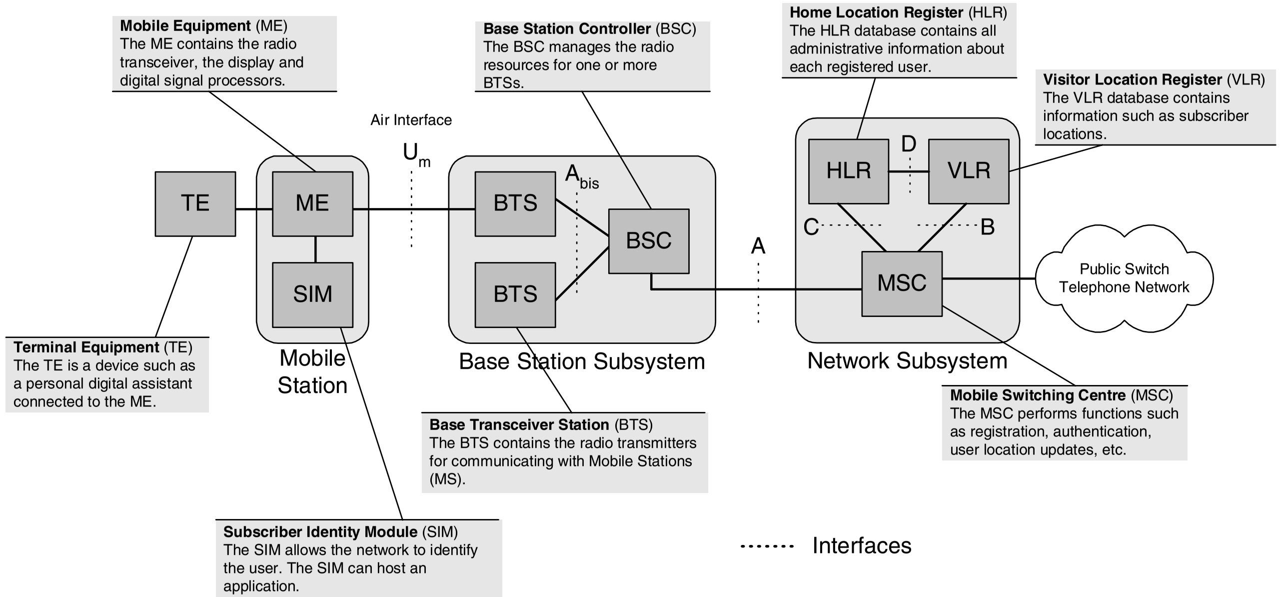


Figure 1.2 GSM architecture

Figure 1.3 Mobile station handset – reproduced by permission of Alcatel Business Systems

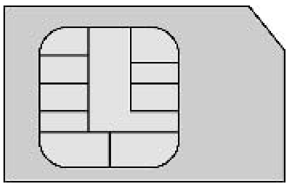


Figure 1.4 SIM microchip

additional storage in the mobile equipment itself (e.g., flash memory). It is now common to find handsets shipped with a PIM. The PIM is usually implemented as an ME internal feature and enables elements such as calendar entries, memos, phonebook entries, and of course messages to be stored in the ME. These elements are managed, by the subscriber, with a suitable graphical user interface. These PIM elements remain in the PIM even when the SIM is removed from the mobile handset. Alternatively, simple elements such as short messages and phonebook entries can be directly stored in the SIM. A SIM can contain from 10 short messages to 50 short messages on high-end solutions. Storing elements in the SIM allows messages to be retrieved from any handset simply by inserting the SIM in the desired handset. The benefit of storing messages in the ME is that the ME storage capacity is often significantly larger than the SIM storage capacity.

1.3.4 Base Transceiver Station

The *Base Transceiver Station* (BTS) implements the air communications interface with all active MSs located under its coverage area (cell site). This includes signal modulation/ demodulation, signal equalizing, and error coding. Several BTSs are connected to a single

Base Station Controller (BSC). In the United Kingdom, the number of GSM BTSs is estimated around several thousands. Cell radii range from 10 to 200 m for the smallest cells to several kilometers for the largest cells. A BTS is typically capable of handling 20–40 simultaneous communications.

1.3.5 Base Station Controller

The BSC supplies a set of functions for managing connections of BTSs under its control. Functions enable operations such as handover, cell site configuration, management of radio resources, and tuning of BTS radio frequency power levels. In addition, the BSC realizes a first concentration of circuits towards the MSC. In a typical GSM network, the BSC controls over 70 BTSs.

1.3.6 Mobile Switching Center and Visitor Location Register

The *Mobile Switching Center* (MSC) performs the communications switching functions of the system and is responsible for call set-up, release, and routing. It also provides functions for service billing and for interfacing other networks.

The *Visitor Location Register* (VLR) contains dynamic information about users who are attached to the mobile network including the user's geographical location. The VLR is usually integrated to the MSC.

Through the MSC, the mobile network communicates with other networks such as the Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), Circuit Switched Public Data Network (CSPDN), and Packet Switched Public Data Network (PSPDN).

1.3.7 Home Location Register

The *Home Location Register* (HLR) is a network element containing subscription details for each subscriber. An HLR is typically capable of managing information for hundreds of thousands of subscribers.

In a GSM network, signaling is based on the Signaling System Number 7 (SS7) protocol. The use of SS7 is complemented by the use of the Mobile Application Part (MAP) protocol for mobile specific signaling. In particular, MAP is used for the exchange of location and subscriber information between the HLR and other network elements such as the MSC. For each subscriber, the HLR maintains the mapping between the *International Mobile Subscriber Identity* (IMSI) and the *Mobile Station ISDN Number* (MSISDN).

For security reasons, the IMSI is seldom transmitted over the air interface and is only known within a given GSM network. The IMSI is constructed according to [ITU-E.212] format. Unlike the IMSI, the MSISDN identifies a subscriber outside the GSM network. The MSISDN is constructed according to [ITU-E.164] format (e.g., +33612345678 for a French mobile subscriber).

1.4 General Packet Radio Service

In its simplest form, GSM manages voice and data communications over circuit-switched connections. The General Packet Radio Service (GPRS) is an extension of GSM which allows subscribers to send and receive data over packet-switched connections. The use of GPRS is particularly appropriate for applications with the following characteristics:

- bursty transmission (for which the time between successive transmissions greatly exceeds the average transfer delay);
- frequent transmission of small volumes of data;
- infrequent transmission of large volumes of data.

These applications do not usually need to communicate permanently. Consequently, the continuous reservation of resources for realizing a circuit-switched connection does not represent an efficient way to exploit scarce radio resources. The basic concept behind the GPRS packet-based transmission lies in its ability to allow selected applications to share radio resources by allocating radio resources for transmission only when applications have data to transmit. Once the data have been transmitted by an application, radio resources are released for use by other applications. Scarce radio resources are used more efficiently with this mechanism. GPRS allows more radio resources to be allocated to a packet-based connection than to a circuit-switched connection in GSM. Consequently, a packet-based connection usually achieves higher bit rates (up to 171.2 Kbps) by using a multislot configuration for uplinks and downlinks as shown in Table 1.1. For instance, a mobile station of multislot GPRS class 6 can have a maximum of three slots allocated to the downlink and a maximum of two slots allocated to the uplink. However, for such a mobile station, a maximum of four slots only can be active at a time for both uplink and downlink. The capacity of each slot depends on the channel encoding used. Four channel encoding schemes are available in GPRS with distinct levels of error protection and are typically selected according to the quality of the radio environment. GPRS can offer ‘‘always on’’ connections (sending or receiving data at any time).

Table 1.1 Multislot GPRS classes

Multislot GPRS class	Downlink slots	Uplink slots	Active slots
1	1	1	2
2	2	1	3
3	2	2	3
4	3	1	4
5	2	2	4
6	3	2	4
7	3	3	4
8	4	1	5
9	3	2	5
10	4	2	5
11	4	3	5
12	4	4	5

1.4.1 GPRS Architecture

The main elements composing the GPRS architecture [3GPP-23.060] are shown in Figure 1.5. A GPRS mobile station is categorized according to its capabilities to support simultaneous modes of operation for GSM and GPRS [3GPP-22.060] which are as follows:

- *Class A*: the mobile station supports simultaneous use of GSM and GPRS services (attachment, activation, monitoring, transmission, etc.). A *class A* mobile station may establish or receive calls on the two services simultaneously. The high complexity of designing *class A* devices makes them prohibitively expensive to produce and, therefore, these devices are typically not available for the mass market.
- *Class B*: the mobile station is attached to both GSM and GPRS services. However, the mobile station can only operate in one of the two services at a time.
- *Class C*: the mobile station is attached to either the GSM service or the GPRS service but is not attached to both services at the same time. Prior to establishing or receiving a call on one of the two services, the mobile station has to be explicitly attached to the desired service.

Before a mobile station can access GPRS services, it must execute a *GPRS attachment* procedure to indicate its presence to the network. After its GPRS attachment, the mobile station activates a Packet Data Protocol (PDP) context with the network in order to be able to transmit or receive data. This procedure is called *PDP context activation*.

The GPRS air interface is identical to that of the GSM network (same radio modulation, frequency bands, and frame structure). GPRS is based on an evolved GSM base station subsystem. However, the GPRS core network relies on a GSM network subsystem in which two additional network elements have been integrated: serving and gateway GPRS support nodes. In addition, *Enhanced Data Rate for Global Evolution* (EDGE) can be supported to improve GPRS performances by introducing an enhanced modulation scheme.

1.4.2 Serving GPRS Support Node

The *Serving GPRS Support Node* (SGSN) is connected to one or more base station subsystems. It operates as a router for data packets for all mobile stations present in a given geographical area. It also keeps track of the location of mobile stations and performs security functions and access control.

1.4.3 Gateway GPRS Support Node

The *Gateway GPRS Support Node* (GGSN) provides the point of attachment between the GPRS domain and other data networks such as the Internet or corporate networks. An *Access Point Name* (APN) is used by the mobile user to establish the connection to the required destination network.

1.5 Universal Mobile Telecommunications System

Since 1990, focus has been given to the standardization of third generation mobile systems. The International Telecommunication Union (ITU) has initiated the work on a set of

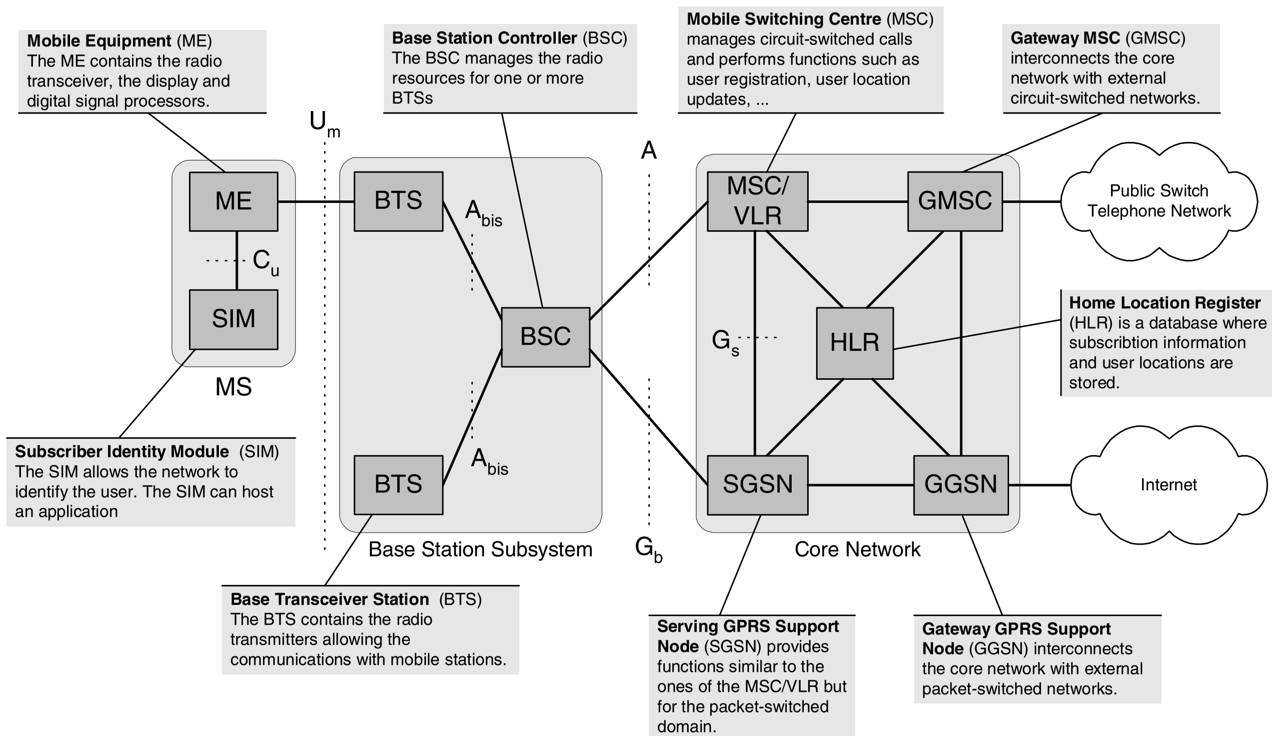


Figure 1.5 GPRS architecture

standards named the International Mobile Telecommunications 2000 (IMT-2000) for the definition of technologies and services for 3G systems. In this family of IMT-2000[1] standards, the Universal Mobile Telecommunications System (UMTS) encompasses the definition of new radio access techniques along with a new service architecture. UMTS aims at providing services such as web browsing, messaging, mobile commerce, videoconferencing, and other services to be developed according to emerging subscribers' needs with the following objectives:

- high transmission rates encompassing circuit-switched and packet-switched connections;
- high spectral efficiency and overall cost improvement;
- definition of common radio interfaces for multiple environments;
- portability of services in various environments (indoor, outdoor, suburban, urban, rural, pedestrian, vehicular, satellite, etc.). This service portability is also known as the *Virtual Home Environment* concept [3GPP-22.121][3GPP-23.127].

The provisioning of services in heterogeneous environments is enabled with an Open Service Architecture (OSA) [3GPP-22.127][3GPP-23.127]. UMTS extends 2G voice and data capabilities to multimedia capabilities with access to higher bandwidth targeting 384 Kbps for full area coverage and 2 Mb/s for local area coverage. UMTS is expected to become the basis for new mobile telecommunications networks with highly personalized and user-friendly services. UMTS should provide a convergence of communications technologies such as satellite, cellular radio, cordless, and wireless LANs. The network operator NTT Docomo introduced 3G services to the Japanese market in October 2001. Elsewhere, the commercial introduction of UMTS networks and services for the mass market started in 2004.

1.5.1 3G Services

Second generation networks provide voice and limited data services. In addition to these 2G services, 3G systems offer multimedia services adapted to the capabilities of multimedia devices and network conditions with a possibility to provide some content specifically formatted according to the subscriber location. The UMTS Forum in [UFRep9][UFRep13] classifies 3G services into the following six groups as illustrated in Figure 1.6:

- *Mobile Internet access*: a mobile access to the Internet with service quality close to the one offered by fixed Internet Service Providers. This includes full Web access, file transfer, electronic mail, and streaming video and audio.
- *Mobile Intranet/Extranet access*: a secure framework for accessing corporate Local Area Networks (LANs) and Virtual Private Networks (VPNs).
- *Customized infotainment*: a device-independent access to personalized content from mobile portals.
- *Multimedia messaging service*: a means of exchanging messages containing multimedia contents including text, images, and video and audio elements. The multimedia messaging service can be considered as an evolution of SMS where truly multimedia messages can

[1] IMT-2000 was formerly known as Future Public Land Mobile Telecommunications System (FPLMTS).

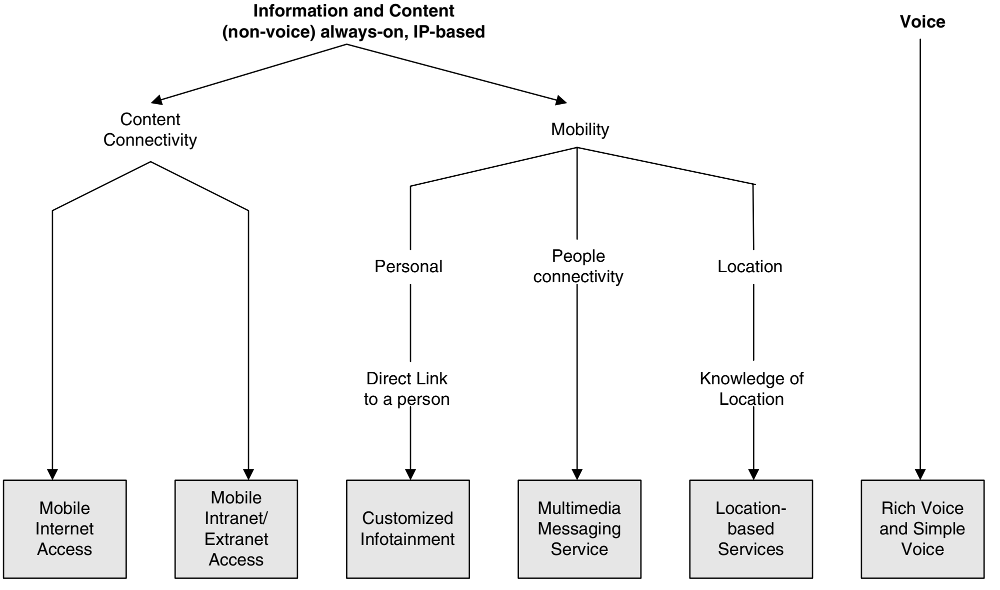


Figure 1.6 3G service categories – source UMTS Forum [UF-Rep-9]

be exchanged between subscribers. An in-depth description of the multimedia messaging service is provided in Chapters 5 and 6.

- *Location-based services*: location-aware services such as vehicle tracking, local advertisements, etc.
- *Rich voice and simple voice*: real-time, two-way voice communications. This includes Voice over IP (VoIP), voice-activated network access, and Internet-initiated voice calls. Mobile videophone and multimedia real-time communications should also be available on high-end multimedia devices.

In the scope of the 3GPP standardization process, the UMTS specification work was divided into two distinct phases. The first phase UMTS, named UMTS Release 99 (also known as Release 3), is a direct evolution from 2G and 2.5G networks (GSM and GPRS networks). The second phase UMTS, also known as UMTS Release 4/5, is a complete revolution introducing new concepts and features.

1.5.2 First Phase UMTS

The UMTS architecture [3GPP-23.101] has to meet the requirements of various UMTS services. These requirements range from real-time voice traffic and bursty data access to mixed multimedia traffic. UMTS is intended to offer a true global service availability. To meet this objective, the UMTS architecture includes terrestrial segments complemented by satellite constellations where necessary.

1.5.3 First Phase UMTS Architecture

The first phase UMTS architecture is based on evolved GSM and GPRS core networks and a specifically tailored *Universal Terrestrial Radio Access Network* (UTRAN). Two duplexing methods, defining how the received signal is separated from the transmitted signal, have been defined as follows:

- *Universal Terrestrial Radio Access/Time Division Duplex* (UTRA/TDD): this method achieves bi-directional transmission by allowing the use of different time slots over the same radio carrier for the transmission of sent and received signals.
- *Universal Terrestrial Radio Access/Frequency Division Duplex* (UTRA/FDD): this method achieves bi-directional transmission by allowing sent and received signals to be transmitted over two separate and symmetrical radio bands for the two links.

The name *Wideband CDMA* (WCDMA) is also used to identify the two UTRA operating modes (TDD and FDD). Elements composing the first phase UMTS architecture are shown in Figure 1.7.

Elements of the UMTS architecture are grouped into three subsystems: the *User Equipment* (UE), the access network (UTRAN), and the switching and routing infrastructure, also known as the *Core Network* (CN). Elements of the UMTS architecture support both circuit-switched connections and packet-switched connections.

1.5.4 User Equipment

The UE, usually provided to the subscriber in the form of a handset, is itself composed of a *Mobile Equipment* (ME) and a *UMTS Subscriber Identity Module* (USIM). The ME contains the radio transceiver, the display, and digital signal processors. The USIM is a 3G application on a *UMTS IC card* (UICC) which holds the subscriber identity, authentication algorithms, and other subscriber-related information. The ME and USIM are interconnected via the Cu electrical interface whereas the UE is connected to the UTRAN via the Uu radio interface. A UE always supports at least one of the operating modes of UTRA: TDD or FDD. In order to allow a smooth transition to UMTS, it is expected that UEs will initially be capable of communicating with legacy systems such as GSM and GPRS. UMTS UEs supporting legacy systems are called multi-mode UEs. The 3GPP classifies multi-mode UEs into the following four categories [3GPP-21.910]:

- *Type 1*: *type 1* user equipment operates in one single mode at a time (GSM or UTRA). It cannot operate in more than one mode at a time. While operating in a given mode, the user equipment does not scan for or monitor any other mode and switching from one mode to another is done manually by the subscriber.
- *Type 2*: while operating in one mode, *type 2* user equipment can scan for and monitor another mode of operation. The user equipment reports to the subscriber on the status of another mode by using the current mode of operation. *Type 2* user equipment does not support simultaneous reception or transmission through different modes. The switching from one mode to another is performed automatically.

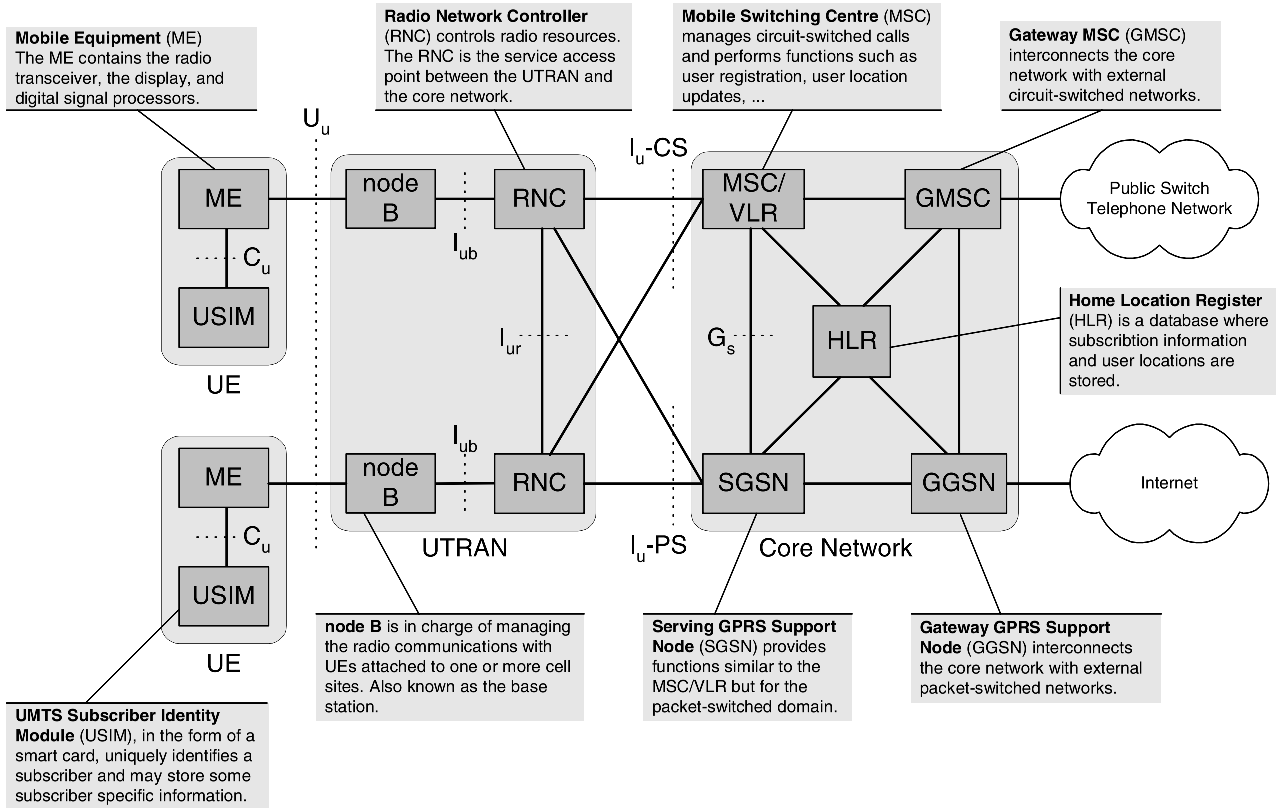


Figure 1.7 UMTS architecture (1st phase)

- *Type 3*: *type 3* user equipment differs from *type 2* user equipment by the fact that the *type 3* UE can receive more than one mode at a time. However, a *type 3* UE cannot emit simultaneously in more than one mode. Switching from one mode to another is performed automatically.
- *Type 4*: *type 4* user equipment can receive and transmit simultaneously in more than one mode. Switching from one mode to another is performed automatically.

1.5.5 UTRA Network

The UTRAN is composed of *nodes B* and *Radio Network Controllers* (RNCs). The *node B* is responsible for the transmission of information in one or more cells, to and from UEs. It also participates partly in the system resource management. The *node B* interconnects with the RNC via the Iub interface. The RNC controls resources in the system and interfaces the core network.

1.5.6 First Phase UMTS Core Network

The first phase UMTS core network is based on an evolved GSM network subsystem (circuit-switched domain) and a GPRS core network (packet-switched domain). Consequently, the UMTS core network is composed of the HLR, the MSC/VLR, and the Gateway MSC (to manage circuit-switched connections), and the SGSN and GGSN (to manage packet-based connections).

1.5.7 Second Phase UMTS

The initial UMTS architecture presented in this chapter is based on evolved GSM and GPRS core networks (providing support for circuit-switched and packet-switched domains, respectively). The objective of this initial architecture is to allow mobile network operators to rapidly roll out UMTS networks on the basis of existing GSM and GPRS networks. From this first phase UMTS architecture, the next phase is to evolve to an architecture with a core network based on an enhanced packet-switched domain only. The objective is to allow a better convergence with the Internet by using IP-based protocols whenever possible. At the end of 1999, 3GPP started the work on the specification of an "all-IP" architecture. In this architecture, the MSC function is split into a control plane part (MSC server) and a user plane part (media gateway). The core network of the second phase UMTS is interfaced with an IP Multimedia Subsystem, abbreviated IMS [3GPP-22.228][3GPP-23.228]. IMS introduces the capability to support IP-based multimedia services such as Voice over IP (VoIP). In IMS, call control is managed with the Session Initiation Protocol (SIP), published by IETF in [RFC-3261], and all network elements are based on IPv6.

IMS can be seen as an additional layer providing signaling, control, and charging functions for IP-based communications. In this context, service platforms that will initially benefit from IMS are the ones enabling services such as Push-To-Talk (PTT), presence, and location-based services. Figure 1.8 shows the architecture of a second phase UMTS solution.

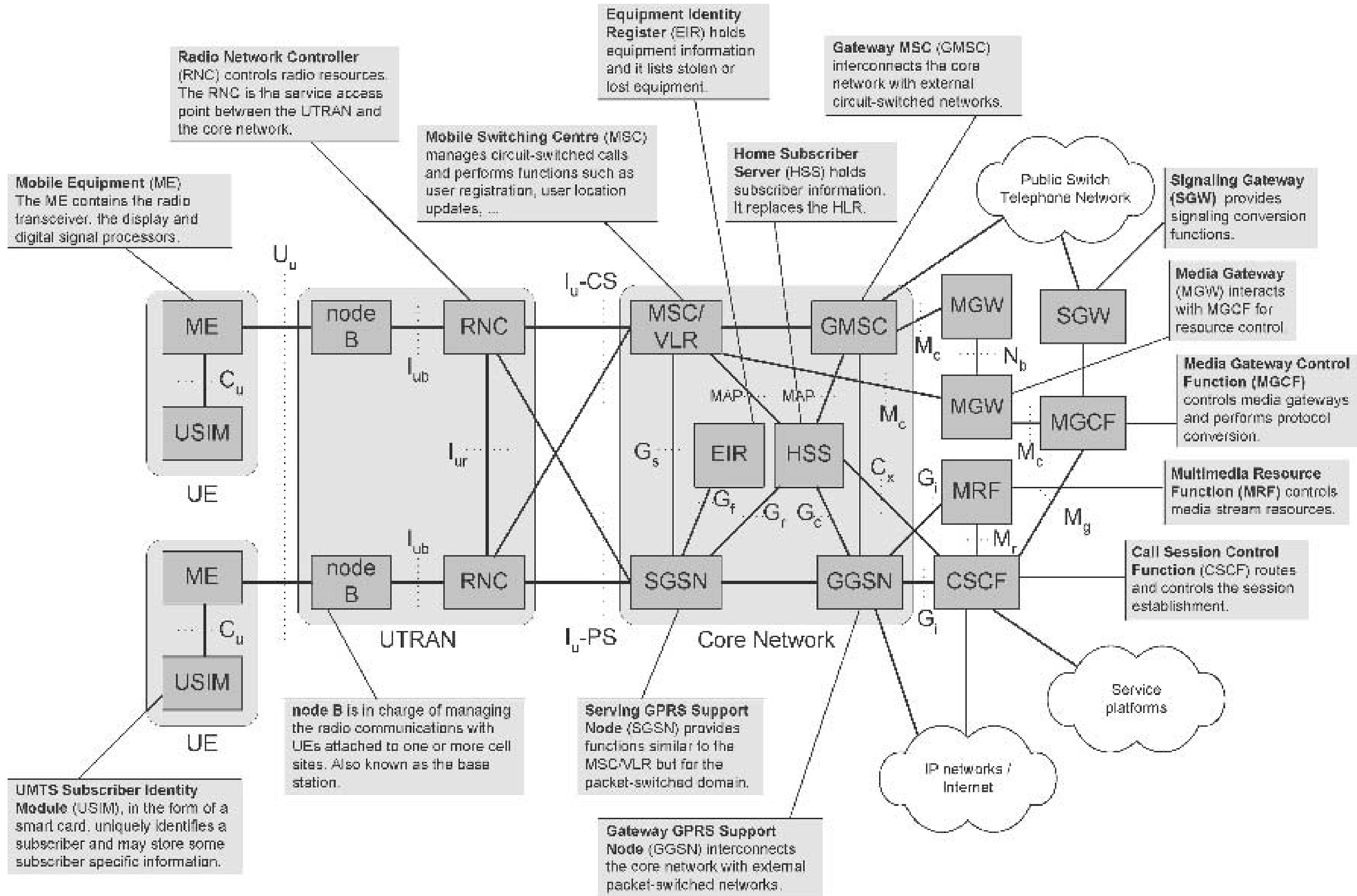


Figure 1.8 UMTS architecture (2nd phase)

1.6 Wireless Application Protocol

The Wireless Application Protocol (WAP) is the result of a collaborative work between many wireless industry players, carried out in the scope of the WAP Forum. The forum, launched in 1997 by Nokia, Phone.Com (now Openwave), Motorola, and Ericsson produced technical specifications enabling the support of applications over various wireless platforms (GSM, GPRS, UMTS, etc.). For this purpose, the WAP Forum identified and defined a set of protocols and content formats according to the standardization process presented in Chapter 2. In 2002, activities of the WAP Forum were transferred to another standardization organization: the Open Mobile Alliance.

1.6.1 Introduction to WAP

The WAP technology is an enabler for building applications (e.g., browsing, messaging, etc.) that run seamlessly over various wireless platforms. The objective of the WAP Forum was to provide a framework for the development of applications with a focus on the following aspects:

- *Interoperability*: applications developed by various parties and hosted on devices, produced by different manufacturers, interoperate in a satisfactory manner.
- *Scalability*: mobile network operators are able to scale services to subscribers' needs.
- *Efficiency*: the framework offers a quality of service suited to the capabilities of underlying wireless networks.
- *Reliability*: the framework represents a stable platform for deploying services.
- *Security*: the framework ensures that user data can be safely transmitted over a serving mobile network, which may not always be the home network. This includes the protection of services and devices and the confidentiality of subscriber data.

In line with these considerations, the WAP technology provides an application model close to the World Wide Web model (also known as the web model). In the web model, content is represented using standard description formats. Additionally, applications known as web browsers retrieve the available content using standard transport protocols. The web model includes the following key elements:

- *Standard naming model*: objects available over the web are uniquely identified by Uniform Resource Identifiers (URIs).
- *Content type*: objects available on the web are typed. Consequently, web browsers can correctly determine the type of a specific content.
- *Standard content format*: web browsers support a number of standard content formats such as the HyperText Markup Language (HTML).
- *Standard protocols*: web browsers also support a number of standard protocols for accessing content on the web. This includes the widely used HyperText Transfer Protocol (HTTP).

The WAP model borrows a lot from the successful web model. However, the web model, as it is, does not efficiently cope with constraints of today's mobile networks and devices. To cope with these constraints, the WAP model leverages the web model by adding the following improvements:

- The *push technology* allows content to be pushed directly from the server to the mobile device without any prior explicit request from the user.
- The adaptation of content to the capabilities of WAP devices relies on a mechanism known as the *User Agent Profile* (UAProf).
- The support of *advanced telephony features* by applications, such as the handling of calls (establishment and release of calls, placing a call on hold, or redirecting the call to another user, etc.).
- The *External Functionality Interface* (EFI) allows "plug-in" modules to be added to browsers and applications hosted in WAP devices in order to increase their overall capabilities.
- The *persistent storage* allows users to organize, access, store, and retrieve content from/to remote locations.
- The *Multimedia Messaging Service* (MMS) is a significant added value of the WAP model over the web model. It relies on generic WAP mechanisms such as the push technology and the UAProf to offer a sophisticated multimedia messaging service to mobile users. MMS is further described in Chapters 5 and 6.

The WAP model uses the standard naming model and content types defined in the web model. In addition, the WAP model includes the following:

- *Standard content formats*: browsers in the WAP environment, known as microbrowsers, support a number of standard content formats/languages including the Wireless Markup Language (WML) and the eXtensible HTML (XHTML). WML and XHTML are both applications of the eXtensible Markup Language (XML). See Box 1.1 for a description of markup languages for WAP-enabled devices.
- *Standard protocols*: microbrowsers communicate according to protocols that have been optimized for mobile networks, including the Wireless Session Protocol (WSP) and HTTP from the web model.

The first WAP technical specifications were made public in 1998 and have since evolved to allow the development of more advanced services. The major milestones for WAP technology were reflected in the availability of what the WAP Forum called "specifications suites." Each specification suite contains a set of WAP technical specifications providing a specific level of features as shown in Table 1.2.

With WAP specification suite 1.x, the WAP device communicates with an application server via a WAP gateway. Communication between the WAP device and the WAP gateway is performed over WSP. In addition, WAP specification suite 2.x allows a better convergence of wireless and Internet technologies by promoting the use of standard protocols from the web model.

Table 1.2 WAP Forum specification suites

WAP Forum specification suites	Delivery date	Description
WAP 1.0	April 1998	Basic WAP framework Almost no available commercial solutions since the published standards did not allow the design of interoperable solutions
WAP 1.1	June 1999	First commercial solutions supporting: - Wireless Application Environment (WAE) - Wireless Session Protocol (WSP) - Wireless Transaction Protocol (WTP) - Wireless Markup Language (WML) - WML script
WAP 1.2	Nov. 1999	Additional features: - Push technology - User Agent Profile (UAProf) - Wireless Telephony Application (WTA) - Wireless Identity Module (WIM) - Public Key Infrastructure (PKI)
WAP 1.2.1	June 2000	Minor corrections
WAP 2.0	July 2001	Convergence with Internet technologies Additional features: - Support of MMS 1.0 (3GPP Release 99) - HTTP, TCP, persistent storage - XHTML, SyncML, client provisioning, etc.

Box 1.1 Markup languages for WAP-enabled devices

The HyperText Markup Language (HTML) is the content format commonly used in the World Wide Web. HTML enables a visual presentation of information (text, images, hyperlinks, etc.) on large screens of desktop computers. eXtensible Markup Language (XML) is another markup language that is generic enough to represent the basis for the definition of many other dedicated languages. Several markup languages supported by WAP-enabled devices are derived from XML. This is the case of WML and XHTML. WML has been optimized for rendering information on mobile devices with limited rendering capabilities. The eXtensible HyperText Markup Language (XHTML) is an XML reformulation of HTML. Both WML and XHTML are extensible since the formats allow the addition of new markup tags to meet changing needs.

1.6.2 WAP Architecture

Figure 1.9 shows the components of a generic WAP architecture. The WAP device can communicate with remote servers directly or via a number of intermediary proxies and

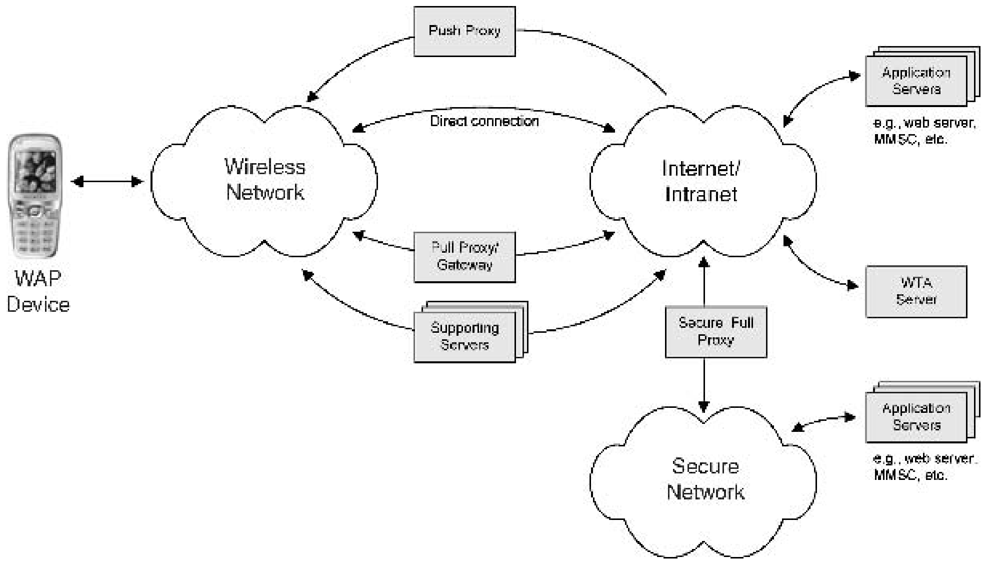


Figure 1.9 Generic WAP architecture

gateways. These proxies/gateways may belong to the mobile network operator or alternatively to service providers. The primary function of proxies/gateways is to optimize the transport of content from servers to WAP devices. Supporting servers, as defined by the WAP Forum, include Public Key Infrastructure (PKI) portals, content adaptation servers, and provisioning servers.

1.6.3 Push Technology

In a typical client/server model, a client retrieves the selected information from an application server by explicitly requesting the download of information from the server. This retrieval method is also known as the *pull technology* since the client pulls some data from a server. Web browsing is an example of models based on pull technology.

In contrast, another technology has been introduced in the WAP model and is known as the *push technology*. With push technology, a server is able to push some data to the WAP device with no prior explicit request from the client. In other words, the pull of information is always initiated by the client, whereas the push of information is always initiated by the server.

The push framework, defined by the WAP Forum in [WAP-250], is shown in Figure 1.10. In the push framework, the *push initiator* initiates the push transaction. The push initiator, usually an application server (e.g., web server, MMS center, etc.) transmits the content to be pushed along with XML-formatted delivery instructions to a *Push Proxy Gateway* (PPG). The PPG then delivers the push content to the WAP device according to the delivery instructions. The push initiator interacts with the PPG using the *Push Access Protocol* (PAP). On the other side, the PPG uses the *Push Over The Air* (OTA) protocol (based on WSP or HTTP) to deliver the push content to the WAP device.

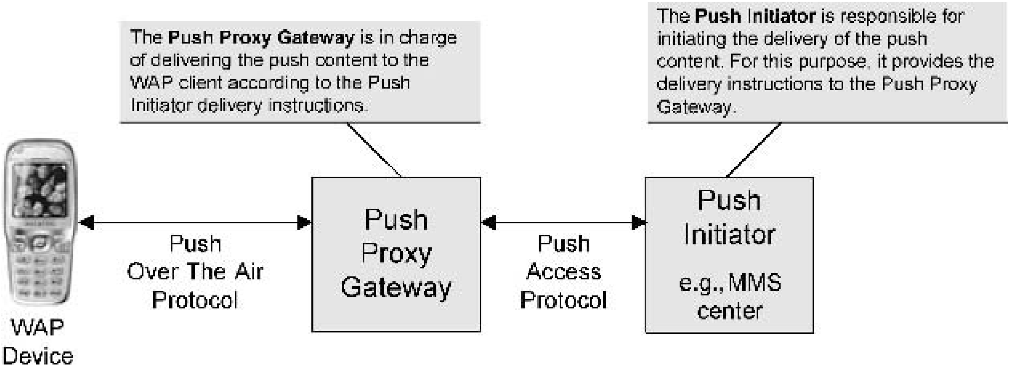


Figure 1.10 The push framework

The PPG may implement network-access-control policies indicating whether push initiators are allowed to push content to WAP devices. The PPG can send back a notification to the Push Initiator to indicate the status of a push request (delivered, canceled, expired, etc.).

Three types of browsing content can be pushed to a WAP microbrowser: Service Indication (SI), Service Loading (SL), and Cache Operation (CO). *Push SI* provides the ability to push content to users to notify them about electronic mail messages awaiting retrieval, news headlines, commercial offers, and so on. In its simplest form, a push SI contains a short message along with an URI. Upon receipt of the push SI, the message is presented to the user who is given the possibility of starting the service (retrieve the content) to which the URI refers. The subscriber may decide to start the service immediately or to postpone it. In contrast to push SI, *push SL* provides the ability to push some content to the WAP device without user explicit request. A push SL contains a URI that refers to the push content. Upon receipt of the push SL, the push content is automatically fetched by the WAP device and is presented to the user. *Push CO* provides a means for invalidating objects stored in the WAP device's cache memory.

In addition to browsing specific push contents, information can also be pushed to other WAP-based applications such as the WTA agent and the provisioning agent. The MMS client embedded in a WAP device also receives application-specific push messages to notify the user about the availability of new messages and for the delivery of reports.

1.6.4 User Agent Profile

The *User Agent Profile* (UAProf) specification was first published in the WAP 1.2 specification suite, improved in WAP 2.0, and further enhanced recently by the Open Mobile Alliance. The objective of this specification is to define a method for describing the capabilities of clients and the preferences of subscribers. In practice, this description (known as a user agent profile) is mainly used for adapting available content to the rendering capabilities of WAP devices. For this purpose, the user agent profile is formatted using a Resource Description Framework (RDF) schema in accordance with Composite Capability/

Preference Profiles (CC/PP). The CC/PP specification defines a high-level framework for exchanging and describing capability, and preference information using RDF. Both RDF and CC/PP specifications have been published by W3C. UAProf, as defined by the WAP Forum and updated by OMA in [OMA-UAProf] (version 2.0), allows the exchange of user agent profiles, also known as *Capability and Preference Information* (CPI), between the WAP device, intermediate network points, and the origin server (web server or MMS center). These intermediate network points and origin servers can use the CPI to tailor the content of WSP/HTTP responses to the capabilities of receiving WAP devices. The UAProf specification defines a set of *components* that WAP-enabled devices can convey within the CPI. Each component is itself composed of a set of attributes or *properties*. Alternatively, a component can contain a URI pointing to a document describing the capabilities of the client. Such a document is stored on a server known as a *profile repository* (usually managed by device manufacturers or by software companies developing WAP microbrowsers). UAProf is composed of the following components:

- *Hardware platform*: this component gathers a set of properties indicating the hardware capabilities of a device (screen size, etc.).
- *Software platform*: this component groups a set of properties indicating the software capabilities of a device (operating system, supported image formats, etc.).
- *Browser user agent*: this component gathers properties characterizing the Internet browser capabilities.
- *Network characteristics*: this component informs on network and environment characteristics such as the bearer capacity.
- *WAP characteristics*: this component advertises WAP browsing capabilities of the device. This includes information on the configuration of the WML browser and so on.
- *Push characteristics*: this component indicates push capabilities of the device. This includes the set of supported content types, the maximum message size that can be handled, and whether or not the device can buffer push messages.
- *MMS characteristics*: this component describes the device capabilities for retrieving and rendering multimedia messages (MMS version, maximum message size, supported content types, etc.).

For a configuration involving a WAP device and a gateway communicating with WSP, RDF descriptions can be encoded in binary with the WAP Binary XML (WBXML). In this context, the CPI is provided by the WAP device as part of the WSP session establishment request. The WAP device can also update its CPI at any time during an active WSP session. Note that the WAP gateway may also override a CPI provided by a device.

The use of UAProf in the context of MMS is further explained in Section 5.21.

1.6.5 WAP 1.x Legacy Configuration

With the objective of fulfilling the requirements of various services in heterogeneous mobile networks, several network configurations can coexist in the WAP environment. This section and the two following ones present the three most common configurations of the WAP environment: WAP 1.x legacy configuration, WAP HTTP proxy with wireless profiled TCP and HTTP, and HTTP with direct access.

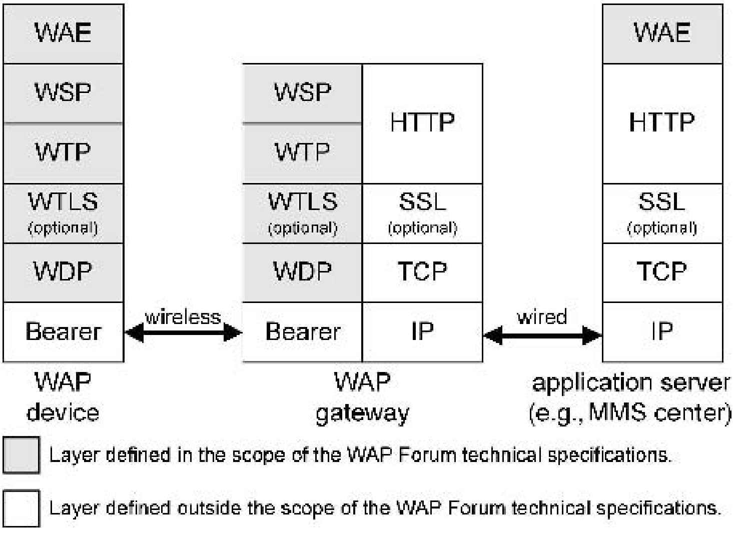


Figure 1.11 WAP 1.x legacy configuration with WAP gateway

Figure 1.11 shows the protocol stack of the configuration defined in the WAP specification suite 1.x. This configuration is also supported by the WAP specification suite 2.0 in addition to other configurations. In this configuration, the WAP device communicates with a remote server via an intermediary *WAP gateway*. The primary function of the WAP gateway is to optimize the transport of content between the remote server and the WAP device. For this purpose, the content delivered by the remote server is converted into a compact binary form by the WAP gateway prior to the transfer over the wireless link. The WAP gateway converts commands conveyed between datagram-based protocols (WSP, WTP, WTLS, and WDP) and protocols commonly used on the Internet (HTTP, SSL, and TCP).

The *Wireless Application Environment* (WAE) is a general-purpose application environment in which operators and service providers can build applications (e.g., MMS client or MMS center) for a wide variety of wireless platforms.

The *Wireless Session Protocol* (WSP) provides features also available in HTTP (requests and corresponding responses). Additionally, WSP supports long-lived sessions and the possibility to suspend and resume previously established sessions. WSP requests and corresponding responses are encoded in a binary form for transport efficiency.

The *Wireless Transaction Protocol* (WTP) is a lightweight transaction-oriented protocol. WTP improves the reliability over underlying datagram services by ensuring the acknowledgment and retransmission of datagrams. WTP has no explicit connection set-up or connection release. Being a message-oriented protocol, WTP is appropriate for implementing mobile services such as browsing. Optionally, Segmentation And Reassembly (SAR) of packets composing a WTP protocol unit can be supported as described in Section 1.6.8.

The optional *Wireless Transport Layer Security* (WTLS) provides privacy, data integrity, and authentication between applications communicating with the WAP technology. This includes the support of a secure transport service. WTLS provides operations for the establishment and the release of secure connections.

The *Wireless Data Protocol* (WDP) is a general datagram service based on underlying low-level bearers. WDP offers a level of service equivalent to the one offered by the Internet's User Datagram Protocol (UDP).

At the bearer level, the connection may be a circuit-switched connection (as found in GSM networks) or a packet-switched connection (as found in GPRS and UMTS networks). Alternatively, the transport of data at the bearer level may be performed over the Short Message Service (e.g., for push messages) or over the Cell Broadcast Service.

1.6.6 WAP HTTP Proxy with Wireless Profiled TCP and HTTP

Figure 1.12 shows a configuration in which the WAP device communicates with application servers via an intermediary WAP proxy. The primary role of the proxy is to optimize the transport of content between the fixed Internet and the mobile network. It also acts as a Domain Name Server (DNS) for mobile devices. With this configuration, Internet protocols are preferred against legacy WAP protocols. This is motivated by the need to support IP-based protocols in an end-to-end fashion, from the application server back to the WAP device. The protocol stack of this configuration, defined in the WAP specification suite 2.0, is shown in Figure 1.12.

The *Wireless Profiled HTTP* (WP-HTTP) is an HTTP profile specifically designed for coping with the limitations of wireless environments. This profile is fully interoperable with HTTP/1.1 and supports message compression.

The optional *Transport Layer Security* (TLS) ensures the secure transfer of content for WAP devices involved in the exchange of confidential information.

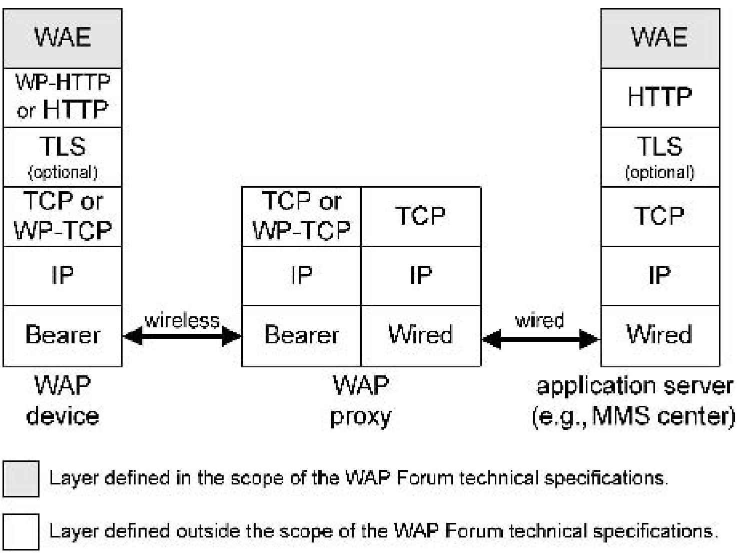


Figure 1.12 Configuration with WAP proxy

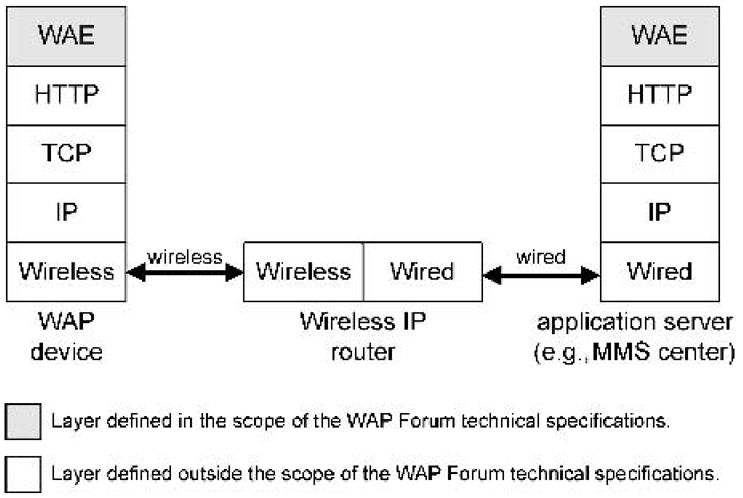


Figure 1.13 WAP configuration with direct access

The *Wireless Profiled TCP* (WP-TCP) offers a connection-oriented service. It is adapted to the limitations of wireless environments but remains interoperable with existing Transmission Control Protocol (TCP) implementations.

1.6.7 HTTP with Direct Access

Figure 1.13 shows a configuration where the WAP device is directly connected to the application server (via a wireless router that provides a bearer-level connection). The protocol stack shown in this configuration is defined in the WAP specification suite 2.0. A WAP device, compliant with the WAP 2.0 specification suite, may support all configurations by supporting WAP 1.x and WAP 2.0 protocol stacks.

1.6.8 WTP Segmentation and Reassembly

In the WAP 1.x legacy configuration, an optional *Segmentation And Reassembly* (SAR) mechanism [WAP-224] allows large transactions to be segmented at the WTP level by the sender and reassembled by the receiver. SAR is specifically used when the size of a transaction (e.g., retrieval of a 50-KB multimedia message) exceeds the WTP Maximum Transmission Unit (MTU). In the context of MMS, SAR is used for transactions including the sending and retrieval of large messages. Note that, in the WAP 1.x configuration, SAR is optional and if it is not supported at the WTP level, then segmentation and reassembly may be supported at an underlying layer (e.g., [RFC-791] for IP, [3GPP-23.040] for SMS, etc.).

With SAR, the WTP transaction is segmented into several packets and packets can be sent by the sender in the form of packet groups. For efficiency, the receiver acknowledges the reception of each single packet group and the sender does not start transmitting packets of a new group if the previous group has not been properly acknowledged by the receiver. A

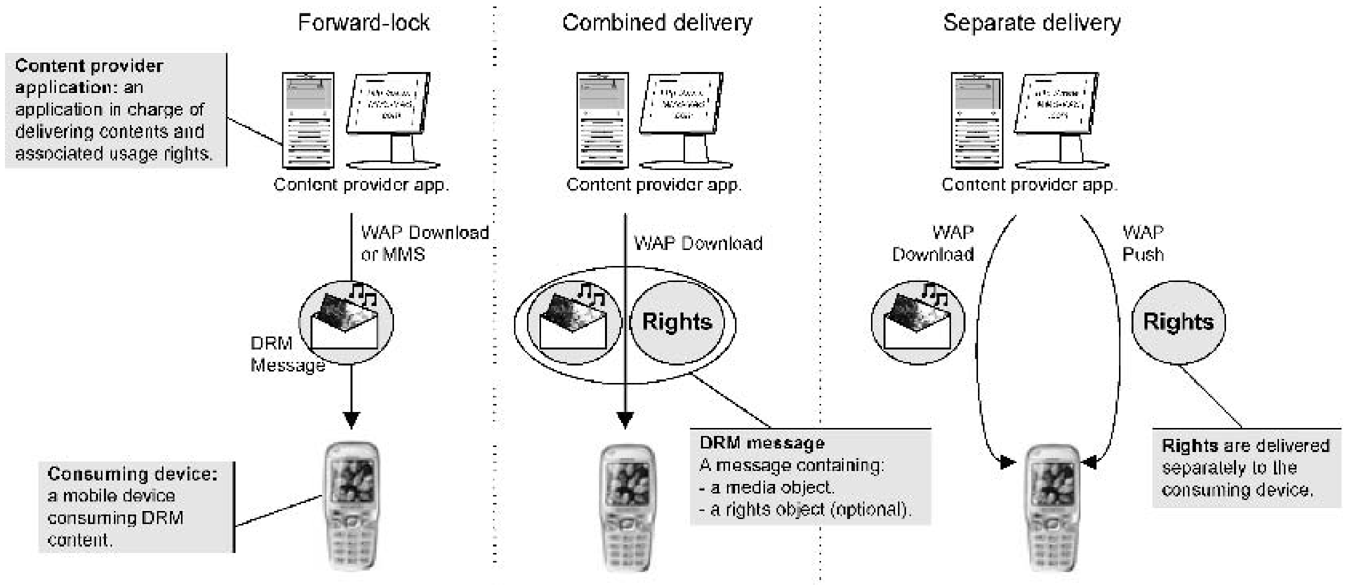


Figure 1.14 OMA digital rights management

group can contain a maximum number of 256 packets. The sender determines the number of packets in a group, preferably according to the characteristics of the network and the ones of the device. The first packet group is sent without knowing the characteristics of the receiver. Therefore, the size of the first packet group should not be too large. SAR allows a selective retransmission of multiple lost packets for a given group. This feature minimizes the number of packets sent over WTP.

1.6.9 OMA Digital Rights Management

At the end of 2002, OMA published technical specifications [OMA-DRM] for mechanisms representing the basis for the management of digital rights associated with media objects downloaded via WAP download or MMS. *Digital Rights Management* (DRM) provides a means, for operators and content providers, to control the usage of media objects once they have been downloaded to a mobile device (also known as a ‘consuming device’ in the DRM context). DRM enables content providers to define usage rules specifying the user’s rights regarding the usage of the corresponding media object. For instance, a content provider can grant a user the rights to preview for free and charge for more sophisticated usages. Three main mechanisms are defined in OMA-DRM as shown in Figure 1.14. They differ in the way rights are communicated to the consuming device and are as follows:

- *Combined delivery* consists of delivering the media object along with the associated rights in a single DRM message.
- *Forward lock* is the simplest of the OMA-DRM mechanisms. This is a special case of the combined delivery mechanism in which the DRM message contains only the media object, without the associated rights. For forward lock, the following set of rights applies: the user is not allowed to forward or modify the media object.
- With *separate delivery*, the media object and corresponding rights are conveyed to the consuming device over separate distribution channels. In this context, the media object is converted into a *DRM Content Format* (DCF) [OMA-DRM-CF]. This conversion consists of a symmetric encryption of the original media object, making the converted object unusable, unless the consuming device has the necessary Content Encryption Key (CEK) to convert the object back to its original form. The CEK along with the rights is delivered to the consuming device separately from the associated media object, typically over WAP push.

OMA DRM forward lock is of particular interest to the content-to-person scenario of MMS and is applicable from MMS 1.2, and the support of combined and separate deliveries has also been introduced in MMS 1.3.

The application of OMA DRM in the context of MMS is further explained in Section 5.31.

2

Standardization

Standardization of telecommunications technologies and associated service enablers is of key importance for the development of communicating systems in a multi-vendor environment. Many parties such as operators, manufacturers, third party application developers, and sometimes regulators collaborate in the scope of standardization activities to produce technical specifications that are widely endorsed for the development of commercial solutions. SMS, EMS, and MMS are three mobile messaging services for which underlying technologies have been subject to significant standardization activities. Standardization does not mean designing from scratch all technologies required for enabling interoperable services. Instead, standardization means identifying the most appropriate elements in the basket of existing technologies in order to allow a rapid roll-out of the service, creating new technologies only when no appropriate solution exists.

Compared with other mobile messaging services such as SMS and EMS, the standardization picture for MMS has become very complex. Several standardization organizations have collaborated in order to produce stable technical specifications for MMS in a timely manner. Organizations that have been actively involved in the design of MMS standards are 3GPP and the WAP Forum. Since 2002, the WAP Forum has merged with other standardization bodies to form the Open Mobile Alliance (OMA). Consequently, MMS activities of the WAP Forum have now been fully transferred to OMA. Most MMS standards produced by these organizations partially rely on existing technologies developed by bodies such as W3C and IETF. The GSM Association (GSMA) is a group mainly composed of network operators and publishes valuable recommendations for the design of interoperable services. 3GPP2 publishes various technical specifications, including specifications for MMS, specifically for the deployment of the service in several Asian and North American markets.

For any engineer involved in designing solutions based on SMS, EMS, or MMS, it becomes essential to acquire a basic understanding on how standardization bodies proceed to produce standards. Most importantly, engineers need to identify dependencies linking messaging standards among themselves and understand how standards get created, reach a

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2

mature stage, and evolve over time. For this purpose, this chapter introduces the working procedures of organizations outlined below and provides an insight of their organizational structure in terms of working groups. In addition, rules for numbering/referencing messaging standards are explained and illustrated with examples.

- *Third Generation Partnership Project* (3GPP): 3GPP is not a standardization development organization in its own right but rather a joint project between several regional standardization bodies from Europe, North America, Korea, Japan, and China. The prime objective of 3GPP is to develop UMTS technical specifications. It is also responsible for maintaining existing GSM specifications and developing further GSM extensions (e.g., GPRS). This encompasses the development of widely accepted technologies and service capabilities. 3GPP is strongly involved in the development of messaging standards (general service requirements, architecture, formats and codecs, and several low level technical realizations).
- *Third Generation Partnership Project 2* (3GPP2): 3GPP2 is another standardization partnership project established out of the International Telecommunication Union's (ITU) International Mobile Telecommunications "IMT-2000" initiative. The role of 3GPP2 is to produce specifications for services deployed in several North American and Asian markets with focus on next generation CDMA networks. In the scope of this project, 3GPP2 looks at refining requirements for MMS and designing alternative realizations of interfaces defined in 3GPP and OMA standards.
- *GSM Association* (GSMA): GSMA is a global trade organization that represents the interest of several hundreds of GSM mobile operators. The association role consists of promoting, protecting, and enhancing the interests of GSM operators. For this purpose, GSMA publishes a number of technical recommendations that are widely endorsed by the GSM community.
- *Internet Engineering Task Force* (IETF): IETF is a large community of academic and industrial contributors that defines the protocols primarily used on the Internet. Messaging services in the mobile world also rely on several IETF protocols.
- *World Wide Web Consortium* (W3C): W3C is a standardization body that concentrates on the development of protocols and formats to be used in the World Wide Web. Well-known formats and protocols published by W3C are the Hypertext Transfer Protocol (HTTP) and the eXtensible Markup Language (XML). Mobile messaging services also rely on proven W3C formats.
- *WAP Forum*: the Wireless Application Protocol (WAP) Forum was a joint project for the definition of WAP technical specifications. This encompassed the definition of a framework for the development of applications to be executed in various wireless platforms. The WAP Forum produced the initial MMS specifications for the support of MMS in the WAP environment. The WAP Forum does not exist anymore and its activities have been transferred to the Open Mobile Alliance.
- *Open Mobile Alliance* (OMA): OMA is a standardization forum established in June 2002. Activities of several existing standardization bodies including the ones of the WAP Forum (MMS and others) have been transferred to OMA. OMA is therefore actively involved in maintaining MMS standards designed by the WAP Forum and producing new standards for next generations of MMS devices.

2.1 Messaging Road Map

The road map of messaging technologies and services is becoming more and more complex. This complexity is mainly due to the fact that services rely on technologies developed by a large number of standardization development organizations.

Figure 2.1 shows the relationships between the introduction of network technologies (GSM, GPRS, and UMTS) and the commercial availability of messaging services. The figure also shows the major milestones for three standardization development organizations: 3GPP, OMA, and WAP Forum.

Several messaging technologies have been developed to meet specific market requirements. One of the first messaging systems to have been introduced in mobile networks is the Short Message Service (SMS). In its simplest form, SMS allows subscribers to exchange short messages containing text only. SMS was first introduced as a basic service of GSM and has been the subject of many extensions. Initial standardization work on SMS was carried out within the scope of the European Telecommunications Standards Institute (ETSI) standardization process until the transfer of responsibility to 3GPP. Standardization work for SMS is now carried out in the scope of the 3GPP standardization process. One of the significant evolutions of SMS is an application-level extension called the *Enhanced Messaging Service* (EMS). EMS allows subscribers to exchange long messages containing text, melodies, pictures, animations, and various other objects. Two versions of EMS are available and are covered in this book under the terms *basic EMS* and *extended EMS*.

Since 1998, standardization bodies have concentrated on the development of a new messaging service called the Multimedia Messaging Service (MMS). MMS enables subscribers to exchange multimedia messages. The standardization of MMS has reached a mature stage and the penetration of MMS devices in the market is growing rapidly. Standardization aspects of MMS are further elaborated in the following section.

2.2 MMS Standards

MMS is a sophisticated multimedia messaging service and has required a tremendous standardization workload. Several standardization bodies have therefore collaborated in order to produce the technical specifications to allow the introduction of MMS-capable devices on the market at the most appropriate time. In this configuration, 3GPP has taken the lead in identifying the high-level service requirements, designing the MMS architecture, producing several low level technical realizations, and identifying appropriate codecs/formats and streaming protocols. On the other hand, the WAP Forum initially took the responsibility of defining the low level technical realizations of the interface bridging the MMS phone and the network in the WAP environment. Additionally, a group of telecommunications vendors, known as the MMS-IOP group, also produced specifications (the MMS conformance document) to guarantee the interoperability between first MMS devices. In 2002, MMS activities of the WAP Forum and the MMS-IOP group were merged into OMA to allow a more efficient standardization development process for MMS. Of course, 3GPP and the WAP Forum/OMA did not produce all MMS specifications from scratch and did manage to build up MMS standards on the basis of existing proven standards such as the ones produced by W3C and IETF, developing new technologies only when not available elsewhere.

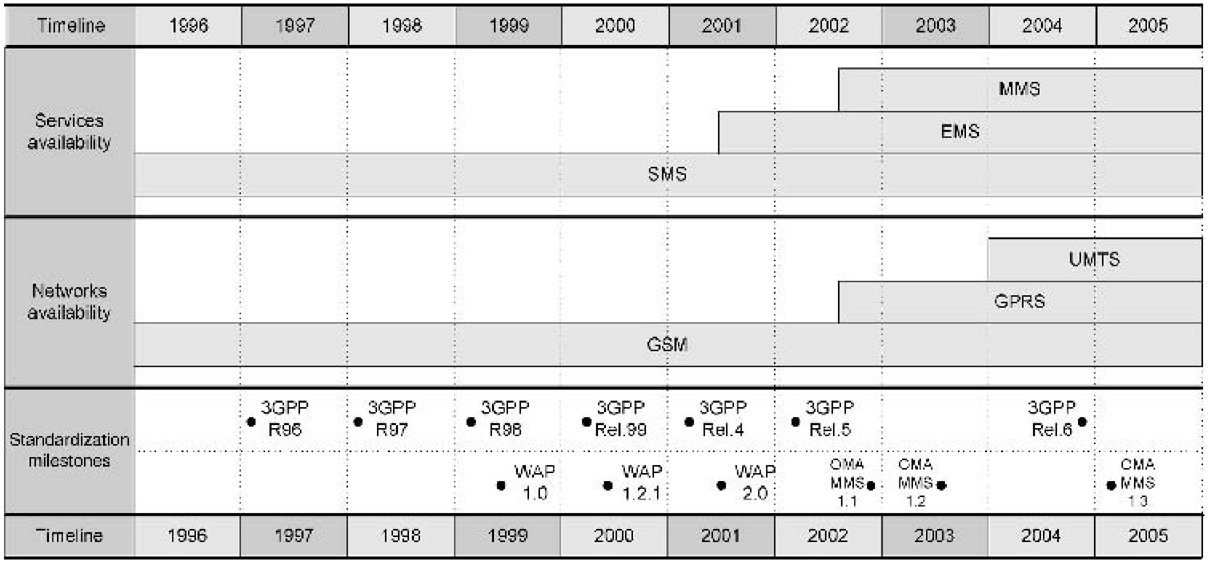


Figure 2.1 Relationships between availability of messaging services and technologies

2.3 Third Generation Partnership Project

The European Telecommunications Standard Institute (ETSI) and the Conférence Européenne des Postes et Télécommunications (CEPT) have carried out work on the GSM standards for a period of almost 18 years. Within the scope of the ETSI standardization organization, the work was carried out by the Special Mobile Group (SMG) technical committee. In 2000, the committee agreed to transfer the responsibility of the development and maintenance of the GSM standards to the Third Generation Partnership Project (3GPP). 3GPP was set in 1998 by five standard development organizations (including ETSI) with the objective of collaborating on the development of interoperable mobile systems (a sixth organization joined the partnership later). The six partner organizations represent telecommunications companies from five different parts of the world:

- European Telecommunications Standards Institute (ETSI) for Europe
- Committee T1 for the United States
- Association of Radio Industries and Businesses (ARIB) for Japan
- Telecommunications Technology Committee (TTC) for Japan
- Telecommunications Technology Association (TTA) for Korea
- China Wireless Telecommunication Standard (CWTS) for China.

Each individual member of one of the six partners can contribute to the development of 3GPP specifications. In order to define timely services and technologies, individual members are helped by several market representative partners. At the time of writing this book, 3GPP market representatives are the UMTS Forum,[1] the Global mobile Suppliers Association (GSA),[2] the GSM Association (GSMA),[3] the IPv6 Forum,[4] the 3G.IP focus group,[5] and the 3G Americas.[6] The responsibility of these market representative partners consists of identifying requirements for services. In this process, the six partner organizations take the role of publishers of 3GPP specifications.

2.3.1 3GPP Structure

The 3GPP standardization process strictly defines how partners should coordinate the standardization work and how individual members should participate in the development of specifications. There is a clear separation between the coordination work of 3GPP partners and the development of specifications by individual members. This separation enables a very efficient and robust standardization process. In order to achieve it, the 3GPP structure is split into the *Project Coordination Group* (PCG) and five *Technical Specifications Groups* (TSGs). PCG is responsible for managing and supervising the overall work carried out within the scope of 3GPP whereas TSGs create and maintain 3GPP specifications. PCG and

[1] http://www.umts-forum.org/
[2] http://www.gsacom.com/
[3] http://www.gsmworld.com/
[4] http://www.ipv6forum.com/
[5] http://www.3gip.org/
[6] http://www.3gamericas.org/

TSGs endeavor to reach consensus on all issues. However, decisions in PCG and TSGs can be made by vote if consensus cannot be reached. In each TSG, several Working Groups (WGs) create and manage specifications for a set of related technical topics (for instance CN WG5 deals with the set of technical topics related to the Open Service Architecture). If the set of technical topics is too broad, then a WG may be further split into Sub-Working Groups (SWGs). This is the case for T WG2 (or also T2 for short) which deals with mobile terminal services and capabilities. T2 is split into the following three SWGs:

- T2 SWG1 deals with the Mobile Execution Environment (MExE);
- T2 SWG2 deals with user equipment capabilities and interfaces;
- T2 SWG3 deals with messaging aspects. Activities of sub-working group T2 SWG3 encompass the development of messaging services and technologies including SMS, EMS, Cell Broadcast Service, and MMS.

Figure 2.2 shows the list of 3GPP TSGs and corresponding WGs. Please note that all TSGs are responsible for their own work items and specifications. However, TSG SA being

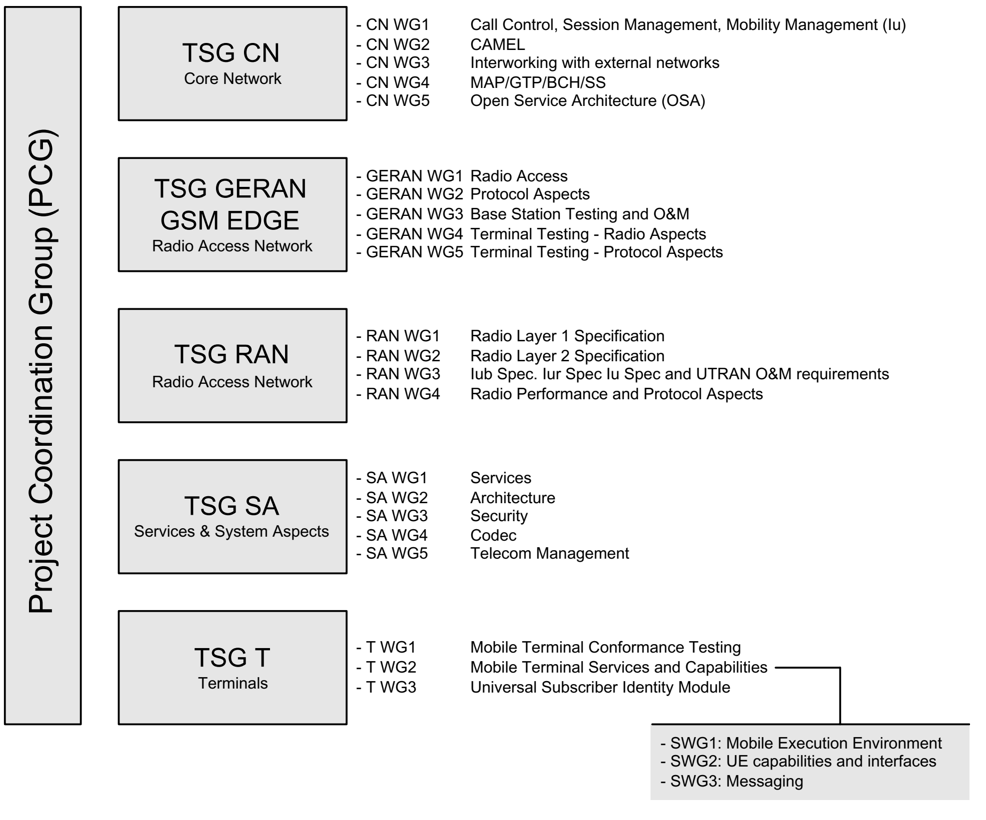


Figure 2.2 3GPP structure

responsible for the overall architecture and service capabilities has an additional responsibility for cross TSG coordination.

2.3.2 3GPP Specifications: Release, Phase, and Stage

Documents produced by the 3GPP are known as specifications. Specifications are either Technical Specifications (TS) or Technical Reports (TR). *Technical Specifications* define a GSM/UMTS standard and are published independently by the six partners (ETSI, Committee T1, ARIB, TTC, TTA, and CWTS). *Technical Reports* are, for example, feasibility studies for new features/services and sometimes become technical specifications later. In order to fulfill ever-changing market requirements, 3GPP specifications are regularly extended with new features. To ensure that market players have access to a stable platform for implementation while allowing the addition of new features, the development of 3GPP specifications is based on a concept of parallel releases. In this process, specifications are regularly frozen. Only essential corrections are permitted for a frozen specification. Standardization work can still be carried out but will be incorporated in the next release of the same specification. An engineer implementing a commercial solution based on one or more 3GPP standards should, as much as possible, base the work on frozen specifications. An unfrozen specification is subject to change and should never be considered as a stable platform on which to build a commercial solution. In 3GPP, technical specifications are typically frozen in intervals of 1–1.5 years. Consequently, releases used to be named according to the expected specification freezing date (Release 98, Release 99, etc.). In 1999, 3GPP decided that releases produced after 1999 would no longer be named according to a year but according to a unique sequential number (Release 5 followed Release 4 which itself followed Release 99). This decision was made to get more flexibility in adjusting the timing of releases to market needs instead of having always one release per year.

Each 3GPP technical specification is usually categorized into one of three possible stages. The concept of characterizing telecommunication services into three stages was first introduced by ITU in [ITU-I.130]. A *stage 1* specification provides a service description from a service-user's perspective. A *stage 2* specification describes a logical analysis of the problem to be solved, a functional architecture, and associated information flows. A *stage 3* specification describes a concrete implementation of the protocols between physical elements onto the elements of the stage 2 functional architecture. A stage 3 implementation is also known as a technical realization. Note that several technical realizations may derive from a common stage 2 specification.

2.3.3 3GPP Specifications: Numbering Scheme

Each 3GPP technical document (report or specification) is uniquely identified by a reference as shown in Figure 2.3. The reference starts with the prefix "3GPP" and is followed by two letters identifying the document type ("TS" for a specification and "TR" for a report). After the document type, follows a specification number which can take one of the following forms: *aa.bbb* or *aa.bb*. In the specification number, *aa* indicates the document intended use as shown in Table 2.1. In the document reference, the document number is followed by a version number in the format *Vx.y.z*. In this format, *x* represents the release, *y* represents the technical version, and *z* represents the editorial version. Table 2.2 shows how the document

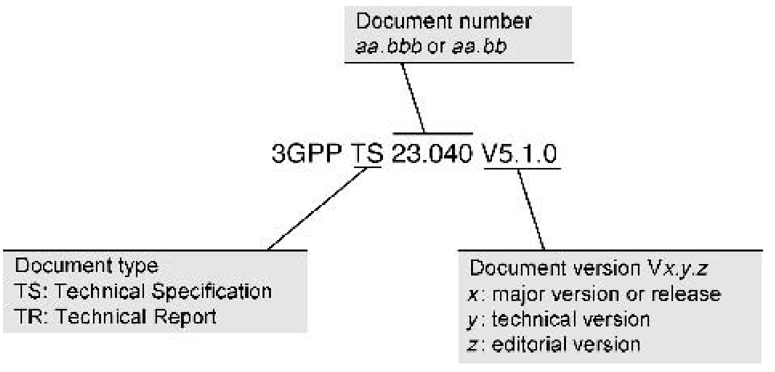


Figure 2.3 3GPP specification type, number, and version

version is formatted according to its associated release. The freezing date for each release is also indicated.

In addition to its reference, a title is also provided for a 3GPP document. For instance, the following document contains the definition of MMS stage 1:

Reference: `3GPP TS 22.140 V5.2.0`
Title: `Multimedia Messaging Service, Stage 1`

Lists of available 3GPP specifications are provided in the documents listed in Table 2.3. 3GPP specifications can be downloaded from the 3GPP website at http://www.3gpp.org.

Table 2.1 3GPP specifications/numbering scheme

Range for GSM up to and including Release 99	Range for GSM Release 4 onwards	Range for UMTS Release 1999 onwards	Type of use
01.*bb*	41.*bbb*	21.*bbb*	Requirement specifications
02.*bb*	42.*bbb*	22.*bbb*	Service aspects
03.*bb*	43.*bbb*	23.*bbb*	Technical realizations
04.*bb*	44.*bbb*	24.*bbb*	Signaling protocols
05.*bb*	45.*bbb*	25.*bbb*	Radio access aspects
06.*bb*	46.*bbb*	26.*bbb*	Codecs
07.*bb*	47.*bbb*	27.*bbb*	Data
08.*bb*	48.*bbb*	28.*bbb*	Signaling protocols
09.*bb*	49.*bbb*	29.*bbb*	Core network signaling protocols
10.*bb*	50.*bbb*	30.*bbb*	Programme management
11.*bb*	51.*bbb*	31.*bbb*	SIM/USIM
12.*bb*	52.*bbb*	32.*bbb*	Charging and OAM&P
13.*bb*			Regulatory test specifications
		33.*bbb*	Security aspects
		34.*bbb*	Test specifications
		35.*bbb*	Algorithms

Table 2.2 3GPP specifications/releases

GSM/EDGE release	3G release	Abbreviated name	Specification number format	Specification version format	Freeze date
Phase 2 + Release 6	Release 6	Rel-6	*aaa.bb* (3G)	6.*x.y* (3G)	December 2004
Phase 2 + Release 5	Release 5	Rel-5	*aa.bb* (GSM)	5.*x.y* (GSM)	March 2002
Phase 2 + Release 4	Release 4	Rel-4	*aaa.bb* (3G) *aa.bb* (GSM)	4.*x.y* (3G) 9.*x.y* (GSM)	March 2001
Phase 2 + Release 99	Release 99	R99	*aaa.bb* (3G) *aa.bb* (GSM)	3.*x.y* (3G) 8.*x.y* (GSM)	March 2000
Phase 2 + Release 98		R98	*aa.bb*	7.*x.y*	Early 1999
Phase 2 + Release 97		R97	*aa.bb*	6.*x.y*	Early 1998
Phase 2 + Release 96		R96	*aa.bb*	5.*x.y*	Early 1997
Phase 2		PH2	*aa.bb*	4.*x.y*	1995
Phase 1		PH1	*aa.bb*	3.*x.y*	1992

Table 2.3 Specifications listing the GSM/UMTS specifications produced by 3GPP

Release	List of GSM specifications	List of UMTS specifications
Release 99	[3GPP-01.01]	[3GPP-21.101]
Release 4	[3GPP-41.102]	[3GPP-21.102]
Release 5	[3GPP-41.103]	[3GPP-21.103]

2.4 Third Generation Partnership Project 2

The Third Generation Partnership Project 2 (3GPP2) is another standardization partnership project established out of the International Telecommunication Union's (ITU) International Mobile Telecommunications "IMT-2000" initiative. The role of 3GPP2 is to produce specifications for industrial players from North American and Asian markets with focus on next generation CDMA networks. In the scope of this project, 3GPP2 looks at refining requirements for MMS and designing alternative realizations of interfaces defined in 3GPP and OMA standards.

3GPP2 specifications can be downloaded from the 3GPP2 website at http://www.3gpp2.org.

2.5 GSM Association

Founded in 1987, the GSM Association (GSMA) is a global trade organization that represents the interest of several hundreds of GSM mobile operators. The association role consists of promoting, protecting, and enhancing the interests of GSM operators. For this purpose, GSMA publishes a number of commercial and technical recommendations that are widely endorsed by the GSM community. GSMA is composed of working groups and regional groups. In addition, many other specialist interest groups look at very specific services or technologies or look at regional requirements.

2.5.1 Working Groups

At the time of writing, there are eight working groups within GSMA. These groups meet on a regular basis to provide guidance for commercial and technical aspects relevant to the interests of the GSM industry. The scopes of working group activities are outlined below:

- *Billing and Accounting Roaming Group* (BARG): BARG supports international roaming through the definition of charging principles together with the related interoperator procedures, billing harmonization, credit control, and liaison with other groups regarding fraud control.
- *Interworking Roaming Expert Group* (IREG): IREG focuses on issues related to signaling and roaming aspects.
- *Security Group* (SG): SG develops algorithms and protocols to secure GSM networks.
- *Fraud Forum* (FF): FF identifies techniques used to perpetrate fraud against GSM operators. It also recommends solutions to cope with these techniques.
- *Service* (SerG): SerG gathers service requirements from GSM operators. These requirements relate mainly to billing and customer care. The working group also provides a connection between operator marketing requirements and corresponding technical realizations.
- *Terminal Working Group* (TWG): TWG focuses on issues related to the use of GSM mobile stations.
- *Environmental Working Group* (EWG): EWG provides information, advice, and strategy on the issues of potential interference and health effects in relation to the use of GSM equipment and services.
- *Transferred Account Data Interchange Group* (TADIG): TADIG defines data interchange procedures. It focuses on billing aspects.

2.5.2 Regional Groups

GSMA is also composed of nine regional interest groups. These groups represent the interests of GSM operators in the following regions: Europe, Asia Pacific, Arab world, Russia, North America, Latin America, Africa, Central Asia, and India.
More information is available from the GSMA website at http://www.gsmworld.com.

2.6 Internet Engineering Task Force

The Internet Engineering Task Force (IETF) produces technical specifications related to Internet protocols and procedures in the scope of the Internet Standards Process. It is an international forum open to any interested party. In IETF, the technical work is carried out by working groups, each one focusing on specific technical topics (routing, transport, security, etc.). Furthermore, working groups are categorized into areas managed by area directors. Area directors are members of the Internet Engineering Steering Group (IESG). In IETF, the Internet Architecture Board (IAB) provides an architectural oversight of the Internet. Both the IESG and the IAB are chartered by the Internet Society (ISOC). In addition, the Internet Assigned Numbers Authority (IANA) has the responsibility for assigning unique numbers for Internet protocols (application ports, content types, etc.).

2.6.1 IETF Documents

IETF communications are primarily performed through the publication of a series of documents known as *Request For Comments* (RFCs). First RFCs on Internet networking were produced in 1969 as part of the original ARPA wide-area NETworking (ARPANET) project. RFCs are numbered in chronological order of creation. An RFC, documenting an Internet standard, is given an additional reference in the form STD*xxx* and becomes part of the STD subseries of the RFC series. RFCs are classified into the following five high level categories:

1. Standard track (including proposed standards, draft standards, and Internet standards);
2. Best current practices;
3. Informational RFCs;
4. Experimental RFCs;
5. Historic RFCs.

For instance, the following specification is published as RFC822. The specification is an Internet standard also known as STD0011:

```
RFC822  Standard for the format of ARPA Internet text messages.
        D.Crocker. Aug-13-1982/Status: STANDARD/STD0011
```

During the development of a specification, draft versions of the document are often made available for informal review and comments. These temporary documents are known as *Internet drafts*. Internet drafts are working documents subject to changes without notice. Consequently, Internet drafts should not be considered as stable specifications on which to base developments for commercial solutions.

2.6.2 Internet Standard Track

Specifications subject to the Internet Standards Process fall into two categories: technical specifications and applicability statements. A *technical specification* is a description of a protocol, service, procedure, convention, or format. On the other hand, an *applicability statement* indicates how one or more technical specifications may be applied to support a particular Internet capability. Specifications that are to become Internet standards evolve over a scale of maturity levels known as the *standard track*. The standard track is composed of the following three maturity levels:

- *Proposed standard*: this is the specification at the entry level. The IESG is responsible for registering an existing specification as a proposed standard. A proposed standard is a specification that has reached a stable stage and has already been the subject of public review. However, implementers should consider proposed standards as immature specifications.
- *Draft standard*: a proposed standard can be moved to the draft standard of the standard track if at least two implementations based on the specification exist and interoperate well.

Since draft standards are considered as almost final specifications, implementers can safely design services on the basis of draft standards.
- *Internet standard*: an Internet standard (also referred to as a standard) is a specification that is the basis of many significant implementations. An Internet standard has reached a high level of technical maturity.

RFCs can be downloaded from the IETF website at http://www.ietf.org.

2.7 World Wide Web Consortium

The World Wide Web Consortium (W3C) is a standardization body, created in 1994, involved in the development of widely accepted protocols and formats for the World Wide Web. Technical specifications published by W3C are known as recommendations. W3C collaborates closely with IETF. W3C activities are organized into groups: *working groups* (for technical developments), *interest groups* (for more general work), and *coordination groups* (for communications among related groups). W3C groups are organized into the following five domains:

- The *architecture domain* includes activities related to the development of technologies which represent the basis of the World Wide Web architecture.
- The *document formats domain* covers all activities related to the definition of formats and languages.
- The *interaction domain* includes activities related to the improvements of user interactions with the World Wide Web. This includes the authoring of content for the World Wide Web.
- The *technology and society domain* covers activities related to the resolution of social and legal issues along with handling public policy concerns.
- The *web accessibility initiative* aims at promoting a high degree of usability for disabled people. Work is carried out in five primary areas: technology, guidelines, tools, education and outreach, and research and development.

To date, significant W3C contributions include the framework of the initial World Wide Web (based on HTML, URIs, and HTTP), XML, XHTML, SVG, and SMIL. W3C follows a dedicated recommendation track to initiate discussions on proposed technologies and to ultimately publish recommendations. The recommendation track defines four maturity levels for technical specifications. These maturity levels are depicted in Figure 2.4.

A *working draft* is the initial status for a technical specification in the W3C recommendation track. It is a work item which is being or will be discussed within the relevant W3C working group. However, the status ‘‘working draft’’ for a technical specification does not imply that there is a consensus between W3C members on the acceptability of the proposed technology.

A *last call working draft* is a special working draft that is regarded by the relevant working group as fulfilling the requirements of its charter. It has the status ‘‘last call working draft’’ when the relevant working group seeks technical review from other W3C groups, W3C members, and the public.

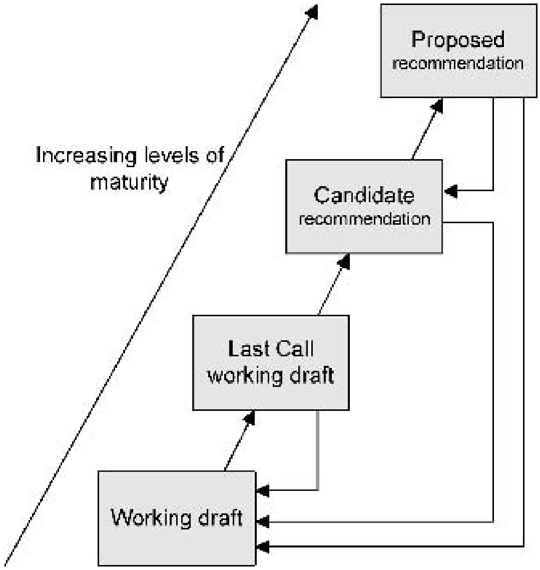


Figure 2.4 W3C recommendation track

A *candidate recommendation* is a technical specification that is believed to fulfill the relevant working group's charter, and has been published in order to gather implementation experience and feedback.

Finally, a *proposed recommendation* is a technical specification which has reached the highest level of maturity in the W3C recommendation track. It obviously fulfills the requirements of the relevant working group's charter but has also benefited from sufficient implementation experience and has been carefully reviewed. Only a proposed W3C recommendation can be considered as a stable technical specification on which to base the development of commercial solutions.

W3C technical specifications can be retrieved from http://www.w3c.org.

2.8 WAP Forum

Prior to its integration into OMA, the WAP Forum concentrated on the definition of a generic platform for the development of applications for various wireless technologies. The WAP Forum managed the following four types of technical documents:

- *Specification*: a specification contains technical or procedural information. At any given time, a specification is associated with a stage such as proposal, draft, etc. This stage indicates the level of maturity of the specification content.
- *Change Request* (CR): an unofficial proposal to change a specification. A change request was proposed by one or more individuals for discussion between WAP Forum members.
- *Specification Change Document* (SCD): an SCD is the draft of a proposed modification of a specification. An SCD could only be produced by the specification working group

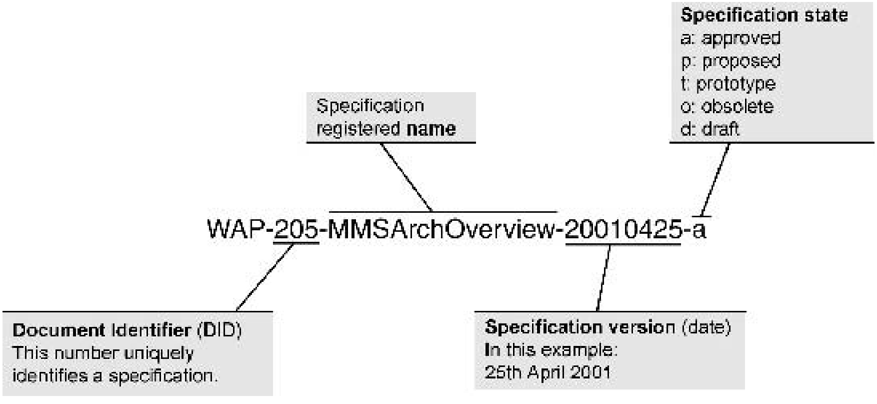


Figure 2.5 WAP Forum specification numbering

responsible for the corresponding specification. An SCD applies to a specific version of a specification.

- *Specification Implementation Note* (SIN): a SIN is an approved modification of a previously published specification. SINs were used to fix bugs or to revise an existing approved specification. A SIN applies to a specific version of a specification.

A WAP Forum document is identified by a *Document IDentifier* (DID). A specification keeps its associated DID for its entire lifespan (all revisions of the specification and the approved specification).

WAP Forum specifications are named according to the convention outlined in Figure 2.5.

Only approved specifications should be considered as a basis for the development of commercial WAP-based solutions. OMA members can download WAP Forum specifications from the OMA website at http://www.openmobilealliance.org.

In 2002, all WAP Forum activities have been transferred to the Open Mobile Alliance. The next section introduces the organization and working procedures of the Open Mobile Alliance.

2.9 Open Mobile Alliance

The Open Mobile Alliance (OMA) is a standardization forum established in June 2002 by nearly 200 companies representing the whole mobile services value chain. It is chartered to develop interoperable application enablers for the mobile industry. It intends to design specifications for applications enablers, which are bearer-agnostic and independent from any operating system. OMA was not created from scratch but was rather organized as a merge of several existing standardization forums. These forums, with sometimes overlapping activities, included the WAP Forum, the Wireless Village (WV), the MMS Interoperability Group (MMS-IOP), the SyncML Initiative, the Location Interoperability Forum (LIF), the Mobile Wireless Internet Forum (MWIF), and the Mobile Games Interoperability Forum (MGIF).

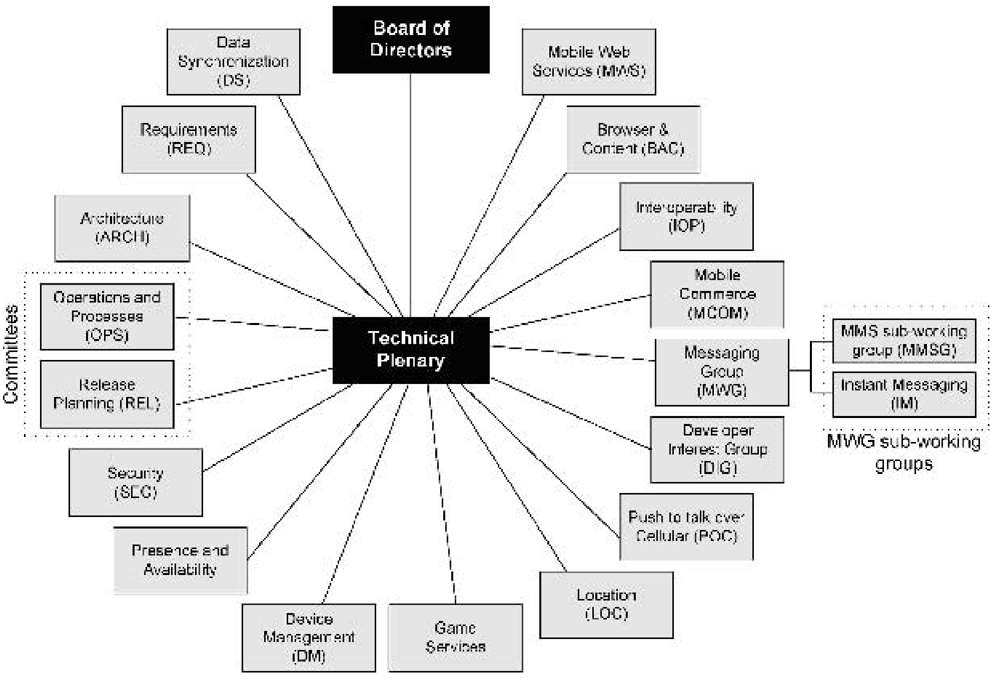


Figure 2.6 OMA organization

2.9.1 OMA Organization

In the OMA organization, the *technical plenary* is a chartered standing committee of the board of directors. The technical plenary is responsible for technical specification drafting activities, approval and maintenance of technical specifications, and resolution of technical issues. The organization of OMA is depicted in Figure 2.6

The task of OMA *Working Groups* (WGs) is to accomplish the technical work as defined by the technical plenary. On the other hand, *committees* are responsible for establishing rules for OMA operations and processes and for controlling the release of OMA specifications. Responsibilities of OMA working groups and committees are described below:

- *Requirements* (REQ): this WG is responsible for identifying use cases for services and identifying interoperability and usability requirements.
- *Architecture* (ARCH): this WG is in charge of the design of the overall OMA system architecture.
- *Messaging Group* (MWG): this WG is responsible for building application enablers for messaging services. Activities of this WG are delegated to sub-working groups. The *MMS Sub-Working Group* (MMSG) is responsible for the design of OMA MMS standards.
- *Mobile Web Services* (MWS): this WG is responsible for defining application enablers for web services in OMA.

- *Presence and Availability*: the objectives of this WG objectives are to identify, specify, and maintain the requirements, architecture, technical protocol/format/interface and interoperability specifications for presence and availability services.
- *Interoperability* (IOP): this WG focuses on testing interoperability and solving identified issues. Activities of this WG are delegated to sub-working groups (browsing, synchronization, IMPS, location, MMS, etc.). It also organizes test events as further described in Chapter 6.
- *Browser and Content* (BAC): this WG is responsible for the specification of application technologies such as download and DRM, push, the standard transcoding interface, and the user agent profile.
- *Device Management* (DM): this WG is in charge of specifying protocols and mechanisms that achieve management of devices (e.g., configuration settings, operating parameters, software installation and parameters, application settings, user preferences, etc.).
- *Data Synchronization* (DS): this WG designs specifications for data synchronization (including but not limited to the SyncML technology).
- *Game Services*: this WG is responsible for defining interoperability specifications, Application Programming Interfaces (APIs), and protocols for network enabled gaming.
- *Mobile Commerce* (MCOM): this WG aims at producing standards for technologies enabling mobile commerce in order to meet the demands of banking and financial industries, retailers, and mobile users.
- *Security* (SEC): this WG focuses on specifying the operation of adequate security mechanisms, features, and services by mobile clients, server, and related entities.
- *Location* (LOC): this WG designs an end-to-end architectural framework with relevant application and contents interfaces, privacy and security, charging and billing, and roaming for location-based services.
- *Push-to-talk over Cellular* (PoC): the PoC group is positioned to develop specifications for PoC services. PoC is a half-duplex form of communications allowing users to engage in immediate communication with one or more receivers simply by pushing a handset button (similar to Walkie-talkie operation).
- *Developer Interest Group* (DIG): in this group, OMA software developers are able to express their requirements into OMA as input for other working groups.

In addition to the working groups, two committees are part of the OMA organization as described below:

- *Operations and Processes* (OPS): this committee provides support for OMA operation and process activities. This includes the support for liaising with other forums, the assessment for IT requirements and staffing needs, etc.
- *Release Planning* (REL): this committee is responsible for planning and managing OMA releases according to OMA specifications and interoperability testing programmes.

2.9.2 OMA Specifications

The OMA process for publishing public technical specifications is based on the delivery of enabler releases and interoperability releases. An enabler release is a set of specifications required for enabling the realization of a service such as MMS, browsing, digital rights management, etc. OMA technical specifications in a draft stage are only made available to OMA members and should not be considered as mature inputs for the design of commercial solutions. When a set of technical specifications has gained enough maturity to be considered for the development of commercial solutions, then OMA publicly releases the set of specifications in the form of an enabler release. Enabler releases evolve over a scale of two maturity phases as shown below:

- *Candidate enabler release* (phase 1): a candidate enabler release is an approved set of OMA specifications forming the basis of product implementations, which can be tested for interoperability.
- *Approved enabler release* (phase 2): an approved enabler release has passed phase 1 (candidate release) and associated interoperability test cases have been designed by OMA.

Additionally, multiple approved enabler releases (phase 2) can be grouped into a single *interoperability release*. For this purpose, a set of devices conform to the approved enabler releases are required to pass end-to-end interoperability tests.
An OMA specification is uniquely identified according to the convention shown in Figure 2.7.

2.9.3 Available Documents

At the time of writing, several sets of OMA specifications have been made publicly available. This includes the following candidate enabler releases (phase 1) – December 2004 status:

- OMA Billing Framework (version 1.0)
- OMA Browsing (version 2.2)
- OMA Client Provisioning (version 1.1)

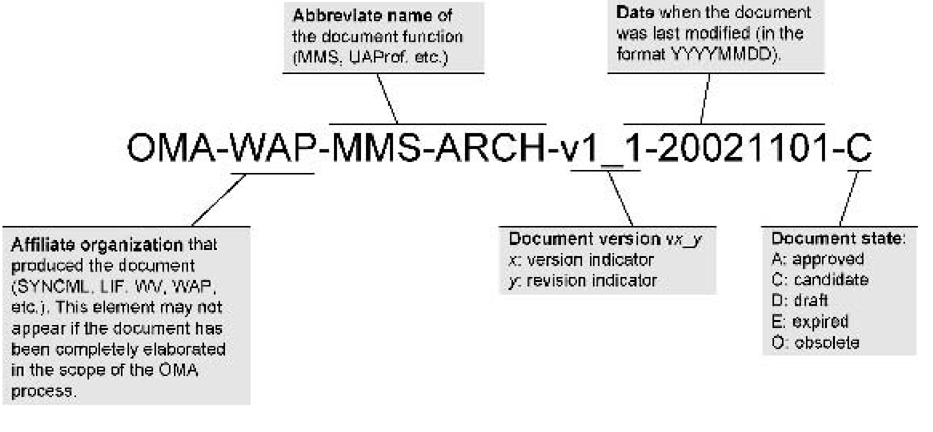


Figure 2.7 OMA specification numbering

- OMA Data Synchronization (version 1.2)
- OMA Digital Rights Management (version 2.0)
- OMA DNS (version 1.0)
- OMA Email Notification (version 1.0)
- OMA External Functionality Interface (version 1.1)
- OMA Games Services (version 1.0)
- OMA Instant Messaging and Presence Service (version 1.2)
- OMA Mobile Location Protocol (version 3.1)
- OMA Multimedia Messaging Service (version 1.2)
- OMA Online Certificate Status Protocol Mobile Profile (version 1.0)
- OMA SyncML Common Specification (version 1.2)
- OMA User Agent Profile (version 2.0)
- OMA Wireless Public Key Infrastructure (version 1.0)

The following approved enabler releases (phase 2) are also publicly available – December 2004 status:

- OMA Data Synchronization (version 1.1.2)
- OMA Device Management (version 1.1.2)
- OMA Digital Rights Management (version 1.0)
- OMA Download (version 1.0)
- OMA Instant Messaging and Presence Service (version 1.1)
- OMA Multimedia Messaging Service (version 1.1)
- OMA SyncML Common Specification (version 1.1.2)
- OMA Web Services (version 1.0)

OMA technical specifications can be downloaded from http://www.openmobilealliance.org.

2.10 Further Reading

[1] J. Huber, D. Weiler, and H. Brand, *UMTS, the mobile multimedia vision for IMT-2000: a focus on standardization*, IEEE Communications Magazine, September 2000.
[2] P. Loshin, *Essential Email standards*, *John Wiley & Sons, Ltd*, Chichester, 1999.
[3] 3GPP TR 21.900, *3GPP working methods.*
[4] WAP work processes, WAP Forum, December 2000 (WAP-181-TAWP-20001213-a).
[5] RFC 2026, *The Internet standards process — Revision 3*, IETF, October 1996.
[6] M.L. Olsson and J. Hjelm, *OMA — Changing the mobile standards game*, Ericsson Review, issue no. 1, 2003.
[7] F. Hillebrand (Editor), *GSM and UMTS: the creation of global mobile communication*, John Wiley & Sons, Ltd, Chichester, 2001.

3

Short Message Service

The Short Message Service (SMS) is a basic service allowing the exchange of short text messages between subscribers. The first short text message is believed to have been transferred in 1992 over signaling channels of a European GSM network. Since this successful trial, SMS usage has been the subject of tremendous growth. The Mobile Data Association[1] reported that the total number of chargeable person-to-person text messages sent across the four UK GSM networks in 2003 totaled 20.5 billion.

This chapter, dedicated to SMS, first introduces common use cases for SMS such as consumer, corporate, and operator applications. Components of a typical SMS-enabled GSM architecture are presented along with basic SMS features. The four-layer transport protocol stack of SMS (application, transfer, relay, and link) is presented and the transfer layer of this stack is described in detail. The transfer layer is the component which needs to be mastered by implementers for the development of SMS-based applications. An insight is also given into techniques available for exchanging messages between servers and applications running in the SIM. Interworking between SMS and Email is presented and the manipulation of messages via AT commands is illustrated.

Note that the content of this chapter also represents the basis for the next chapter. Chapter 4 describes two application-level extensions of SMS in the form of the Enhanced Messaging Service (EMS).

3.1 Service Description

Developed as part of the GSM Phase 1 ETSI technical specifications, the Short Message Service (SMS) allows mobile stations and other network-connected devices to exchange short text messages. Work on the standardization of SMS was initiated by ETSI and is now being carried out in the scope of 3GPP activities. Since its initial introduction in GSM networks, SMS has been ported to other network technologies such as GPRS and CDMA.

[1]http://www.mda-mobiledata.org/

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2

The Short Message Service allows users to exchange messages containing a short amount of text. These messages can be sent from GSM/UMTS mobile devices but also from a wide range of other devices such as Internet hosts, telex, and facsimile. The SMS is a very mature technology supported by 100% of GSM handsets and by most GSM networks worldwide.

3.2 SMS Use Cases

SMS was intended to be a means of exchanging limited amounts of information between two mobile subscribers. This limited capability has become a building block for the development of more compelling services ranging from the download of ringtones to professional applications such as remote monitoring or fleet tracking. SMS use cases described in this chapter are representative of typical applications based on SMS, including consumer applications, corporate applications, and operator applications.

3.2.1 Consumer Applications Based on SMS

In this category are grouped services such as person-to-person messaging, information services, download services, or chat applications. Consumers have access to these services for customizing their handsets, receiving information from remote servers, or simply for exchanging information between friends.

Person-to-Person Messaging

This is the original use case for which SMS has been designed. This use case relates to the exchange of a short text message between two mobile subscribers. The subscriber that originates the message first composes the message using the man–machine interface of the mobile device. Usually, the message text is entered via the handset keyboard and echoed on the handset display. After composition, the subscriber enters the phone number of the message recipient and sends it to the serving network. The message is then transported over one or more networks before reaching the recipient mobile network. If the recipient handset is able to handle the message immediately, then the message is transferred to the recipient mobile handset and the recipient subscriber is notified that a new message has arrived. Otherwise the message is kept temporarily by the network until the recipient handset becomes available. It is not easy to input text with a small handset keypad. In order to cope with this limitation, handsets are usually shipped with a predictive text input mechanism. These mechanisms anticipate which word the user is trying to enter by analyzing the most recent characters or symbols which have been typed in and checking those against entries of a static or dynamic dictionary. Entries from the dictionary which closely match the partially entered word are then presented to the subscriber who can select one of them if appropriate. These mechanisms significantly reduce the number of keys that need to be pressed to input text for the composition of a message. Predictive text input mechanisms are available for various different languages. With SMS, the two most well known predictive text input algorithms are T9® from Tegic[2] and ZI from ZI corporation.[3] It has to be noted that

[2]http://www.t9.com/
[3]http://www.zicorp.com/

Figure 3.1 Mobile device with external keyboard – reproduced by permission of Alcatel Business Systems

advanced handsets are sometimes equipped with a built-in QWERTY keyboard. Alternatively, a small external keyboard can sometimes be connected to a handset as shown in Figure 3.1. Because of the difficulty of entering text with mobile handsets, subscribers have forged their own dialect. This dialect consists of using abbreviated terms that are quick to compose on mobile handsets. For instance, ‘‘RUOK’’ means ‘‘Are you OK’’? and ‘‘C U L8ER’’ means ‘‘See you later!’’ This is sometimes complemented by the use of text-based pictograms or ‘‘smileys.’’ For instance, :-) represents a ‘‘happy mood’’ and :-(represents a ‘‘sad mood.’’

Information Services

This is probably one of the most common use cases in the machine-to-person scenario. With information services, weather updates and financial reports can be prepared by value-added service providers and pushed to mobile handsets with SMS. For these services to be activated, it is usually necessary for the user to first subscribe manually to the service prior to receiving associated reports and updates.

Voice Message and Fax Notifications

This use case is widely supported in GSM mobile networks. This use case relates to the reception of messages containing notifications for voice messages and fax waiting in a remote message inbox.

Internet Email Alerts

With Email alerts via SMS, subscribers are notified that one or more Email messages are waiting to be retrieved. Such an alert usually contains the address of the message originator along with the message subject and the first few words from the Email message body.

Download Services
It has become popular for mobile subscribers to customize their mobile handset. This can be done by associating ringtones to persons in the phone contact directory. With such a configuration, an incoming phone call from an identified person triggers the selected ringtone. It is also common to change switch on and off animations or to change the general look-and-feel (also known as "skin") of the handset's graphical user interface. All these objects used to customize the mobile handset can be downloaded as part of one or more short messages.

Chat Applications
During a chat session, several users can exchange messages in an interactive fashion. All messages exchanged during a session are kept in chronological order in a chat history. In the chat history, messages sent from a recipient are differentiated from messages sent from other users. Several existing mobile chat applications are based on SMS for the transport of messages.

Smart Messaging
Smart Messaging is a proprietary service developed by Nokia. This service enables the exchange of various objects via SMS. This includes the transfer of Internet configuration parameters, business cards for PIM updates, etc. A popular feature of this service is called "Picture Messaging," which allows the association of one bitmap picture to the text of a message.

3.2.2 Corporate Applications Based on SMS

In this category are grouped services tailored for the need of professionals. This includes vehicle positioning and remote monitoring of machines.

Vehicle Positioning
The Global Positioning System (GPS) is a technology for determining global positions on Earth. The device determines its location by analyzing signals broadcast from a group of satellites. The location information is usually expressed in terms of longitude, latitude, and sometimes altitude. A GPS receiver coupled to a handset, built-in or as an accessory, can provide the location of a person or equipment. This location information can be formatted in a short message and sent to a remote server via SMS. The server interprets locations received from several handsets and displays them on associated geographical maps. Such an application can help logisticians to keep track of a fleet of trucks or policemen to track down stolen vehicles.

Remote Monitoring
Messages can transport information about the state of remote devices. For instance, system administrators can be notified by a short message that a server is running low of resources or that a fault has been detected on a remote computer.

3.2.3 Operator Applications Based on SMS

Operators have used SMS as a building block for enabling the realization of several services including the ones listed below.

SIM Lock
Operators sometimes require handsets to be locked and usable with only one specific SIM. After a minimal subscription period, the user may request the operator to deactivate the lock in order to be able to use the mobile handset with another SIM (from the same operator or from another operator). If the operator agrees on the lock being deactivated, then the operator sends a short message containing a code allowing the device to be unlocked.

SIM Updates
With SMS, operators can remotely update parameters stored in the SIM. This is performed by sending one or more messages with new parameters to a mobile device. In the past, operators have used this method for updating voice mail access numbers, customer service profiles (determining which network services are accessible to the subscriber), operator name for display in idle mode on the device screen, and address book entries.

Message Waiting Indicator
Operators have used SMS as a simple way to update message waiting indicators on the receiving handset. With this mechanism, a short message contains the type of indicator (voice mail, etc.) to be updated along with the number of waiting messages.

WAP Push
The SMS can be used as a bearer for realizing the WAP push. With this configuration, a WSP protocol data unit or the URI of the content to be retrieved is encoded in a short message and sent to the receiving device. Upon reception of such a message, the WAP microbrowser intercepts the message, interprets the pushed content, and presents the content to the subscriber.

3.2.4 Value Chain of SMS-Based Applications

In the person-to-person scenario, SMS involves two persons, the message originator and the message recipient. In addition, one or more network operators are involved in transporting the message. In the machine-to-person scenario, the business model may become more complex. Indeed, the business model involves a person, the message recipient, one or more network operators to transport the message, and various intermediaries such as service providers, portal providers, SMS resellers, etc.

To cope with the high demand for transferring short messages in the machine-to-person scenario, a business model has been put in place by operators to provide wholesale SMS to service providers. This allows service providers to purchase SMS resources in bulk at a wholesale price from operators and to resell short messages with customized content to mobile users.

3.3 Architecture for GSM Networks

The realization of SMS implies the inclusion of several additional elements in the network architecture (GSM, GPRS, or UMTS). Figure 3.2 shows the architecture of an SMS-enabled GSM network.

The two additional network elements in the architecture are the SMS center and the Email gateway. In addition, an element called the short message entity, usually in the form of a software application in a mobile device, is necessary for the handling of messages (sending,

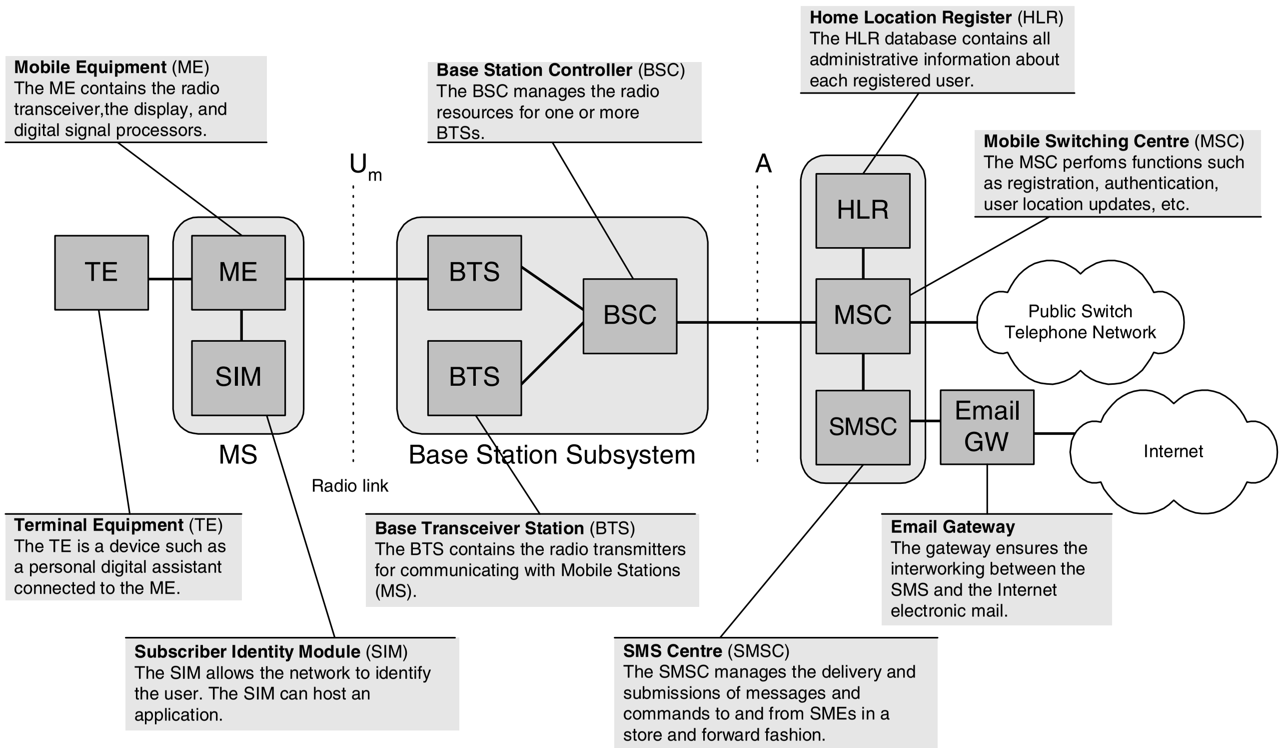


Figure 3.2 SMS-enabled GSM network architecture

reception, storage, etc.). The short message entity is not shown in Figure 3.2. The SMS center, Email gateway, and the short message entity are presented in the following sections.

3.3.1 Short Message Entity

Elements that can send or receive short messages are named *Short Message Entities* (SME). An SME can be a software application in a mobile handset but it can also be a facsimile device, telex equipment, remote Internet server, etc. A mobile handset has to be configured in order to operate properly in a mobile network. The handset is typically pre-configured during the manufacturing process but a manual configuration can also be performed. Figure 3.3 shows an example of the graphical user interface for updating SMS settings in a GSM/GPRS handset.

An SME can be a server that interconnects to the SMS center directly or via a gateway. Such an SME is also known as an *External SME* (ESME). Typically, an ESME represents a WAP proxy/server, an Email gateway, or a voice mail server.

For the exchange of a short message, the SME which generates and sends the short message is known as the *originator SME* whereas the SME which receives the short message is knows as the *recipient SME*.

3.3.2 Service Center

The *Service Center* (SC) or *SMS Center* (SMSC) plays a key role in the SMS architecture. The main functions of the SMSC are the relaying of short messages between SMEs and the store-and-forwarding of short messages (storage of messages if the recipient SME is not available). The SMSC may be integrated as part of the mobile network (e.g., integrated to the MSC) or as an independent network entity. The SMSC may also be located outside the network and be managed by a third party organization. Practically, it is very common for network operators to acquire one or more SMSCs since SMS is now considered as a very popular service to be provided by any mobile network. In theory, one single SMSC could manage SMS for several mobile network operators. However, this latter scenario is seldom

Figure 3.3 SMS settings–handset screenshot

encountered in real life and one or more SMSCs are often dedicated to the management of SMS operations in one single mobile network.

Mobile network operators usually have mutual commercial agreements to allow the exchange of messages between networks. This means that a message sent from an SME attached to a network A can be delivered to another SME attached to a mobile network B. This ability for users to exchange messages even if they are not subscribers to the same network and sometimes located in different countries is undoubtedly one of the key features that makes SMS so successful.

Today's commercial SMS centers are typically able to process more than 1000 messages per second each. Large mobile operators often make use of several SMS centers to fulfill the needs of their customers.

3.3.3 Email Gateway

The *Email gateway* enables an Email-to-SMS interoperability by interconnecting the SMSC with the Internet. With the Email gateway, messages can be sent from an SME to an Internet host, and vice versa. The role of the Email gateway is to convert message formats (from SMS to Email and vice versa) and to relay messages between SMS and Internet domains.

3.4 SMS Basic Features

SMS encompasses a number of basic features. This includes message submission, message delivery, handling of status reports, requests for command execution, reply path, etc. These features are presented in the following sections.

3.4.1 Message Submission and Delivery

The two most basic features of SMS are the sending and the receipt of a short message.

Message Sending

Mobile-originated messages are messages which are submitted from an MS to an SMSC. These messages are addressed to other SMEs such as other mobile users or Internet hosts. An originator SME may specify a message validity period after which the message is no longer valid. A message which is no longer valid may be deleted by an SMSC during the message transfer. With the first GSM networks, not all handsets supported message submission. Presently, almost all handsets support message submission. This feature is also known as the Short Message-Mobile Originated (SM-MO).

Message Delivery

Mobile-terminated messages are messages delivered by the SMSC to the MS. Nearly all GSM handsets support message reception. This feature is also known as Short Message Mobile Terminated (SM-MT). Mobile-Originated and Mobile-Terminated short messages can be delivered or submitted even while a voice call or data connection is under progress. Messages can be sent or received over GSM signaling channels, but also over GPRS channels. In GSM, messages are sent over SDCCH or SACCH channels whereas, in GPRS, short messages are sent over PDTCH channels. The choice of the bearer for transporting a message is usually made according to a predefined network policy.

3.4.2 Status Reports

It is possible for an originator SME to request that a status report be generated upon delivery of the short message to the recipient SME. The status report indicates to the originator whether or not the short message has been successfully delivered to the recipient SME.

3.4.3 Reply Path

The reply path can be set by the originator SME (or the serving SMSC) to indicate that the serving SMSC is able and willing to directly handle a reply from the recipient SME in response to the original message. In this situation, the recipient SME usually submits the reply message directly to the SMSC that serviced the submission of the original message.

This feature is sometimes used by operators to allow the message recipient to provide a reply message ‘‘free of charge’’ for the message recipient.

Additionally, for networks supporting several SMSCs, operators sometimes use this feature to get reply messages to be returned to a particular SMSC. For example, an operator could have several SMSCs but only one connected to the Email gateway. In this configuration, if a message is originated from the Internet domain, then the operator uses the reply path to indicate that any message reply associated with this Email-originated message should be submitted to the SMSC connected to the Email gateway. In this situation, the SMSC can appropriately request the Email gateway to convert reply messages into Email messages and to deliver them to the Email recipient(s).

3.4.4 Addressing Modes

With SMS, several modes are possible for addressing message recipients. The most common addressing mode consists of using the Mobile Station ISDN Number (MSISDN) in the [ITU-E.164] format (e.g., +33612345678). However, other less commonly used addressing modes are available such as Email addressing, as defined by the IETF in [RFC-2822], or operator specific numbering schemes (short codes for instance).

An optional addressing feature of SMS consists of conveying sub-addressing information as part of a message. The sub-addressing information is appended at the end of the recipient address by the originator SME before message sending. When this scheme is applied, the SMSC extracts the sub-addressing information from the recipient address specified for the submitted message and appends it to the originator address for the message to be delivered. This optional feature can be used to maintain session identifications for the exchange of messages or for identifying a specific service code to which a message relates. The sub-addressing information is separated from the normal addressing information by the ‘‘#’’ delimiter (this character is part of the sub-addressing information) as shown in Figure 3.4.

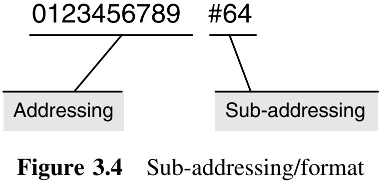


Figure 3.4 Sub-addressing/format

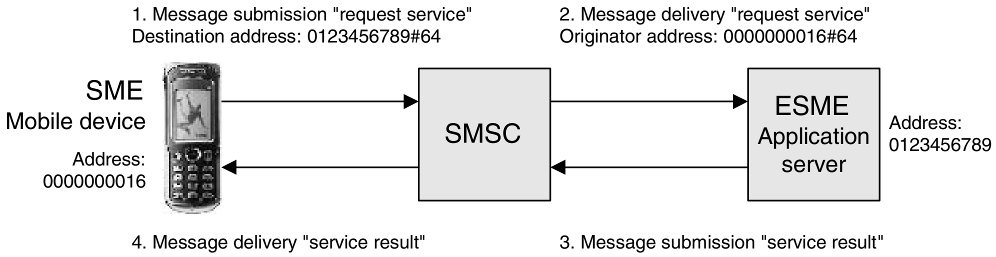


Figure 3.5 Sub-addressing/example

The sub-addressing information is a combination of digits and "#" and "*" characters. For instance, a mobile handset may request a weather forecast update by sending a message to an application server. The identification of the requested service (e.g., service 64) can be specified as part of the sub-addressing information as shown in Figure 3.5.

A message originator has the possibility of indicating an alternate reply address as part of a submitted message. Such an alternate reply address should be used by the recipient SME if the recipient wishes to reply to the message.

Sub-addressing and alternate reply address features are recent evolutions of the SMS standard and still have to be supported by SMEs and SMSCs.

Note that a limitation of SMS, and to some extent EMS, resides in its inability to support group sending, at the transport level. A message can only be addressed to one single recipient. Group sending can be emulated by the originator SME by sending the same message a number of times to the network, one time per message recipient. However, this method is not very efficient in terms of usage of network resources. Alternatively, operators sometimes cope with this limitation by offering a distribution list service. With this service, subscribers are able to manage (create/modify/delete) personal distribution lists. Each distribution list is identified by a short code. Once a distribution list has been configured, the subscriber can send a message to all recipients of a distribution list by sending the message just once to the corresponding short code. Upon reception of the message, the network generates and delivers separate copies of the message to all recipients. In this situation, the message is only transferred once by the originator SME to the network. This leads to a more efficient use of network resources. The message originator is still charged by the network operator for the number of copies generated by the network, as if they had been submitted directly from the originator SME.

3.4.5 Validity Period

A message originator has the possibility of indicating a validity period for a message. This validity period defines the deadline after which the message content is to be discarded. If a message has not been delivered to the message recipient before the expiry date, then the network usually discards the message without further attempts to deliver it to the recipient. For instance, a subscriber may send a message with the following content "please phone me in the coming hour to get your answer." Additionally, the subscriber may wisely indicate that the message validity period is limited to 1 hour. In the situation where the message recipient does not turn on his/her mobile device in the hour following the message sending, then the network can decide to discard the message. Consequently, in this example, the message will never be delivered unless the recipient successfully retrieves the message in the hour following the message sending.

Note that mobile operators often assign a default validity period for messages transiting in their network (e.g., 2 days).

3.5 Technical Specification Synopsis

SMS is defined in a number of 3GPP technical specifications as shown in Table 3.1.

3.6 Protocol Layers

The SMS protocol stack is composed of four layers: the application layer, the transfer layer, the relay layer, and the link layer. This book outlines each layer and provides an in-depth description of the transfer layer. SMS-based applications are directly based on the transfer layer. Consequently, any engineer willing to develop applications, for which SMS is a building block, needs to master the transfer layer.

The *application layer* is implemented in SMEs in the form of software applications that send, receive, and interpret the content of messages (e.g., message editor, games, etc.). The application layer is also known as SM-AL for Short-Message-Application-Layer.

At the *transfer layer*, the message is considered as a sequence of octets containing information such as message length, message originator or recipient, date of reception, etc. The transfer layer is also known as the SM-TL for Short-Message-Transfer-Layer.

The *relay layer* allows the transport of a message between various network elements. A network element may temporarily store a message if the next element to which the message is to be forwarded is not available. At the relay layer, the MSC handles two functions in addition to its usual switching capabilities. The first function called SMS gateway MSC (SMS-GMSC) consists of receiving a message from an SMSC and interrogating the HLR to obtain routing information and further to deliver the message to the recipient network. The second function called SMS InterWorking MSC (SMS-IWMSC) consists of receiving a

Table 3.1 SMS technical specification synopsis

TS reference	Title
3GPP TS 22.205	Services and service capabilities Note: this specification identifies the high-level requirements for SMS
3GPP TS 23.011	Technical realization of supplementary services Note: this specification presents the use of radio resources for the transfer of short messages between the MS and the MSC or the SGSN.
3GPP TS 23.038	Alphabets and language-specific information
3GPP TR 23.039	Interface protocols for the connection of SMSCs to SMEs
3GPP TS 23.040	Technical realization of the short message service
3GPP TS 23.042	Compression algorithm for text messaging services
3GPP TS 24.011	Point-to-point Short Message Service (SMS) support on mobile radio interface
3GPP TS 27.005	Use of Data Terminal Equipment–DTE–DCE interface for SMS and CBS.
3GPP TS 43.041	Example protocol stacks for interconnecting SCs and MSCs.

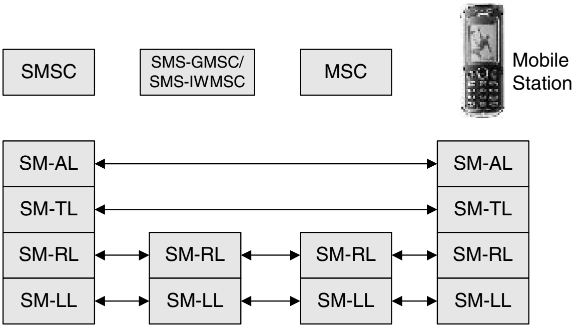


Figure 3.6 SMS protocol stack

message from a mobile network and submitting it to the serving SMSC. The relay layer is also known as the SM-RL for Short-Message-Relay-Layer.

The *link layer* allows the transmission of the message at the physical level. For this purpose, the message is protected to cope with low-level channel errors. The link layer is also known as the SM-LL for Short-Message-Link-Layer.

The stack of transport protocol layers for SMS is shown in Figure 3.6. For transport purposes, an application maps the message content and associated delivery instructions onto a *Transfer Protocol Data Unit* (TPDU) at the transfer layer. A TPDU is composed of various parameters indicating the type of the message, specifying whether or not a status report is requested, containing the text part of the message, etc. Each parameter is prefixed by the abbreviation `TP` for Transfer Protocol such as `TP-Message-Type-Indicator` (abbreviated `TP-MTI`), `TP-Status-Report-Indicator` (abbreviated `TP-SRI`), `TP-User-Data` (abbreviated `TP-UD`), etc.

At the transfer layer, the exchange of a message from the originator SME to the recipient SME consists of two to three steps. The three steps are shown in Figure 3.7. After creation by the originator SME, the message is submitted to the SMSC (step 1). The SMSC may verify with other network elements that the message originator is allowed to send messages (e.g., sufficient prepaid credit, subscriber belongs to the network, etc.). The SMSC delivers the message to the recipient SME (step 2). If the recipient SME is not available for the message delivery, then the SMSC stores the message temporarily until the recipient SME becomes available or until the message validity period expires. Upon delivery of the message or upon message deletion by the network, a status report might be transferred back to the originator SME (step 3), only if this report had been requested by the originator SME during message submission.

3.6.1 SMS Interworking Between Mobile Networks

With GSM/GPRS technologies, mobile operators can easily support the exchange of messages between distinct networks. For this purpose, operators have commercial agreements. Each mobile network counts the number of messages being sent from another network. After a given period of time, these counts are compared and there is a commercial settlement between operators. In the simplest configuration, MAP signaling transactions for SMS are allowed between the two mobile networks. In this configuration, the SMSC of the

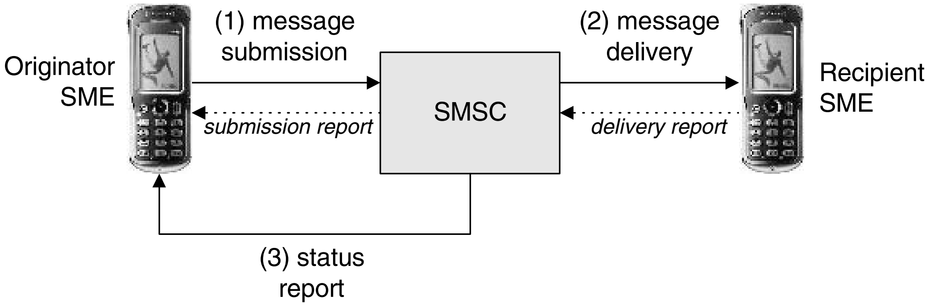


Figure 3.7 Message transfer between two SMEs

originator SME queries the HLR of the relevant destination network to obtain the necessary routing information and forwards the message directly towards the message recipient. In this case, the SMSC of the message recipient is not involved in the message delivery. At the transfer layer, the different steps involved in the exchange of a message, in this configuration, are basically those presented in Figure 3.7. This configuration is supported by the four GSM mobile network operators in the United Kingdom.

Unfortunately, things are more complicated for ensuring interoperability between networks based on different technologies (e.g., between GSM/GPRS and CDMA) or when signaling interconnections cannot be supported between distinct networks. This is the case, for instance, for many North American networks[4] where a wide variety of technologies are available. For these configurations, the exchange of messages between distinct networks can still be offered by interconnecting the two mobile networks with a gateway or interconnecting the two SMSCs with a proprietary exchange protocol. This latter configuration, considered from the transfer layer, is depicted in Figure 3.8. In this configuration, the exchange of a message between two subscribers consists of three to four steps. After creation by the message originator, the originator SME submits the message to the originator SMSC (step 1). The originator SMSC forwards the message towards the recipient SMSC (step 2) and the recipient SMSC delivers the message to the recipient SME (step 3). If a status report was requested by the message originator, then the recipient SMSC generates a status report and transfers it back to the originator SME (step 4).

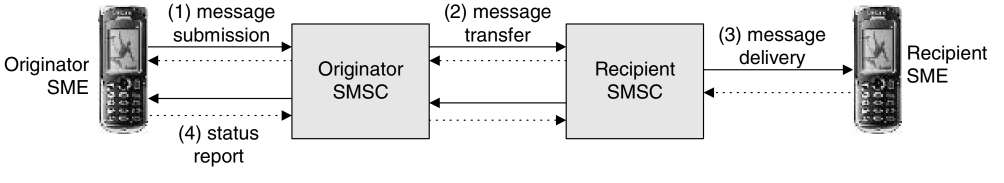


Figure 3.8 Message transfer

[4] Due to the difficulty of ensuring SMS interoperability in these configurations, operators sometimes decide not to offer the ability for their subscribers to exchange messages with subscribers belonging to another network

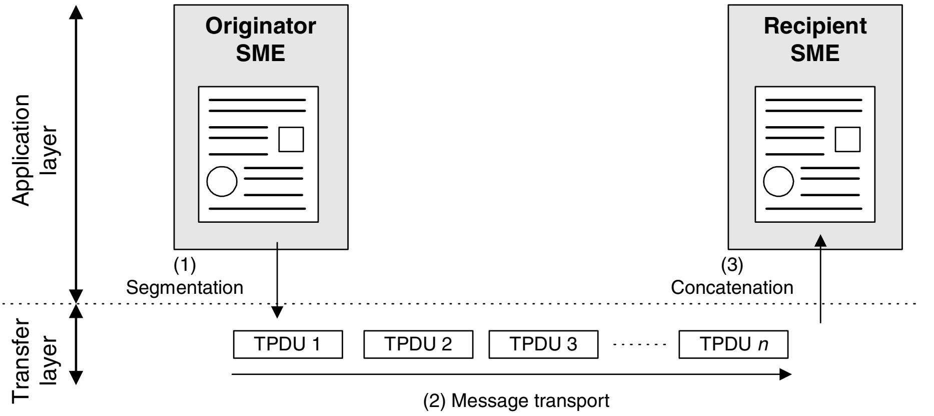


Figure 3.9 Message structure

3.6.2 Message Structure

In this chapter, the following terminology is used for referring to messages.

A *message* refers to the subscriber's perception of the message composed of text and/or elements such as pictures, melodies, etc. For transport purposes and due to limitations at the transfer layer, an application may need to segment the message into several pieces called *message segments*. A one-segment message is also known as a short message. A message segment is an element manipulated by an application.

A message segment has a limited payload size. In order to convey a large amount of data, several message segments can be combined into a *concatenated message*.[5] The message concatenation is handled at the application layer. In order to be transported, the message segment needs to be mapped onto a TPDU at the transfer layer as shown in Figure 3.9.

In the scope of the download service, a message containing one or more objects being downloaded is called a *download message*.

In this book, the word *octet* refers to a group of eight bits (also known as a byte) whereas a *septet* refers to a group of seven bits.

3.6.3 SME-SMSC Transactions/Submit, Deliver, Report, and Command

At the transfer layer, six types of transactions can occur between an SME and an SMSC. A TPDU type corresponds to each one of the transaction types. Figure 3.10 shows the possible transactions between an SME and the serving SMSC.

- SMS-SUBMIT: this transaction corresponds to the submission of a message segment from the SME to the SMSC. Upon submission of the message segment, the SMSC acknowledges the submission with the SMS-SUBMIT-REPORT transaction.

[5] In several documents dealing with SMS, a concatenated message is also known as a long message

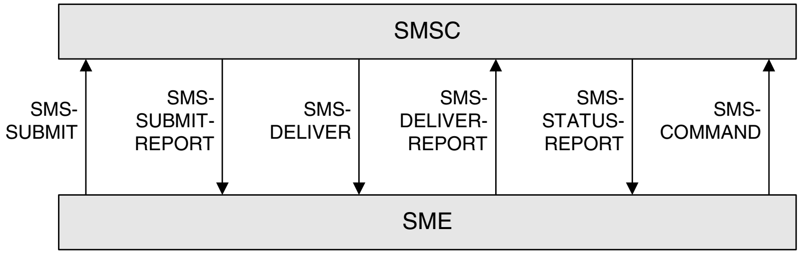


Figure 3.10 Transaction types

- SMS-DELIVER: this transaction corresponds to the delivery of a message segment from the SMSC to the SME. Upon delivery of the message segment, the SME acknowledges the delivery with the SMS-DELIVER-REPORT transaction.
- SMS-STATUS-REPORT: this transaction corresponds to the transfer of a status report from an SMSC back to an SME.
- SMS-COMMAND: this transaction corresponds to the request from an SME, usually an external SME, for the execution of a specific command by the SMSC.

3.7 Structure of a Message Segment

A message segment is associated with a number of parameters. These parameters indicate the message type, class, coding group, etc. In addition, parameters also contain the message content which is provided by the subscriber, the content provider, or content which is automatically generated by a machine.

3.7.1 Transport Protocol Data Unit

As introduced in this chapter, a TPDU type has been assigned to each transaction that can occur between an SME and an SMSC at the transfer layer. Depending on its type, a TPDU is composed of a varying number of parameters organized according to a predefined TPDU layout. The partial representation of a TPDU is shown in Figure 3.11.

High-level parameters in the TPDU (the ones for which the name starts with `TP`) inform on the transaction type (`TP-Message-Type-Indicator`), the presence of binary elements in the message such as concatenation instructions (`TP-User-Data-Header-Indicator`), etc.

One of the important parameters is the user data parameter[6] (`TP-User-Data`). If present, this parameter contains the text part of a message segment and may also contain binary elements such as concatenation instructions, pictures, melodies, etc. To cope with the complexity of this parameter, the parameter is divided into two sub-parts. The first sub-part, known as the User-Data-Header (UDH), contains binary elements whereas the remaining

[6]The user data parameter is not present in the Command transaction type

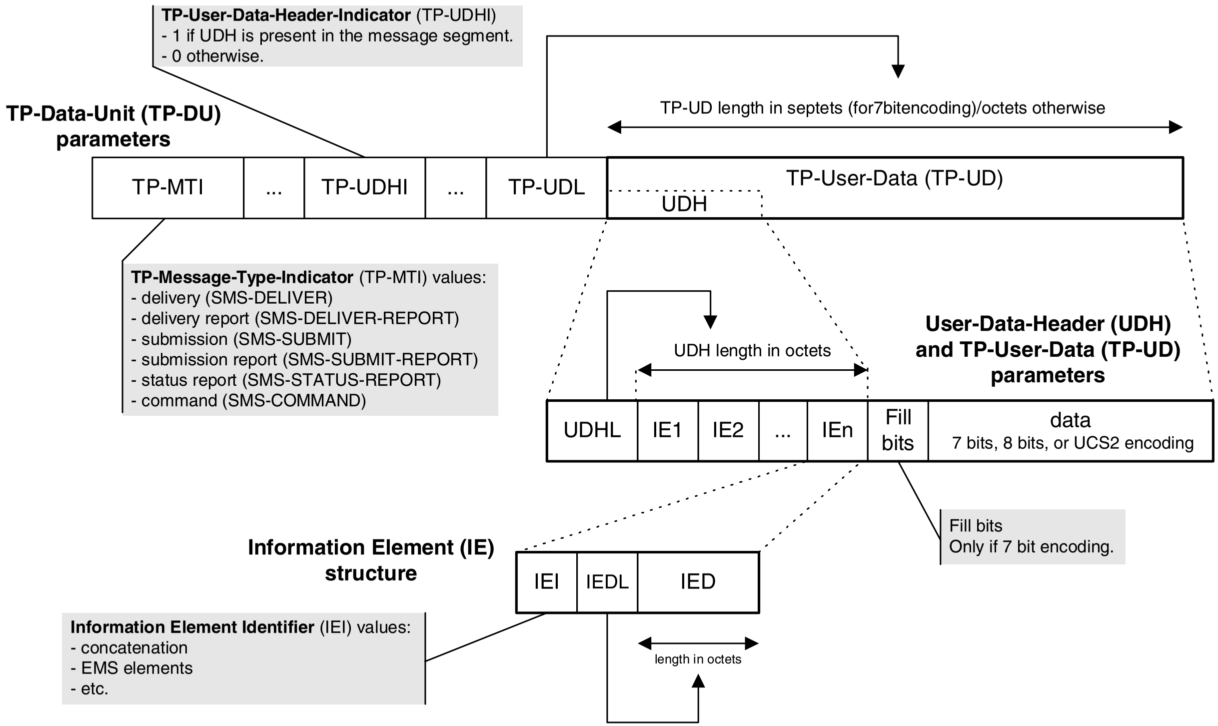


Figure 3.11 Structure of the `TP-Data-Unit`

sub-part contains the message text. The UDH is itself structured as a sequence of sub-parameters. The first sub-parameter, the User-Data-Header-Length (UDHL), indicates the length of the UDH in octets. A set of information elements immediately follows the UDHL.

Whatever the coding of the `TP-User-Data`, the User-Data-Header is always 8-bit encoded. Consequently, if the `TP-User-Data` is 7-bit aligned, then fill bits may be inserted between the UDH and the remaining part of the `TP-User-Data`. With this method, a 7-bit data part always starts on a 7-bit data boundary of the `TP-User-Data` as shown in Figure 3.11. This allows older handsets that do not support the User-Data-Header concept to still be able to present the 7-bit text part of the message to the subscriber.

An in-depth presentation of the `TP-User-Data` and the User-Data-Header are provided in Section 3.15.

3.7.2 Message Types

The transaction type associated with a message is indicated in one of the message parameters. The set of possible transaction types that can occur between an SMSC and an originator or receiving SME are the message submission (`SMS-SUBMIT`) and the corresponding submission report (`SMS-SUBMIT-REPORT`), the message delivery and the corresponding message delivery report (`SMS-DELIVERY-REPORT`), the transfer of a status report (`SMS-STATUS-REPORT`), and the submission of a command (`SMS-COMMAND`). A dedicated value, assigned to the `TP-Message-Type-Indicator` (abbreviated `TP-MTI)` parameter, corresponds to each transaction type.

3.7.3 Text Coding Schemes

The text part of a message can be encoded according to several text alphabets. The two text coding schemes that can be used in SMS are the GSM 7-bit default alphabet defined in [3GPP-23.038] and the Universal Character Set (UCS2) defined in [ISO-10646]. The two text alphabets are presented in Appendix C. The amount of text that can be included in a message segment needs to fit into 140 octets. Since the two text coding schemes utilize one septet and two octets, respectively, to encode a character/symbol, the amount of text that can be included in a message segment is as shown in Table 3.2.

The value assigned to the `TP-Data-Coding-Scheme` parameter indicates which coding scheme has been used for encoding the message content. In its simplest form, the GSM 7-bit default alphabet is composed of 128 characters plus 9 additional characters (extension table) including the Euro sign. The Universal Character Set with 2-octet symbols

Table 3.2 Relation between coding scheme and text length

Coding scheme	Text length per message segment	`TP-DCS`	
		bit 3	bit 2
GSM alphabet, 7 bits	160 characters	0	0
8-bit data	140 octets	0	1
USC2, 16 bits	70 complex characters	1	0

(USC2) is used for encoding complex sets of non-Latin characters such as Chinese and Arabic.

3.7.4 Text Compression

In theory, the text part of a message may be compressed [3GPP-23.042]. However, none of the handsets currently available on the market support text compression. It is therefore not advised to generate messages with compressed text unless one can ensure that text decompression is supported by the recipient SME. A message with compressed text cannot be displayed properly by an MS that does not support text decompression.

3.7.5 Message Classes

In addition to its type, a message belongs to a class. The `TP-Data-Coding-Scheme` (`TP-DCS`) parameter of the TPDU indicates the class to which the message belongs. Four classes have been defined and indicate how a message should be handled by the receiving SME. Table 3.3 provides a short description of each message class.

It has to be noted that in most cases, a message does not belong to any of the four classes. In this situation the message is known as a *no-class* message and is usually handled as a class 1 or 2 message by the receiving SME.

3.7.6 Coding Groups

A message may belong to one of three coding groups. The coding group indicates what the receiving SME should do with the message once it has been read or interpreted. The

Table 3.3 Message classes

Class	Description	`TP-DCS` value bits 7...4 = 00xx or bits 7...4 = 1111	
		bit 1	bit 0
Class 0	**Immediate display message** Messages belonging to class 0 are immediately presented on the recipient device display.	0	0
Class 1	**Mobile equipment specific message** If possible, messages belonging to class 1 are stored in the ME. Otherwise, class 1 messages may be stored in the SIM.	0	1
Class 2	**SIM specific message** Messages belonging to class 2 are stored in the SIM.	1	0
Class 3	**Terminal equipment specific message** Messages belonging to class 3 are transferred to the terminal equipment (PDA, personal computer, etc.) which is connected to the ME.	1	1

Table 3.4 Coding groups

Group name	Description	TP-DCS value bits 7...4
Message marked for **automatic deletion**	Upon reading, a message marked for automatic deletion is deleted by the recipient SME. The message can be of any class. A configuration setting may allow the user to disable the automatic deletion.	0 1 x x
Message waiting indication group: **discard message**	Messages belonging to this group are used for informing the subscriber of the status of messages waiting for retrieval. The message is processed independently from SME storage availability and is usually discarded after processing.	1 1 0 0
Message waiting indication group: **store message**	Messages belonging to this group are used for informing the subscriber of the status of messages waiting retrieval. If possible, the ME updates the message waiting indicator status on the (U)SIM. Otherwise, statuses are stored in the ME.	1 1 0 1

`TP-Data-Coding-Scheme` (`TP-DCS`) indicates the coding group to which the message belongs. The list of available coding groups is provided in Table 3.4.

3.7.7 Protocol Identifiers

Previous sections have shown that a short message has a type, and may belong to a class and to a coding group. A message is also associated with a protocol identifier. The protocol identifier indicates how the receiving messaging application should handle an incoming message (normal case, ME data download, SIM data download, etc.). The protocol identifier values shown in Table 3.5 can be assigned to the `TP-Protocol-Identifier` (`TP-PID`) parameter.

3.8 Settings and Message Storage in the SIM

SMS settings (default protocol identifier, service center address, etc.), user preferences (validity period, default recipient address, etc.), and messages can be stored in the SIM. The internal architecture of a SIM card is organized around a processor and there are the following three different types of memory:

- *Read Only Memory* (ROM): this memory contains the card operating system along with one or more applications. The memory can only be read and cannot be modified by applications.
- *Electrically Erasable Programmable Read Only Memory* (EEPROM): this memory contains all parameters defined by the GSM/3GPP technical specifications and data

Table 3.5 Protocol identifiers

TP-PID value (hex)	Description
0x00	**Normal case** This protocol identifier is used for a simple transfer from the SME to the SMSC and from the SMSC to the SME. The message is handled by the handset according to its class and presented accordingly to the subscriber.
0x01...0x1F	No telematic interworking, but **SME-to-SME protocol**.
0x20...0x3F	**Telematic interworking** Telematic interworking means that the message is sent from or to a telematic device such as a telefax, a teletex, or an Internet email system. All telematic interworking devices that can be identified with these protocol identifiers are listed in Appendix A.
0x40	**Short message type 0** The ME acknowledges the reception of such a message but discards its content. The common use case for using this message type is to page a mobile to check if the mobile is active without the user being aware that the mobile station has been paged.
0x41...0x47	**Replace short message type *n* (*n* from 1...7)** The replacement of short messages is not supported by all handsets. If supported, a handset replaces the previously received "replace short message type *n*" message by the lastly received message with the same protocol identifier (same replace type code) and same originating address.
0x48...0x5D	**Reserved** (22 values)
0x5E	**Obsolete** The intention was to use this value for indicating that the message contains some EMS content. However, the use of this value was incompatible with several SMSC implementations. Consequently, this value was later made obsolete in order to avoid any interoperability problem.
0x5F	**Return call message** A return call short message is used to indicate to the user that a call can be established to the originator address. As for the protocol identifier "replace short message type *n*," a return call message overwrites the one which had been previously stored in the mobile station (only if it has the same `TP-PID` and same originator address).
0x60...0x7B	**Reserved** (28 values)
0x7C	**ANSI-136 R-DATA** Messages with such a protocol identifier are transmitted to the (U)SIM.
0x7D	**ME data download** This protocol identifier indicates that the associated message is to be handled directly by the ME. Messages with this protocol identifier belong to class 1.
0x7E	**ME de-personalization short message** Messages with this protocol identifier instruct the ME to de-personalize according to the content of the message. Messages associated with this protocol identifier belong to class 1.
0x7F	**(U)SIM data download** Messages with such a protocol identifier are transmitted to the (U)SIM.
0x80...0xBF	**Reserved** (64 values)
0xC0...0xFF	**SMSC specific use** This range of protocol identifier values is left for service center specific use.

manipulated by applications. Information saved in this memory is persistently stored even if the mobile station has been powered off.

- *Random Access Memory* (RAM): this memory contains data manipulated by applications. Information stored in this memory is lost when the mobile station is powered off.

The storage structure of a SIM card is based on a hierarchy of folders and files. The root folder is known as the Master File (MF), a normal folder is known as a Dedicated File (DF), and a file is known as an Elementary File (EF).

Four elementary files are used for storing SMS settings, user preferences, and messages in the SIM. These SMS related elementary files are stored in the folder $DF_{TELECOM}$ as shown in Figure 3.12. The four elementary files, as defined in [3GPP-51.11], are described in Table 3.6.

Table 3.6 SMS-related SIM elementary files

Elementary file	Description
EF_{SMS}	**Storage of a message segment** This file can contain several records, each representing a message segment. A record contains the SMSC address followed by the message segment TPDU. Each record also indicates one of the following message status: • message received by the mobile station from network, message read • message received by the mobile station from network, message not read • message originated by the mobile station, message to be sent • message originated by the mobile station, message sent. If the message is originated by the mobile station and has been sent, then the corresponding record also indicates whether or not a status report was requested. If a status report was requested, then the record indicates whether or not the status report has been received and optionally refers to the corresponding EF_{SMSR} record.
EF_{SMSS}	**Storage of the status of the service** This elementary file indicates the status of the service. In this file, a flag is set if a message has been rejected because the SIM capacity for storing messages has exceeded. The file also contains the last message reference used for uniquely identifying messages sent by the mobile station.
EF_{SMSR}	**Storage of a status report for a message segment** Records in this elementary file represent status reports corresponding to message segments stored in the SIM. Note that this elementary file is seldom supported in existing commercial implementations.
EF_{SMSP}	**Storage of SMS parameters** This elementary file is used for storing the default values for the following parameters: • message recipient address • SMSC address • message protocol identifier • message data coding scheme • message validity period.

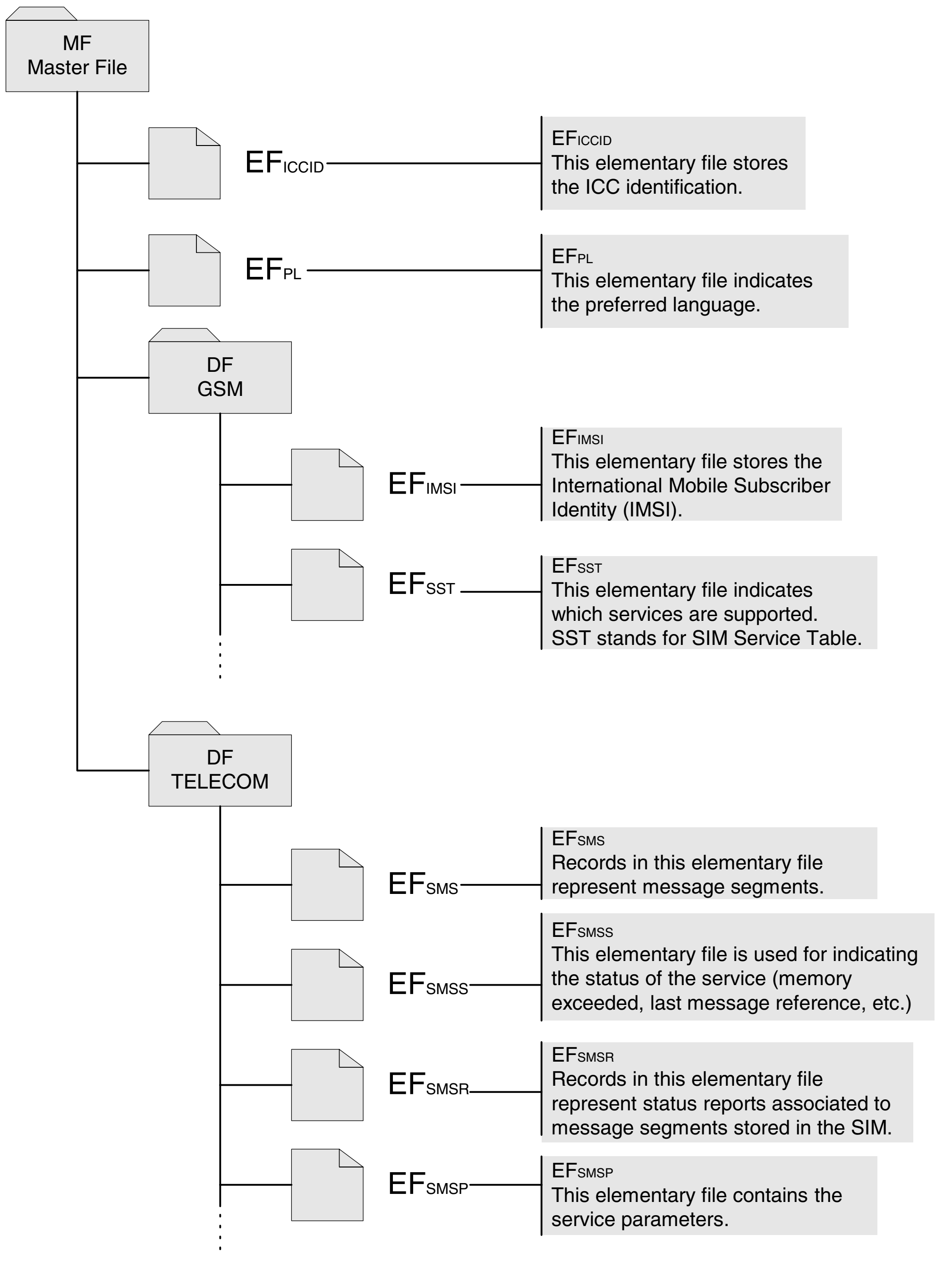


Figure 3.12 SIM storage

The partial SIM structure for the storage of SMS user settings and messages is shown in Figure 3.12.

Note that not all service parameters can be stored using the elementary file EF_{SMSP}. For instance, the default value for the Email gateway and the default setting for the request for a status report cannot be saved in standard SIM elementary files. To cope with this limitation, default values to be assigned to these parameters are sometimes stored directly in the ME memory (e.g., flash memory). This means that these parameters cannot be automatically retrieved if the subscriber inserts his/her SIM in another handset.

3.9 Message Submission

In the context of SMS, the term *submission* refers to the transfer of a message segment from the originator SME to the serving SMSC. Figure 3.13 shows interactions between an originator SME and an SMSC for the submission of a message segment.

Once the message segment has been successfully received by the serving SMSC, the originator SMSC queries the HLR in order to route forward the message segment towards the recipient SME. At the transfer layer, a message segment is conveyed as part of a TPDU of type `SMS-SUBMIT`. The TPDU can contain the following parameters:

- Message type (`SMS-SUBMIT`)
- Request for rejecting duplicated messages
- Message validity period
- Request for reply path
- Request for a status report
- Message reference
- Address of the recipient SME
- Protocol identifier
- Data coding scheme
- User data header
- User data (with associated length).

Upon receipt of the message by the serving SMSC, the SMSC provides a submission report to the originator SME. Two types of reports can be provided: a *positive submission report* for a successful submission or a *negative submission report* for a failed submission. If the submission report is not received after a given period of time, then the originator SME concludes that the message submission has failed.

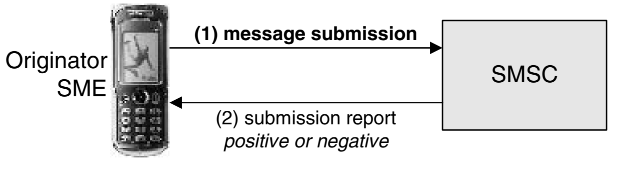


Figure 3.13 Message submission

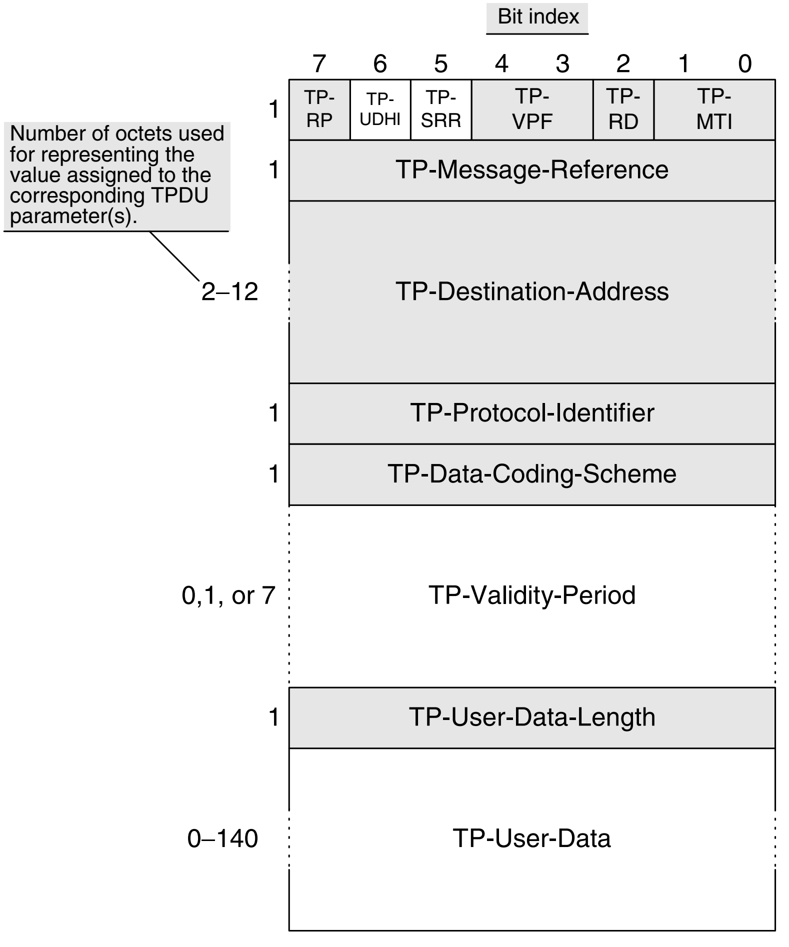


Figure 3.14 TPDU layout/type SMS-SUBMIT

3.9.1 TPDU Layout

A transfer layer, the TPDU of type `SMS-SUBMIT`, has the layout shown in Figure 3.14.

In this chapter, a specific graphical convention is used for the representation of TPDU layouts. Mandatory parameters are represented by gray-shaded boxes and optional parameters are represented by white boxes.

3.9.2 TPDU Parameters

The TPDU of type `SMS-SUBMIT` contains the parameters listed in Table 3.7.

The SMS standard defines three representations for numeric values and alphanumeric values assigned to TPDU parameters. These representations (integer representation, octet representation, and semi-octet representation) are defined in Appendix B.

Table 3.7 Message submission/TPDU parameters

Abbreviation	Reference	P	R	Description
TP-MTI	TP-Message-Type-Indicator	●	2 bits	Message type (bits 0 and 1 of first octet) bit 1 bit 0 Message Type 0 0 SMS-DELIVER 0 0 SMS-DELIVER-REP 1 0 SMS-STATUS-REP 1 0 SMS-COMMAND **0 1 SMS-SUBMIT** 0 1 SMS-SUBMIT-REP
TP-RD	TP-Reject-Duplicates	●	1 bit	Indication of whether the SMSC shall accept or reject duplicated message segments. A message segment is a duplicate of an original message segment if it has the same TP-MR, TP-DA, and TP-OA.
TP-VPF	TP-Validity-Period-Format	●	2 bits	Presence and format of the TP-VP field (bits 3 and 4 of first octet). bit 4 bit 3 0 0 TP-VP not present. 0 1 TP-VP - enhanced format. 1 0 TP-VP - relative format. 1 1 TP-VP - absolute format.
TP-RP	TP-Reply-Path	●	1 bit	Reply path (bit 7 of first octet) bit 7 at 0: reply-path is not set. bit 7 at 1: reply-path is set.
TP-UDHI	TP-User-Data-Header-Indicator	○	1 bit	Presence of a user data header in the user data part (bit 6 of first octet). bit 6 at 0: no user data header. bit 6 at 1: a user data header is present.
TP-SRR	TP-Status-Report-Request	○	1 bit	Request for a status report bit 5 at 0: no status report requested. bit 5 at 1: a status report is requested.
TP-MR	TP-Message-Reference	●	1 octet Integer rep.	Message segment reference number in the range 0...255 (decimal values).
TP-DA	TP-Destination-Address	●	2–12 octets	The destination address identifies the originating SME. The address format is defined in Section 3.9.6.
TP-PID	TP-Protocol-Identifier	●	1 octet	Protocol identifier as defined in Section 3.7.7.
TP-DCS	TP-Data-Coding-Scheme	●	1 octet	Data coding scheme as defined in Section 3.7.
TP-VP	TP-Validity-Period	○	1 octet or 7 octets	Validity period identifies the time from when the message is no longer valid.
TP-UDL	TP-User-Data-Length	●	1 octet Integer rep.	The user data length is expressed in septets (GSM 7-bit default alphabet) or octets (UCS2 or 8-bit encoding).
TP-UD	TP-User-Data	○	TP-DCS dependent	The user data and user data header are defined in Section 3.15.

In the table field names, P stands for Provision and R for Representation. In the table body, ● stands for Mandatory whereas ○ stands for Optional.

3.9.3 Rejection of Duplicates

It sometimes happens that a submission report gets lost. In this case, the originator SME has no means to determine if the message, for which the submission report has been lost, has been successfully submitted to the serving SMSC or not. If the originator SME re-transmits the message, while the first submission attempt was successful, the message will be transmitted twice to the message recipient. To avoid this situation, the originator SME has the ability to inform the SMSC that a previous submission was attempted for the message being submitted. In this case, if the SMSC detects that the previous submission had been successful, then the message is automatically discarded and not transmitted to the recipient SME. This ensures that the recipient SME receives the message only once.

Two parameters are associated with this ability to reject duplicates. The first parameter is the `TP-Reject-Duplicates`. This Boolean flag is set to 1 (true) for the new submission attempt and set to 0 (false) otherwise. The second parameter is the `TP-Message-Reference` which allows the SMSC to identify that the message, for which the `TP-Reject-Duplicates` is set, has already been successfully submitted.

3.9.4 Validity Period

The validity period of a message indicates the time after which the message content is no longer valid. The value assigned to the `TP-Validity-Period` parameter can take three different forms (as indicated by the field `TP-Validity-Period-Format`):

- **Relative format** (1 octet, integer representation). The value assigned to the `TP-Validity-Period` parameter in a relative format defines the length of the validity period starting from the time the message was received by the serving SMSC. The representation of the value assigned to the `TP-Validity-Period` parameter is as follows:

`TP-validity-period`	Validity period value
0–143	(`TP-Validity-Period`) × 5 minutes
144–167	(12 hours + (`TP-Validity-Period` – 143) × 30 minutes)
168–196	(`TP-Validity-Period` – 166) × 1 day
197–255	(`TP-Validity-Period` – 192) × 1 week

- **Absolute format** (7 octets, semi-octet representation). The value assigned to the `TP-Validity-Period` parameter, in an absolute format, defines the date when the validity period terminates. The 7-octet value is an absolute time representation as defined in Section 3.9.5.
- **Enhanced format** (7 octets). The first octet of the 7-octet `TP-Validity-Period`, in the enhanced format, indicates how the following 6 octets are used. The presence of all octets is mandatory even if all of them are not used. The first octet is structured as shown in Figure 3.15.

Note that the value assigned to the `TP-Validity-Period` parameter is always expressed with either 1 octet or 7 octets, depending on its format. Any reserved or unused bit is set to 0.

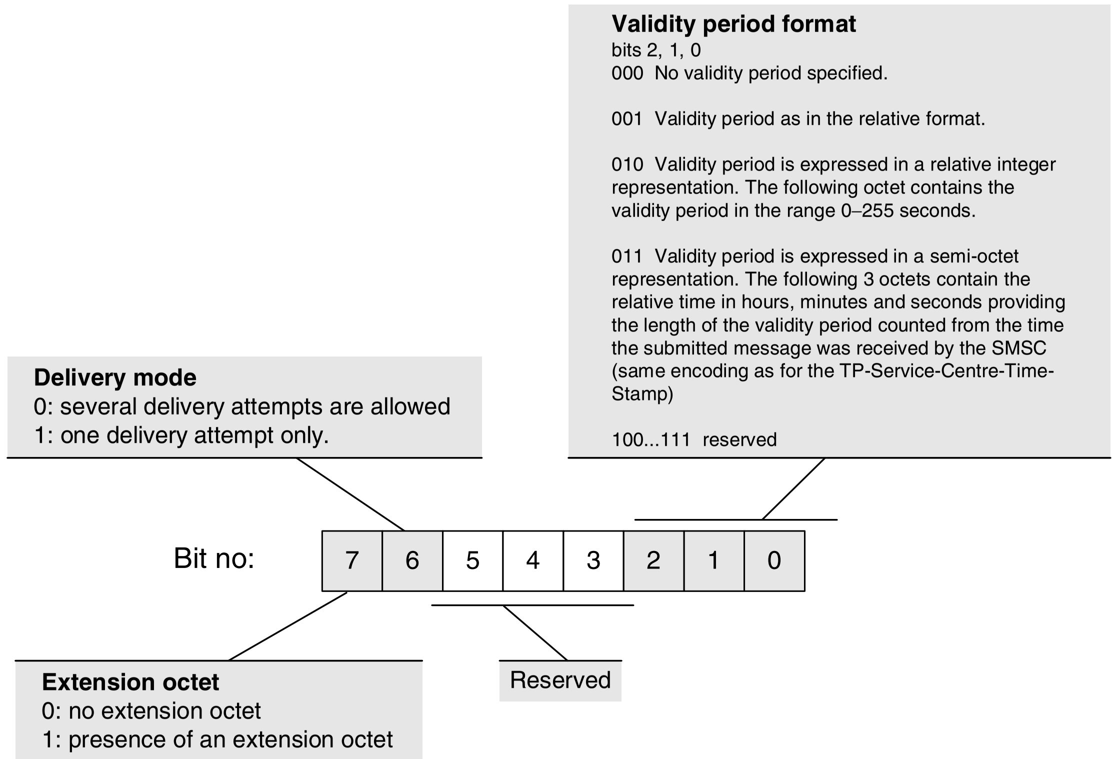


Figure 3.15 `TP-Validity-Period` in the enhanced format

3.9.5 Absolute Time Representation

Values assigned to several TPDU parameters represent an absolute time definition. This is the case for the `TP-Validity-Period`, `TP-Service-Center-Time-Stamp`, and the `TP-Discharge-Time`. For these parameters, the absolute time representation is decomposed into a sequence of time-related parameters, as described in Figure 3.16, which shows the absolute time 23rd December 01, 9:53:42 AM, GMT + 1 hour. Note that the time zone is expressed in quarters of an hour.

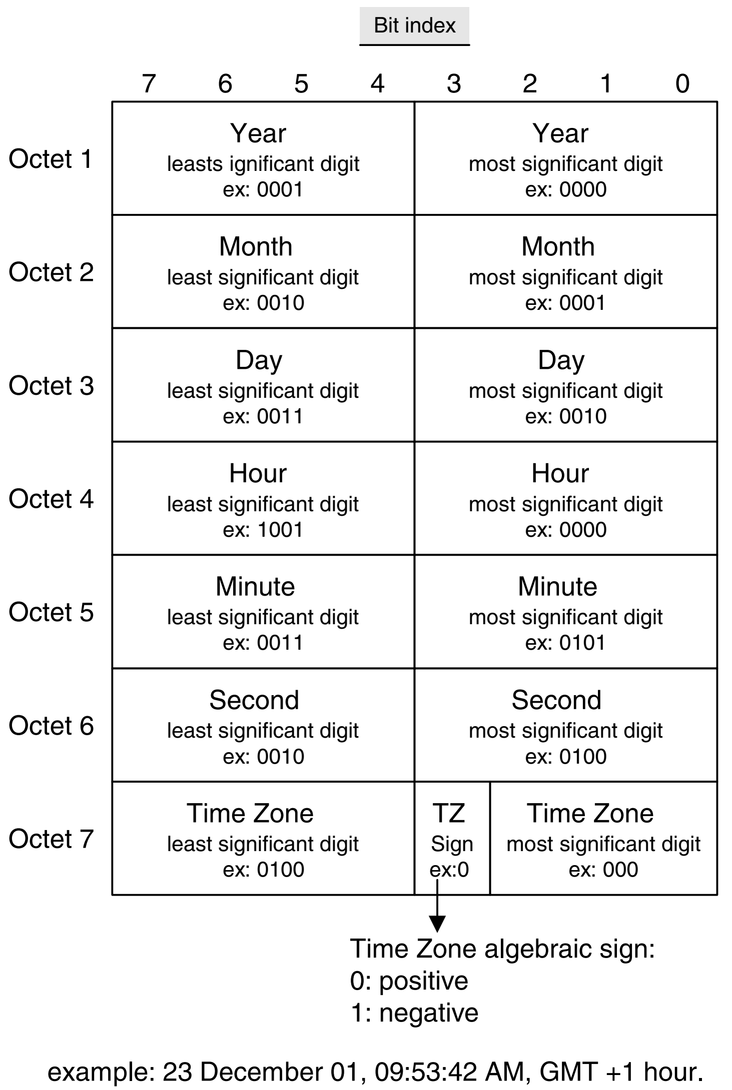


Figure 3.16 Absolute time definition

3.9.6 Destination Address

The value assigned to the `TP-Destination-Address` parameter represents the address of the recipient SME. This value is formatted as shown in the following section.

3.9.7 SME Addressing

Values assigned to the following parameters represent SME addresses:

- `TP-Destination-Address`
- `TP-Recipient-Address`
- `TP-Originator-Address`.

An SME address is decomposed into the following four sub-parameters:

- Address length (represents the number of useful semi-octets in the address value sub-parameter, the maximum length is 20 semi-octets)
- Type of number
- Numbering plan identification
- Address value.

The values assigned to the three addressing parameters are formatted as shown in Figure 3.17.

The values listed in Table 3.8 can be assigned to the `type-of-number` sub-parameter.

The values listed in Table 3.9 can be assigned to the `numbering-plan-identification` sub-parameter.

At the transfer layer, SMS does not offer the group sending feature which consists of submitting one message addressed to several recipients by submitting one message only to the SMSC. However, this feature is sometimes emulated at the application layer at the cost of submitting one message to the SMSC for each recipient.

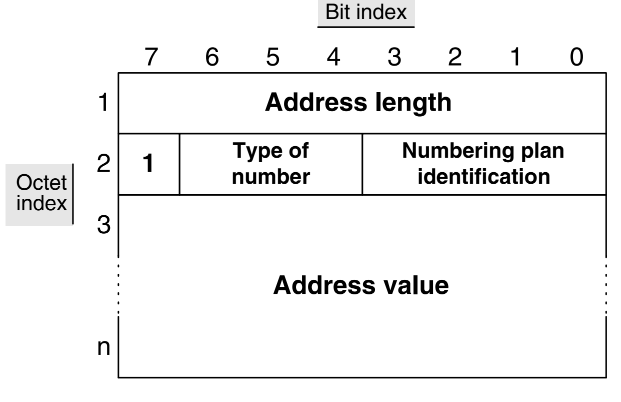


Figure 3.17 SMS addressing

Table 3.8 SMS addressing/type of number

Type-of-number			
Bit 6	Bit 5	Bit 4	Description
0	0	0	Unknown (address sub-parameters are organized according to the network dialing plan)
0	0	1	International number
0	1	0	National number
0	1	1	Network specific number (administration/service number specific to the serving network)
1	0	0	Subscriber number
1	0	1	Alphanumeric (coded in GSM 7-bit default alphabet)
1	1	0	Abbreviated number
1	1	1	Reserved

Table 3.9 SMS addressing/numbering-plan-identification

Numbering-plan-identification				
Bit 3	Bit 2	Bit 1	Bit 0	Description
0	0	0	0	Unknown
0	0	0	1	ISDN/telephone numbering plan
0	0	1	1	Data numbering plan (X.121)
0	1	0	0	Telex numbering plan
0	1	0	1	SMSC specific plan (external SMEs attached to the service center)
0	1	1	0	SMSC specific plan (external SMEs attached to the service center)
1	0	0	0	National numbering plan
1	0	0	1	Private numbering plan
1	0	1	0	ERMES numbering plan

All other values that can be assigned to this sub-parameter are reserved.

3.10 Message Submission Report

After the submission of a message segment from an originator SME to the serving SMSC, the SMSC acknowledges the submission by sending a report back to the originator SME. This report indicates the status of the submission. A positive submission report is sent back if the submission was successful, otherwise a negative submission report is generated. Interactions between the SME and the SMSC are shown in Figure 3.18.

Note that, with existing network configurations, submission reports are not always used. Instead, the acknowledgment of a message submission is often limited to a lower layer confirmation (relay layer).

If provided, the submission report is conveyed in the form of a TPDU of type `SMS-SUBMIT-REPORT` at the transfer layer.

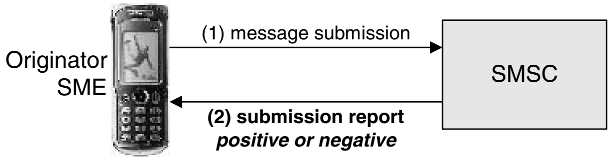


Figure 3.18 Submission report

3.10.1 Positive Submission Report

The positive submission report can contain several of the following parameters:

- Message type (`SMS-SUBMIT-REPORT`)
- Parameter indicator (presence of protocol identifier, data coding scheme, and user data length)
- Protocol identifier
- Data coding scheme
- Service center time stamp (time at which the SMSC received the associated message)
- User data header
- User data (with associated length).

Upon receipt of the submission report, the originator SME may indicate to the subscriber whether or not the submission was successful. If the submission was not successful, then the originator may request the subscriber to modify the message in order to re-attempt the message submission.

After message submission, if the originator SME does not receive a submission report from the serving SMSC, then the SME can conclude that either

- the message submission has failed, or
- the submission report has been lost.

In this situation, the originator SME may attempt another message submission. For this purpose, the originator SME can indicate for the new message submission that the message had already been submitted previously. This notice allows the serving SMSC to discard the newly submitted message if the first submission attempt was successful. The rejection of duplicate messages is described in Section 3.9.3.

Figure 3.19 shows the TPDU layout of a positive submission report. The positive submission report TPDU can contain several of the parameters listed in Table 3.10.

3.10.2 Negative Submission Report

Upon reception of a message segment, the serving SMSC may not be able to route forward the message (message badly formatted, SMSC busy, etc.). In this situation, the SMSC sends a negative submission report back to the originator SME. At the transfer layer, a negative

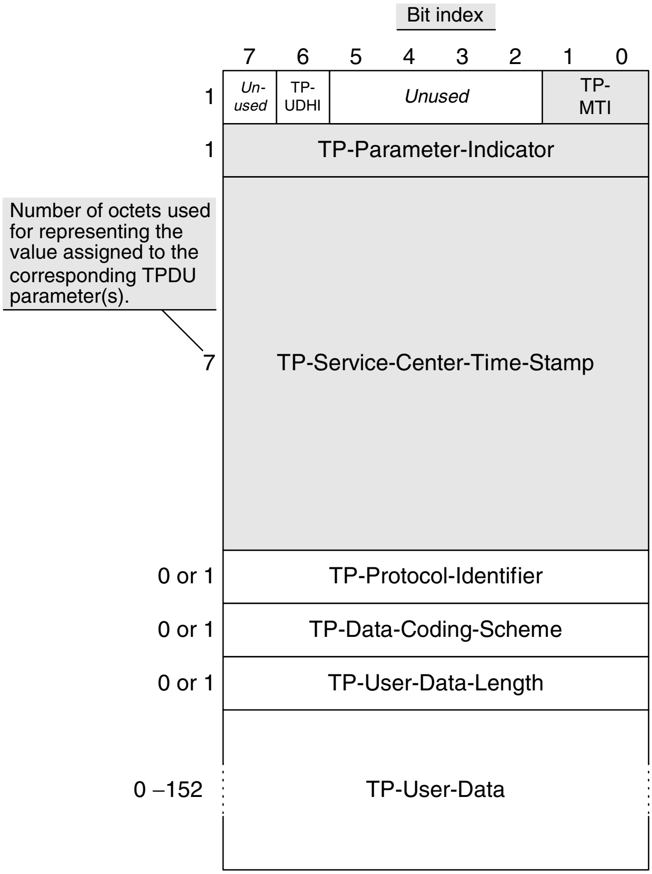


Figure 3.19 Positive submission report/layout

submission report is transported as a TPDU of type `SMS-SUBMIT-REPORT`. This TPDU can contain several of the following parameters:

- Message type (`SMS-SUBMIT-REPORT`)
- Parameter indicator (presence of protocol identifier, data coding scheme, and user data length)
- Protocol identifier
- Failure cause
- Data coding scheme
- Service center time stamp (time at which the SMSC received the associated message)
- User data header
- User data (with associated length).

Figure 3.20 shows the TPDU layout of the negative submission report.

Table 3.10 Positive submission report/TPDU parameters

Abbreviation	Reference	P	R	Description
TP-MTI	TP-Message-Type-Indicator	●	2 bits	Message type (bits 0 and 1 of first octet) bit 1 bit 0 Message Type 0 0 SMS-DELIVER 0 0 SMS-DELIVER-REP 1 0 SMS-STATUS-REP 1 0 SMS-COMMAND 0 1 SMS-SUBMIT **0 1 SMS-SUBMIT-REP**
TP-UDHI	TP-User-Data-Header-Indicator	○	1 bit	Presence of a user data header in the user data part (bit 6 of first octet). bit 6 at 0: no user data header. bit 6 at 1: a user data header is present.
TP-PI	TP-Parameter-Indicator	●	1 octet	Presence of TP-PID, TP-DCS and TP-UDL fields. The format of the TP-Parameter-Indicator is defined in Section 3.10.3.
TP-SCTS	TP-Service-Center-Time-Stamp	●	7 octets	Service center time stamp represents the time the SMSC received the message. The structure of the SMSC time stamp is defined in Section 3.10.4.
TP-PID	TP-Protocol-Identifier	○	1 octet	Protocol identifier as defined in Section 3.7.7.
TP-DCS	TP-Data-Coding-Scheme	○	1 octet	Data coding scheme as defined in Section 3.7.
TP-UDL	TP-User-Data-Length	○	1 octet Integer rep.	The user data length is expressed in septets (GSM 7-bit default alphabet) or octets (UCS2 or 8-bit encoding).
TP-UD	TP-User-Data	○	TP-DCS dependent	The user data and user data header are defined in Section 3.15.

In the table field names, P stands for Provision and R for Representation. In the table body, ● stands for Mandatory whereas ○ stands for Optional.

The different reasons for which the serving SMSC can acknowledge a message submission negatively are described in Table 3.11 (corresponding reason identification to be assigned to the TP-Failure-Cause parameter.

The negative submission report TPDU can contain several of the parameters listed in Table 3.12.

3.10.3 Parameter Indicator

The parameter indicator informs whether or not the following parameters are present in the TPDU: TP-Protocol-Identifier, TP-Data-Coding-Scheme, and TP-User-Data-Length. The structure of this parameter is shown in Figure 3.21. If the TP-User-Data-Length is not present in the TPDU, then the TP-User-Data is not present either.

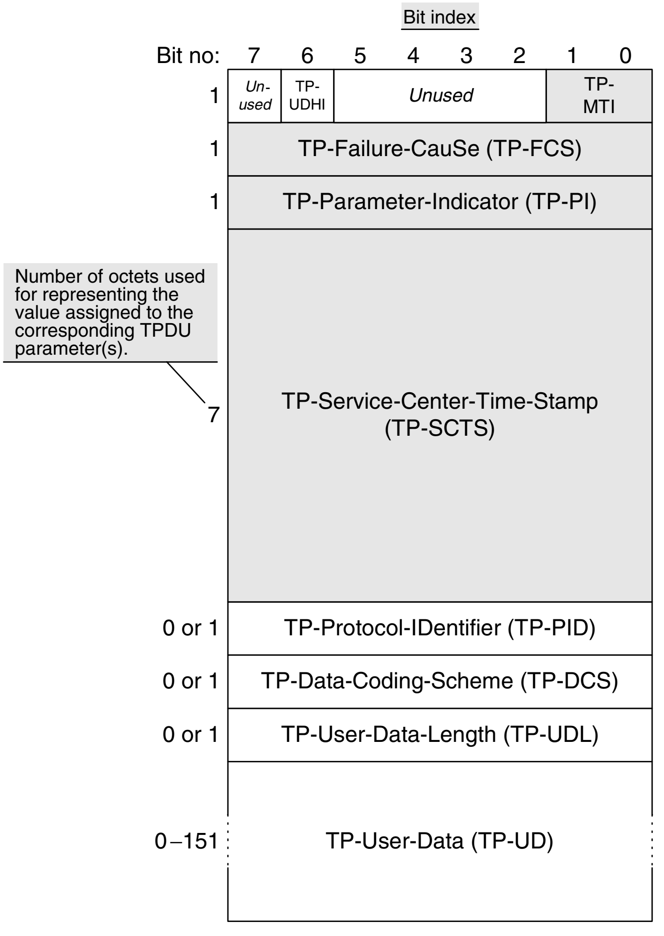


Figure 3.20 Negative submission report/layout

3.10.4 Service Center Time Stamp

For a positive or negative report, the service center time stamp parameter (`TP-Service-Center-Time-Stamp`) indicates the time at which the associated message was received by the serving SMSC. The value assigned to this parameter is formatted in an absolute time representation as defined in Section 3.9.5.

3.11 Message Delivery

In the context of SMS, the term *delivery* refers to the transfer of a message segment from the serving SMSC to the recipient SME as shown in Figure 3.22. If the recipient SME is not available for the delivery of the message segment, then the SMSC stores the message temporarily. The SMSC attempts to deliver the message until a delivery report is received from the recipient SME or until the message validity period expires.

Table 3.11 Negative submission report/failure causes

Reason Id. (hex)	Description
0x80	Telematic interworking not supported.
0x81	Short message Type 0 not supported.
0x82	The short message cannot be replaced.
0x8F	Unspecified `TP-PID` error.
0x90	Data coding scheme (alphabet) not supported.
0x9F	Unspecified `TP-DCS` error.
0xA0	Command cannot be executed.
0xA1	Command not supported.
0xAF	Unspecified `TP-Command` error.
0xB0	TPDU not supported.
0xC0	SMSC busy.
0xC1	No SMSC subscription.
0xC2	SMSC system failure.
0xC3	Invalid SME address.
0xC4	Destination SME barred.
0xC5	Message rejected – duplicate message.
0xC6	`TP-Validity-Period-Format` not supported.
0xC7	`TP-Validity-Period` not supported.

Upon receipt of a negative or positive report or upon message deletion, the serving SMSC may send a status report back to the originator SME (step 3). The status report is generated only if the originator SME requested it during message submission.

At the transfer layer, the message is delivered in the form of a TPDU of type `SMS-DELIVER`. The TPDU can contain several of the following parameters:

- Message type (`SMS-DELIVER`)
- Indication that there are more messages to be received
- Request for reply path
- Request for a status report
- Address of the originator SME
- Protocol identifier
- Data coding scheme
- Service center time stamp (time at which the SMSC received the message)
- User data header
- User data (with associated length).

Upon receipt of the message, the recipient SME provides a delivery report back to the serving SMSC. The delivery report indicates the status of the message delivery. Two types of reports can be provided: a *positive delivery report* for a successful message delivery or a *negative delivery report* for a failed delivery.

If the delivery report is not received after a given period of time, then the serving SMSC concludes that the message delivery has failed and may try to re-transmit the message later.

Table 3.12 Negative submission report/TPDU parameters

Abbreviation	Reference	P	R	Description
TP-MTI	TP-Message-Type-Indicator	●	2 bits	Message type (bits 0 and 1 of first octet) bit 1 / bit 0 / Message Type 0 0 SMS-DELIVER 0 0 SMS-DELIVER-REP 1 0 SMS-STATUS-REP 1 0 SMS-COMMAND 0 1 SMS-SUBMIT **0 1 SMS-SUBMIT-REP**
TP-UDHI	TP-User-Data-Header-Indicator	○	1 bit	Presence of a user data header in the user data part (bit 6 of first octet). bit 6 at 0: no user data header. bit 6 at 1: a user data header is present.
TP-FCS	TP-Failure-Cause	●	1 octet	Reason for submission failure (see Table 3.11). Values in the ranges 0x00...0x7F, 0x83...0x8E, 0x92...0x9E, 0xA2...0xAE, 0xB1...0xBF, 0xC8...0xCF, 0xD6...0xDF are reserved. The range 0xE0...0xFE contains values specific to an application. The value FF is used for an unspecified cause.
TP-PI	TP-Parameter-Indicator	●	1 octet	Presence of TP-PID, TP-DCS and TP-UDL fields. The format of the TP-Parameter-Indicator is defined in Section 3.10.3.
TP-SCTS	TP-Service-Center-Time-Stamp	●	7 octets	Service center time stamp represents the time the SMSC received the message. The structure of the SMSC time stamp is defined in Section 3.10.4.
TP-PID	TP-Protocol-Identifier	○	1 octet	Protocol identifier as defined in Section 3.7.7.
TP-DCS	TP-Data-Coding-Scheme	○	1 octet	Data coding scheme as defined in Section 3.7.
TP-UDL	TP-User-Data-Length	○	1 octet Integer rep.	The user data length is expressed in septets (GSM 7-bit default alphabet) or octets (UCS2 or 8-bit encoding).
TP-UD	TP-User-Data	○	TP-DCS dependent	The user data and user data header are defined in Section 3.15.

In the table field names, P stands for Provision and R for Representation. In the table body, ● stands for Mandatory whereas ○ stands for Optional.

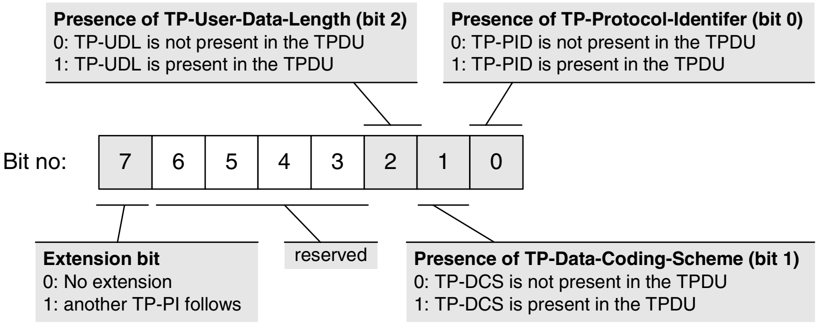


Figure 3.21 `TP-parameter-indicator` structure

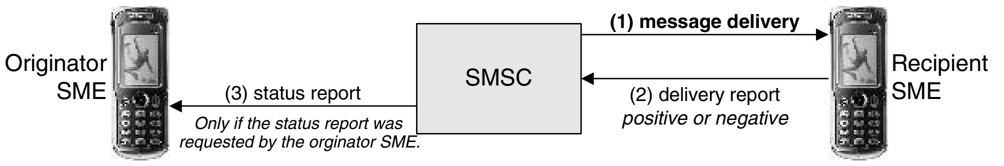


Figure 3.22 Message delivery

3.11.1 TPDU Layout

At the transfer layer, the message delivery TPDU has the layout shown in Figure 3.23.

3.11.2 TPDU Parameters

The message delivery TPDU can contain several of the parameters listed in Table 3.13.

3.11.3 Status Report Indicator

The status report indicator (`TP-Status-Report-Indicator` parameter) indicates whether or not the originator of the message requested a status report. The following values can be assigned to this 1-bit parameter:

- Value 0: no status report requested
- Value 1: a status report is requested.

3.11.4 Service Center Time Stamp

The service center time stamp (`TP-Service-Center-Time-Stamp` parameter) indicates the time at which the message has been received by the serving SMSC. The value assigned to this parameter is formatted in a time absolute representation as defined in Section 3.9.5.

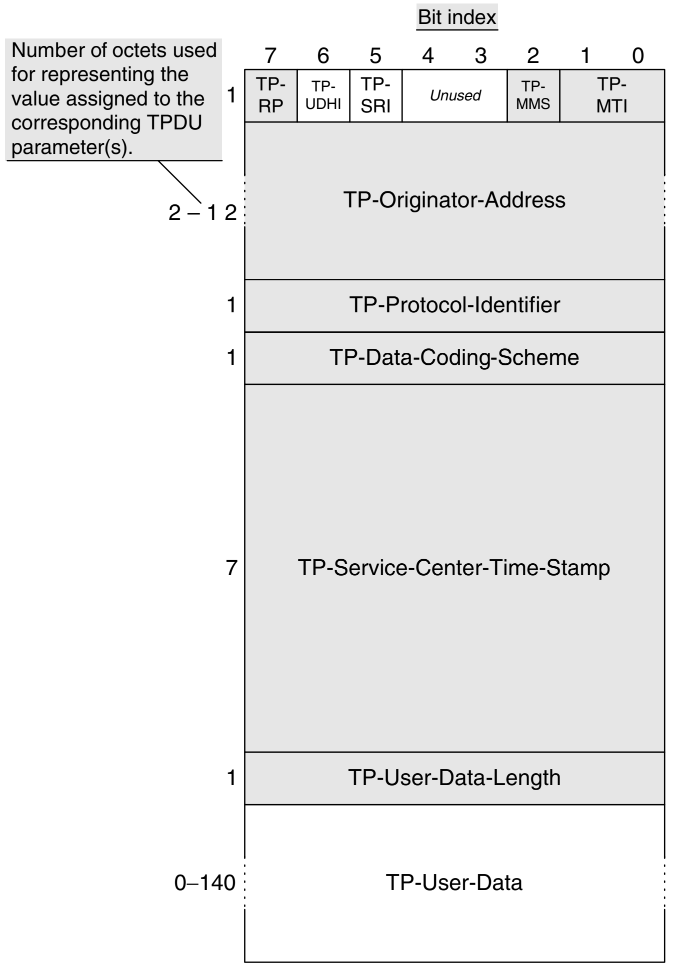


Figure 3.23 Message delivery/layout

3.12 Message Delivery Report

Upon delivery of a message from the serving SMSC to the recipient SME, the SME acknowledges the message delivery by sending back a delivery report to the serving SMSC. A positive delivery report is sent back if the delivery was successful, otherwise a negative delivery report is generated. The reception of the delivery report is necessary for the serving SMSC, which will then stop attempting to deliver the message to the recipient SME. If the originator SME requested a status report to be generated, then the serving SMSC generates the status report according to the delivery report received from the recipient SME as shown in Figure 3.24.

Note that with existing network configurations, delivery reports are not always used. Instead, the acknowledgment of a message delivery is often limited to a lower layer confirmation (relay layer).

Table 3.13 Message delivery/TPDU parameters

Abbreviation	Reference	P	R	Description
TP-MTI	TP-Message-Type-Indicator	●	2 bits	Message type (bits 0 and 1 of first octet) bit 1 bit 0 Message Type **0 0 SMS-DELIVER** 0 0 SMS-DELIVER-REP 1 0 SMS-STATUS-REP 1 0 SMS-COMMAND 0 1 SMS-SUBMIT 0 1 SMS-SUBMIT-REP
TP-MMS	TP-More-Messages-to-Send	●	1 bit	Presence of more messages to send at the SMSC (bit 2 of first octet). bit 2 at 0: more messages are waiting bit 2 at 1: no more messages are waiting
TP-RP	TP-Reply-Path	●	1 bit	Reply path (bit 7 of first octet) bit 7 at 0: reply-path is not set. bit 7 at 1: reply-path is set.
TP-UDHI	TP-User-Data-Header-Indicator	○	1 bit	Presence of a user data header in the user data part (bit 6 of first octet). bit 6 at 0: no user data header. bit 6 at 1: a user data header is present.
TP-SRI	TP-Status-Report-Indicator	○	1 bit	Indication that a status report is requested (bit 5 of the first octet). bit 5 at 0: no status report requested. bit 5 at 1: a status report is requested.
TP-OA	TP-Originator-Address	●	2...12 octes.	The originating address identifies the originating SME. The address format is defined in Section 3.9.7.
TP-PID	TP-Protocol-Identifier	●	1 octet	Protocol identifier as defined in Section 3.7.7.
TP-DCS	TP-Data-Coding-Scheme	●	1 octet	Data coding scheme as defined in Section 3.7.
TP-SCTS	TP-Service-Center-Time-Stamp	●	7 octets	Service center time stamp represents the time the SMSC received the message. The structure of the SMSC time stamp is defined in Section 3.10.4.
TP-UDL	TP-User-Data-Length	●	1 octet Integer rep.	The user data length is expressed in septets (GSM 7-bit default alphabet) or octets (UCS2 or 8-bit encoding).
TP-UD	TP-User-Data	○	TP-DCS dependent	The user data and user data header are defined in Section 3.15.

In the table field names, P stands for Provision and R for Representation. In the table body, ● stands for Mandatory whereas ○ stands for Optional.

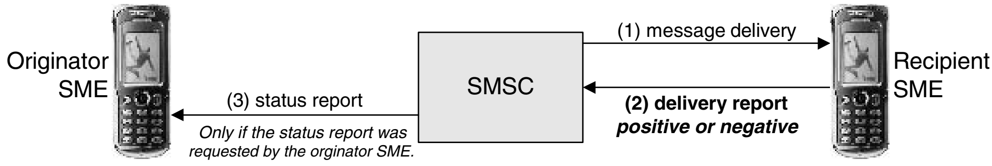


Figure 3.24 Delivery report

If provided, the delivery report is conveyed in the form of a TPDU of type `SMS-DELIVER-REPORT` at the transfer layer.

3.12.1 Positive Delivery Report

The positive delivery report TPDU can contain several of the following parameters:

- Message type (`SMS-DELIVER-REPORT`)
- Parameter indicator (presence of protocol identifier, data coding scheme, and user data length)
- Protocol identifier
- Data coding scheme
- User data header
- User data (with associated length).

At the transfer layer, a positive delivery report TPDU has the layout shown in Figure 3.25. A positive delivery report TPDU contains the parameters listed in Table 3.14.

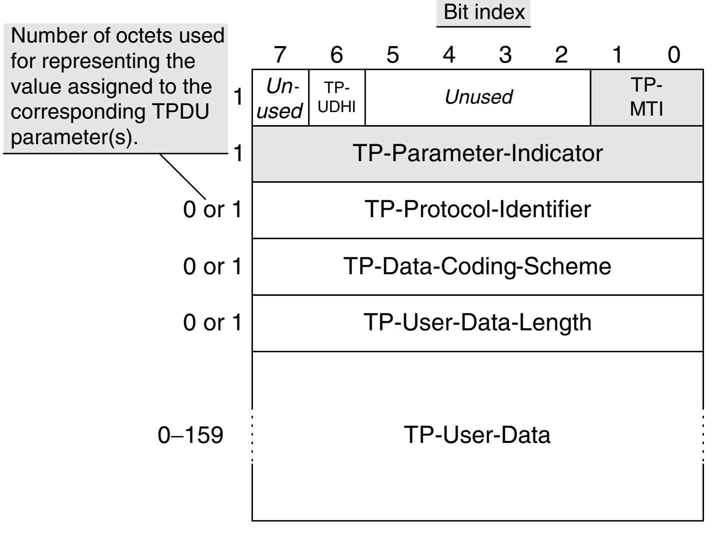


Figure 3.25 Positive delivery report/TPDU parameters

Table 3.14 Positive delivery report/TPDU parameters

Abbreviation	Reference	P	R	Description
TP-MTI	TP-Message-Type-Indicator	●	2 bits	Message type (bits 0 and 1 of first octet) bit 1 bit 0 Message Type 0 0 SMS-DELIVER **0 0 SMS-DELIVER-REP** 1 0 SMS-STATUS-REP 1 0 SMS-COMMAND 0 1 SMS-SUBMIT 0 1 SMS-SUBMIT-REP
TP-UDHI	TP-User-Data-Header-Indicator	○	1 bit	Presence of a user data header in the user data part (bit 6 of first octet). bit 6 at 0: no user data header. bit 6 at 1: a user data header is present.
TP-PI	TP-Parameter-Indicator	●	1 octet	Presence of TP-PID, TP-DCS, and TP-UDL fields. The format of the TP-Parameter-Indicator is defined in Section 3.10.3.
TP-PID	TP-Protocol-Identifier	○	1 octet	Protocol identifier as defined in Section 3.7.7.
TP-DCS	TP-Data-Coding-Scheme	○	1 octet	Data coding scheme as defined in Section 3.7.
TP-UDL	TP-User-Data-Length	○	1 octet Integer rep.	The user data length is expressed in septets (GSM 7-bit default alphabet) or octets (UCS2 or 8-bit encoding).
TP-UD	TP-User-Data	○	TP-DCS dependent	The user data and user data header are defined in Section 3.15.

In the table field names, P stands for Provision and R for Representation. In the table body, ● stands for Mandatory whereas ○ stands for Optional.

3.12.2 Negative Delivery Report

In some situations, the recipient SME is not able to handle the message correctly (message badly formatted, storage capacity exceeded, etc.). In order to inform the serving SMSC that the message cannot be handled, the receiving SME generates a negative delivery report. At the transfer layer, the negative delivery report is transported in the form of a TPDU of type SMS-DELIVER-REPORT. The TPDU can contain several of the following parameters:

- Message type (SMS-DELIVER-REPORT)
- Parameter indicator (presence of protocol identifier, data coding scheme, and user data length)
- Protocol identifier
- Failure cause
- Data coding scheme
- User data header
- User data (with associated length).

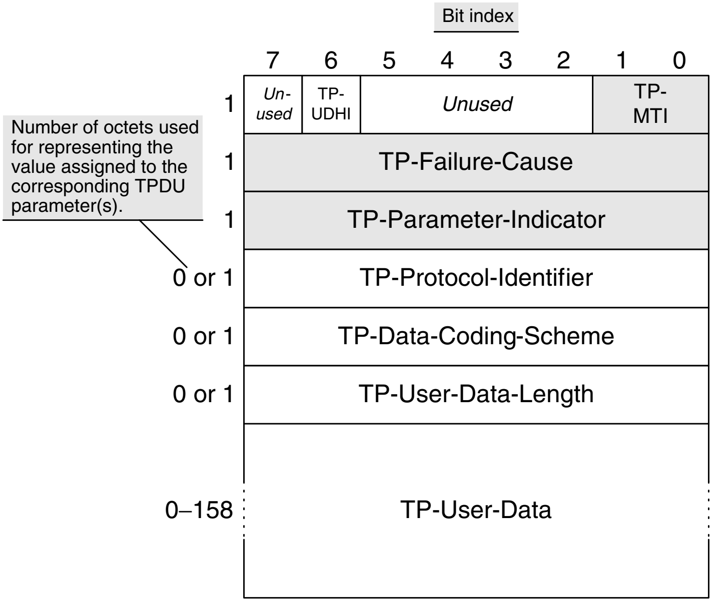


Figure 3.26 Negative delivery report/layout

At the transfer layer, a negative delivery report TPDU has the layout shown in Figure 3.26.

The different reasons for which the recipient SME can generate a negative delivery report are listed in Table 3.15 (corresponding reason identification to be assigned to the `TP-Failure-Cause` parameter).

The parameters listed in Table 3.16 are included in the TPDU of a negative delivery report.

Table 3.15 Negative delivery report/failure causes

Reason Id. (hex)	Description
0x81	Short message type 0 not supported.
0x82	The short message cannot be replaced.
0x8F	Unspecified `TP-PID` error.
0x91	Message class not supported.
0x9F	Unspecified `TP-DCS` error.
0xB0	TPDU not supported.
0xD0	(U)SIM SMS storage full.
0xD1	No SMS storage capability in (U)SIM.
0xD2	Error in MS.
0xD3	Memory capacity exceeded.
0xD4	(U)SIM application toolkit busy.
0xD5	(U)SIM data download error.
0xE0...0xFE	Application-specific errors.
0xFF	Unspecified error cause.

Other values are reserved.

Table 3.16 Negative delivery report/TPDU parameters

Abbreviation	Reference	P	R	Description
TP-MTI	TP-Message-Type-Indicator	●	2 bits	Message type (bits 0 and 1 of first octet) bit 1 bit 0 Message Type 0 0 SMS-DELIVER **0 0 SMS-DELIVER-REP** 1 0 SMS-STATUS-REP 1 0 SMS-COMMAND 0 1 SMS-SUBMIT 0 1 SMS-SUBMIT-REP
TP-UDHI	TP-User-Data-Header-Indicator	○	1 bit	Presence of a user data header in the user data part (bit 6 of first octet). bit 6 at 0: no user data header. bit 6 at 1: a user data header is present.
TP-FCS	TP-Failure-Cause	●	1 octet	Reason for delivery failure (see table above).
TP-PI	TP-Parameter-Indicator	●	1 octet	Presence of TP-PID, TP-DCS and TP-UDL fields. The format of the TP-Parameter-Indicator is defined in Section 3.10.3.
TP-PID	TP-Protocol-Identifier	○	1 octet	Protocol identifier as defined in Section 3.7.7.
TP-DCS	TP-Data-Coding-Scheme	○	1 octet	Data coding scheme as defined in Section 3.7.
TP-UDL	TP-User-Data-Length	○	1 octet Integer rep.	The user data length is expressed in septets (GSM 7-bit default alphabet) or octets (UCS2 or 8-bit encoding).
TP-UD	TP-User-Data	○	TP-DCS dependent	The user data and user data header are defined in Section 3.15.

In the table field names, P stands for Provision and R for Representation. In the table body, ● stands for Mandatory whereas ○ stands for Optional.

3.13 Status Report

Upon delivery of a message segment to a recipient SME, the serving SMSC may generate a status report and transfer it back to the originator SME. The status report is sent only if the originator SME requested it during the message submission. The serving SMSC generates the status report when the associated delivery report has been received from the recipient SME or when the message is discarded by the SMSC without delivery (e.g., validity period has expired). Figure 3.27 shows the three steps leading to the delivery of a status report.

A status report may also be generated upon receipt or execution of a message command (see next section). At the transfer layer, the status report is transported in the form of a TPDU of type SMS-STATUS-REPORT. The TPDU can contain several of the following parameters:

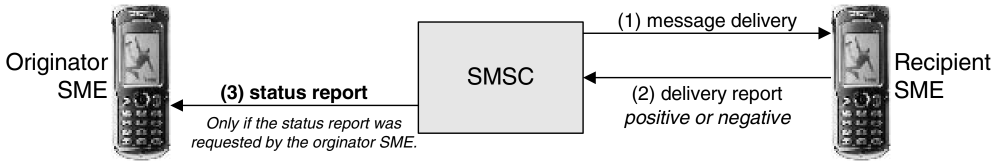


Figure 3.27 Status report

- Message type (SMS-STATUS-REPORT)
- Parameter indicator (presence of protocol identifier, data coding scheme, and user data length)
- Indication that there are more messages to be received
- Protocol identifier
- Status report qualifier
- Delivery status
- Discharge time
- Message reference (from original message)
- Recipient address
- Service center time stamp (from original message)
- Data coding scheme
- User data header
- User data (with associated length).

Upon receipt of the status report, the originator SME is able to identify the original message, to which the status report refers, by checking that the status report recipient address is equal to the destination address of the original message and that the status report message reference is equal to the original message reference. In addition, the originator SME may also check that the status report service center time stamp corresponds to the service center time stamp specified in the submit report of the original message. Note that the service center time stamp is not always provided by the serving SMSC for a message submission.

If the original message is stored in the SIM, then the status of the corresponding EF_{SMS} file is updated (message originated by the mobile station, message sent, and status report received). Additionally, a record may be created in the EF_{SMSR} file containing the status report.

If the original message has been deleted, then the originator SME may discard the corresponding status report or may alternatively present the message to the subscriber as a normal message (as if it had been delivered as a TPDU of type SMS-DELIVER).

The status of the delivery, conveyed by the status report, can take one of the values listed in Table 3.17 (to be assigned to the TP-Status parameter of the status report TPDU).

3.13.1 TPDU Layout

The layout of the status report TPDU is shown in Figure 3.28.

3.13.2 TPDU Parameters

The status report TPDU contains the parameters listed in Table 3.18.

Table 3.17 Status report codes

	Status Id. (hex)	Description
Short message transaction completed.	0x00	Message successfully received by the recipient SME.
	0x01	Message forwarded by the SMSC to the SME but the SMSC is unable to confirm delivery.
	0x02	Message has been replaced by the SMSC.
	0x10...0x1F	Values specific to each SMSC.
Temporary error, SC still trying to transfer the short message.	0x20	Congestion.
	0x21	SME busy.
	0x22	No response from SME.
	0x23	Service rejected.
	0x24	Quality of service not available.
	0x30...0x3F	Values specific to each SMSC.
Permanent error, SMSC is not making delivery attempts anymore.	0x40	Remote procedure error.
	0x41	Incompatible destination.
	0x42	Connection rejected by the SME.
	0x43	Not obtainable.
	0x44	Quality of service not available.
	0x45	No interworking available.
	0x46	Message validity period expired.
	0x47	Message deleted by originating SME.
	0x48	Message deleted by SMSC administration.
	0x49	Message does not exist (or the SMSC has no knowledge of the associated message).
	0x50...0x5F	Values specific to each SMSC.
Temporary error, SMSC is not making delivery attempts anymore.	0x60	Congestion.
	0x61	SME busy.
	0x62	No response from SME.
	0x63	Service rejected.
	0x64	Quality of service not available.
	0x65	Error in SME.
	0x70...0x7F	Values specific to each SMSC.

Unused status index values are reserved.

3.13.3 Discharge Time

If the associated message has been successfully delivered, then the discharge time represents the time at which the serving SMSC delivered the message. Otherwise, the discharge time represents the time at which the serving SMSC last attempted to deliver the message. The value assigned to this parameter (`TP-Discharge-Time`) is formatted as an absolute time representation as defined in Section 3.9.5.

3.14 Command

An originator SME can request the originator SMSC to execute a command. This is performed by sending a specific command message to the serving SMSC. The command can be a request for the generation of a status report by the serving SMSC, the deletion of a previously submitted message segment, etc. The command is typically not submitted from a

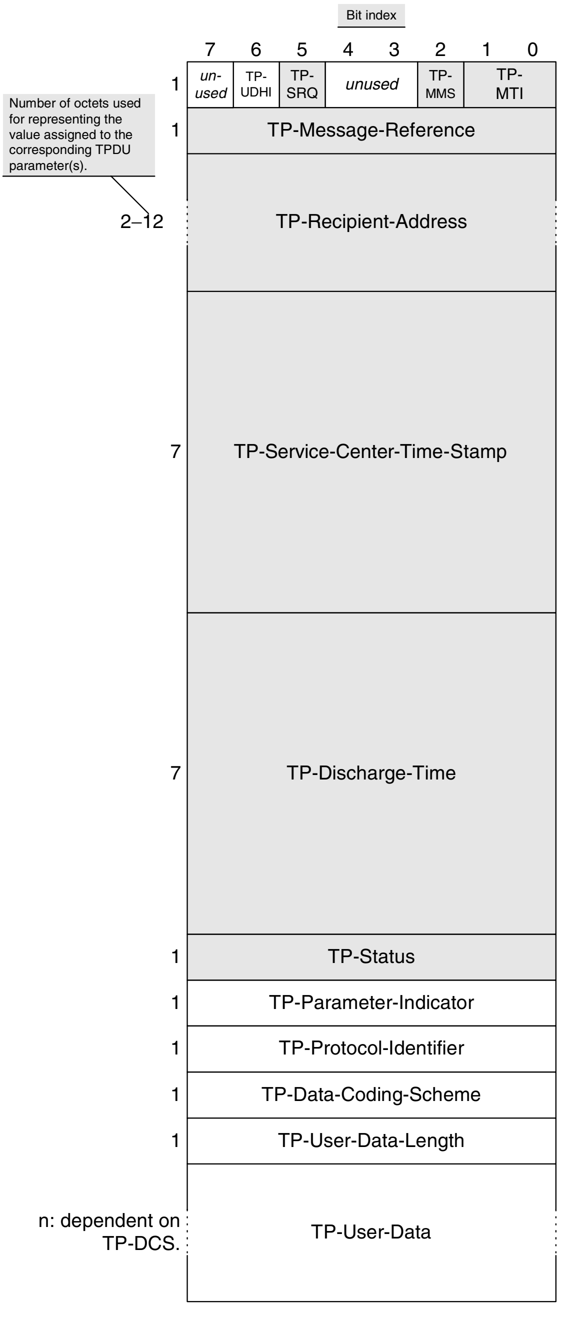


Figure 3.28 Status report/layout

Table 3.18 Status report/TPDU parameters

Abbreviation	Reference	P	R	Description
TP-MTI	TP-Message-Type-Indicator	●	2 bits	Message type (bits 0 and 1 of first octet) bit 1 bit 0 Message Type 0 0 SMS-DELIVER 0 0 SMS-DELIVER-REP **1 0 SMS-STATUS-REP** 1 0 SMS-COMMAND 0 1 SMS-SUBMIT 0 1 SMS-SUBMIT-REP
TP-MMS	TP-More-Messages-to-Send	●	1 bit	Presence of more messages to send at the SMSC (bit 2 of first octet). bit 2 at 0: more messages are waiting. bit 2 at 1: no more messages are waiting.
TP-SRQ	TP-Status-Report-Qualifier	●	1 bit	Indication whether the associated TPDU was a submitted SM or a command. bit 5 at 0: associated TPDU was a submitted SM. bit 5 at 1: associated TPDU was a command.
TP-UDHI	TP-User-Data-Header-Indicator	○	1 bit	Presence of a user data header in the user data part (bit 6 of first octet). bit 6 at 0: no user data header. bit 6 at 1: a user data header is present.
TP-MR	TP-Message-Reference	●	1 octet Integer rep.	Message segment reference number in the range 0...255
TP-RA	TP-Recipient-Address	●	2...12 octets	The recipient address identifies the SME which was identified by the TP-Destination-Address of the associated submitted message. The value assigned to this parameter is formatted as defined in Section 3.9.7.
TP-SCTS	TP-Service-Center-Time-Stamp	●	7 octets	Service center time stamp represents the time the SMSC received the message. The structure of the SMSC time stamp is defined below.
TP-DT	TP-Discharge-Time	●	7 octets	See description in Section 3.13.3.
TP-ST	TP-Status	●	1 octet	Status of the associated message or command as defined in table. Values in the ranges 0x03... 0x0F, 0x26... 0x2F, 0x4A...0x4F, 0x4F, 0x66...0x6F, 0x80...0xFF are reserved.
TP-PI	TP-Parameter-Indicator[1]	○	1 octet	Presence of TP-PID, TP-DCS, and TP-UDL fields. The format of the TP-Parameter-Indicator is defined in Section 3.10.3.

(Continued)

[1] Mandatory if any of the optional parameters following TP-Parameter-Indicator is present, otherwise optional.

Table 3.18 (*Continued*)

Abbreviation	Reference	P	R	Description
`TP-PID`	`TP-Protocol-Identifier`	○	1 octet	Protocol identifier as defined in Section 3.7.7.
`TP-DCS`	`TP-Data-Coding-Scheme`	○	1 octet	Data coding scheme as defined in Section 3.7.
`TP-UDL`	`TP-User-Data-Length`	○	1 octet Integer rep.	The user data length is expressed in septets (GSM 7-bit default alphabet) or octets (UCS2 or 8-bit encoding).
`TP-UD`	`TP-User-Data`	○	`TP-DCS` dependent	The user data and user data header are defined in Section 3.15.

In the table field names, P stands for Provision and R for Representation. In the table body, ● stands for Mandatory whereas ○ stands for Optional.

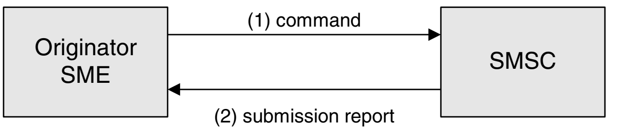


Figure 3.29 Command and submission report

mobile station. The execution of a command is usually requested by an application server (external SME) as shown in Figure 3.29.

The commands listed in Table 3.19 can be executed by the serving SMSC (command identification to be assigned to the `TP-Command-Type` parameter of the command TPDU).

The submission report (step 2) for a command has the same characteristics (layout/TPDU parameters) as those of a submission report for a message submission.

Table 3.19 Command identifiers

Command identifier (hex)	Description
0x00	Enquiry relating to a previously submitted message. With this command, a request is made for the generation of a status report.
0x01	Cancel status report request relating to a previously submitted message. No status report is to be generated for this command execution.
0x02	Delete a previously submitted message. No status report is to be generated for this command execution.
0x03	Enable status report request relating to a previously submitted message. No status report is to be generated for this command execution.
0xE0...0xFF	Values specific for each SMSC.

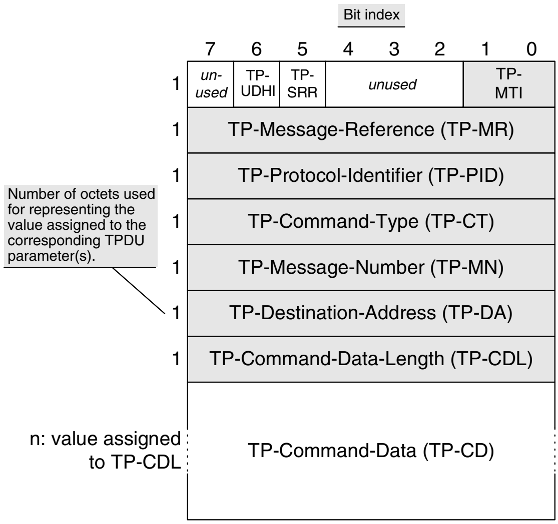


Figure 3.30 Command/layout

3.14.1 TPDU Layout

The layout of the command TPDU is shown in Figure 3.30.

3.14.2 TPDU Parameters

The command TPDU can contain several of the parameters listed in Table 3.20.

3.15 User Data Header and User Data

As shown in previous sections, the `TP-User-Data` (`TP-UD`) contains the text part of a message segment. Optionally, the `TP-UD` may also contain an 8-bit aligned User-Data-Header (UDH). The User-Data-Header is composed of a User-Data-Header-Length followed by a sequence of information elements. Information elements are used for the following purposes:

- *SMS control*: in this category, information elements contain some SMS control instructions such as concatenation information, application ports, SMSC control parameters, etc.
- *Basic and extended EMS objects*: in this category, information elements contain the definition of EMS objects such as melodies, pictures, animations, etc.

The structure of the `TP-User-Data` parameter is shown in Figure 3.31. The first octet of the User-Data-Header, called the User-Data-Header-Length (UDHL), indicates the length of the User Data Header. If the text is 7-bit encoded, then fill bits may be necessary between the User-Data-Header and the remaining part of the `TP-User-Data`. These fill bits ensure that the text, which follows the User-Data-Header, always starts on a septet boundary. This is

Table 3.20 SMS command/TPDU parameters

Abbreviation	Reference	P	R	Description
`TP-MTI`	`TP-Message-Type-Indicator`	●	2 bits	Message type (bits 0 and 1 of first octet) bit 1 bit 0 Message Type 0 0 SMS-DELIVER 0 0 SMS-DELIVER-REP 1 0 SMS-STATUS-REP **1 0 SMS-COMMAND** 0 1 SMS-SUBMIT 0 1 SMS-SUBMIT-REP
`TP-SRR`	`TP-Status-Report-Request`	○	1 bit	Request for a status report bit 5 at 0: no status report requested. bit 5 at 1: a status report is requested.
`TP-UDHI`	`TP-User-Data-Header-Indicator`	○	1 bit	Presence of a user data header in the user data part (bit 6 of first octet). bit 6 at 0: no user data header. bit 6 at 1: a user data header is present.
`TP-MR`	`TP-Message-Reference`	●	1 octet Integer rep.	Message segment reference number in the range 0...255.
`TP-PID`	`TP-Protocol-Identifier`	●	1 octet	Protocol identifier as defined in Section 3.7.7.
`TP-CT`	`TP-Command-Type`	●	1 octet	Type of command to be executed as defined in table. Values in range 0x04...0x1F are reserved.
`TP-MN`	`TP-Message-Number`	●	1 octet	Message number of the message on which the command is to be executed.
`TP-DA`	`TP-Destination-Address`	●	2...12 octets	The destination address identifies the originating SME. The address format is defined in Section 3.9.7.
`TP-CDL`	`TP-Command-Data-Length`	●	1 octet Integer rep.	Length of the `TP-Command-Data` in octets.
`TP-CD`	`TP-Command-Data`	○	Value of `TP-CDL` (in octets)	This parameter contains the user data.

In the table field names, P stands for Provision and R for Representation. In the table body, ● stands for Mandatory whereas ○ stands for Optional.

important to allow the oldest SMEs, that do not support the concept of User-Data-Header, to still be able to interpret the text part of the message.

3.15.1 Information Elements

An *Information Element* (IE), if recognized, is decoded and interpreted by the recipient SME. An IE can be an EMS element such as a command for text formatting, a melody, an

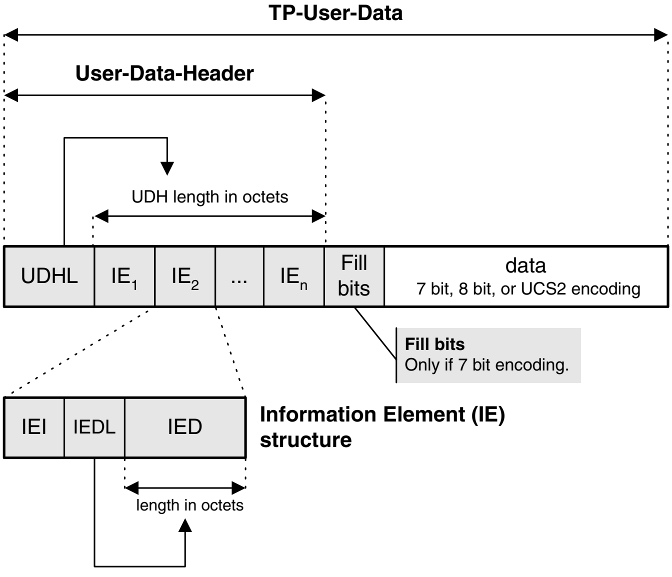


Figure 3.31 `TP-User-Data` structure

animation, or a picture. It can also be an information element for SMS control. The structure of an IE is designed to allow easy introduction of new IEs in the standard along with maintaining forward compatibility between heterogeneous SMEs. The structure of an IE is shown in Figure 3.32.

The *IE Identifier* (IEI) indicates the information element type. Possible values for the IE Identifier parameter are shown in Table 3.21. The *IE Data Length* (IEDL) indicates the length of the *IE Data* (IED) in octets. The IE data represents the information element payload. For an IE representing a bitmap picture, the payload is the picture position in the text along with the picture dimensions and picture bitmap. For an IE representing

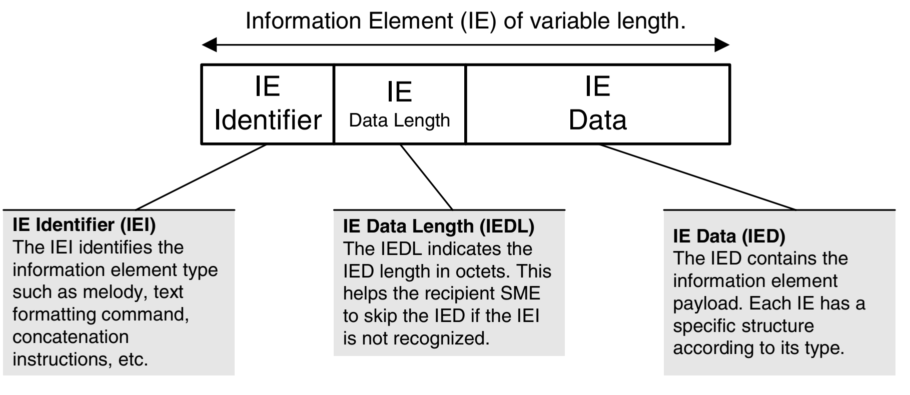


Figure 3.32 Information element structure

Table 3.21 Information element identifiers

IEI value	Description	from Rel.	Type	Segment repeat	Message repeat	See Section
0x00	Concatenated short message, 8 bit reference number	99	SMS	No	Yes	3.15.2
0x01	Special SMS message indication	99	SMS	Yes	Yes	3.15.3
0x02...0x03	Reserved	N/A	N/A	N/A	N/A	–
0x04	Application port addressing scheme 8 bit address	99	SMS	No	Yes	3.15.4
0x05	Application port addressing scheme 16 bit address	99	SMS	No	Yes	3.15.4
0x06	Service center control parameters	99	SMS	No	Yes	3.15.5
0x07	User-Data-Header source indicator	99	SMS	Yes	Yes	3.15.6
0x08	Concatenated short message 16 bit reference number	99	SMS	No	Yes	3.15.2
0x09	Wireless control message protocol	99	SMS	No	Yes	3.15.8
0x0A	Text formatting	99	EMS 1	Yes	Yes	4.3.1 and 4.4.18
0x0B	Predefined sound	99	EMS 1	Yes	Yes	4.3.3.1
0x0C	User defined sound[1]	99	EMS 1	Yes	Yes	4.3.3.2
0x0D	Predefined animation	99	EMS 1	Yes	Yes	4.3.4.1
0x0E	Large animation	99	EMS 1	Yes	Yes	4.3.4.2
0x0F	Small animation	99	EMS 1	Yes	Yes	4.3.4.2
0x10	Large picture	99	EMS 1	Yes	Yes	4.3.2.1
0x11	Small picture	99	EMS 1	Yes	Yes	4.3.2.2
0x12	Variable-size picture	99	EMS 1	Yes	Yes	4.3.2.3
0x13	User prompt indicator	4	EMS 1	Yes	Yes	4.3.5
0x14	Extended object	5	EMS 2	Yes	Yes	4.4.1
0x15	Reused extended object	5	EMS 2	Yes	Yes	4.4.2
0x16	Compression control	5	EMS 2	Yes	Yes	4.4.3
0x17	Object distribution indicator	5	EMS 1	Yes	Yes	4.3.6
0x18	Configurable-size WVG object	5	EMS 2	Yes	Yes	4.4.17.2
0x19	Character size WVG object	5	EMS 2	Yes	Yes	4.4.17.1
0x1A	Extended object data request command	5	EMS 2	No	No	4.4.20
0x1B...0x1F	Reserved for future EMS features	N/A	N/A	N/A	N/A	–
0x20	RFC 822 Email header	99	SMS	No	Yes	3.20.2
0x21	Hyperlink format element	5	SMS	Yes	Yes	4.4.19
0x22	Alternate reply address	5	SMS	No	Yes	3.15.9
0x23	Enhanced voice mail notification	6	SMS	No	Yes	3.15.10
0x24...0x6F	Reserved for future use	99	N/A	N/A	N/A	–
0x70...0x7F	(U)SIM toolkit security header	99	SMS	No	No	3.15.7
0x80...0x9F	SME to SME specific use	N/A	N/A	N/A	N/A	–
0xA0...0xBF	Reserved for future use	N/A	N/A	N/A	N/A	–
0xC0...0xDF	SMSC specific use	N/A	N/A	N/A	N/A	–
0xE0...0xFF	Reserved for future use	N/A	N/A	N/A	N/A	–

from Rel.: this table field indicates the release of the technical specification in which the information element was introduced.

Type: this table field indicates whether the information element is related to SMS control (SMS), basic EMS (EMS 1), or extended EMS (EMS 2).

[1] iMelody format with a maximum size of 128 octets.

concatenation instructions, the payload consists of the concatenation segment index, message reference number, and number of segments in the concatenated message.

Note that one single information element needs to fit into one message segment. An information element cannot be segmented and spread over several message segments.

If the receiving SME does not recognize a particular IE identifier, then the SME interprets the IEDL and skips the entire IE and continues the processing of the immediately following IE. For instance, this can occur if an SME was designed before the inclusion of an IE in the standard. Regarding this method of handling IEs, it has to be noted that older SMEs may completely ignore certain IEs. This can become the source of awkward message presentation or application misbehavior.

SMS control information elements are presented in this chapter. EMS information elements are presented in Chapter 4.

An IE is segment repeatable if several IEs with the same identifier can appear more than once in a message segment. Additionally, an IE is message repeatable if several IEs with the same identifier can appear more than once in a message.

In the situation where several IEs with identical identifiers are present in a message segment and are not segment repeatable, then the last occurrence of the IE is used. If two IEs contain mutually exclusive instructions (e.g. an 8-bit port address and a 16-bit port address), then the last occurring IE is used.

The values listed in Table 3.21 can be assigned to the 1-octet IE identifier parameter. For each information element, a table summarizes the structure of the information element, as described in Figure 3.33.

3.15.2 Concatenation of Message Segments

The concatenation information element indicates that a message segment represents one segment from a concatenated message. Parameters conveyed in this information element are the concatenated message reference number (not to be confused with the value assigned to the `TP-Message-Reference` parameter[7]), the number of segments in the concatenated message and the sequence number of the segment in the concatenated message. Two distinct information elements have been defined. The two concatenation information elements differ in the number of bits that are used for representing the message reference number.

- Information element identifier with IE identifier 0x00 allows the reference number to be represented with 8 bits (range 0...255, decimal values).
- Information element identifier with IE identifier 0x08 allows the reference number to be represented with 16 bits (range 0...65,535, decimal values).

The information element, for an 8-bit-reference-number concatenation, is structured as shown in Table 3.22.

[7]All message segments composing a concatenated message have distinct values assigned to the `TP-Message-Reference` parameter. However, they have the same value assigned to the concatenated message reference number in the concatenation information element.

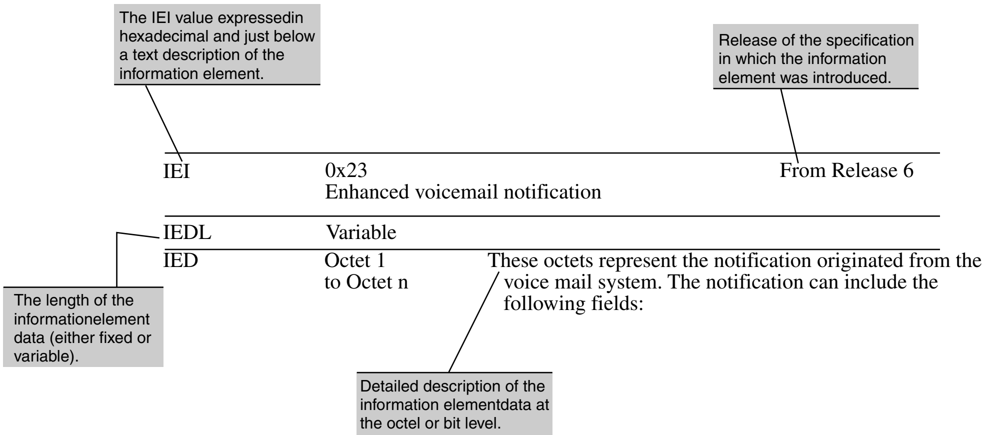


Figure 3.33 How to read the IE description table

Note that if the concatenation sequence number of a segment is 0 or is greater than the number of segments in the concatenated message, then the concatenation information element is usually ignored by the recipient SME. In this case, message segments forming the concatenated message are presented independently as one-segment messages.

Figure 3.34 shows the coding of two message segments composing a concatenated message. In this example, the first message segment has a concatenation reference number of 0x14 and a sequence number equal to 1. The second message segment has the same concatenation reference number and a sequence number equal to 2. Before reconstruction, the receiving SME checks that the two segments have the same originator address.

Similarly, the information element for a 16-bit reference number concatenation is structured as shown in Table 3.23.

Table 3.22 IE/concatenation with 8-bit reference number

IEI	0x00 Concatenation with 8-bit reference number		From Release 99
IEDL	0x03 (3 octets)		
IED	Octet 1	Concatenated message reference number. This is a modulo 256 number which remains the same for all segments composing a concatenated message.	
	Octet 2	Number of segments in the concatenated message. This number represents the number of segments composing the concatenated message.	
	Octet 3	Sequence number of this segment in the concatenated message. The first segment of a concatenated message has a sequence number of 1. Value 0 is reserved.	

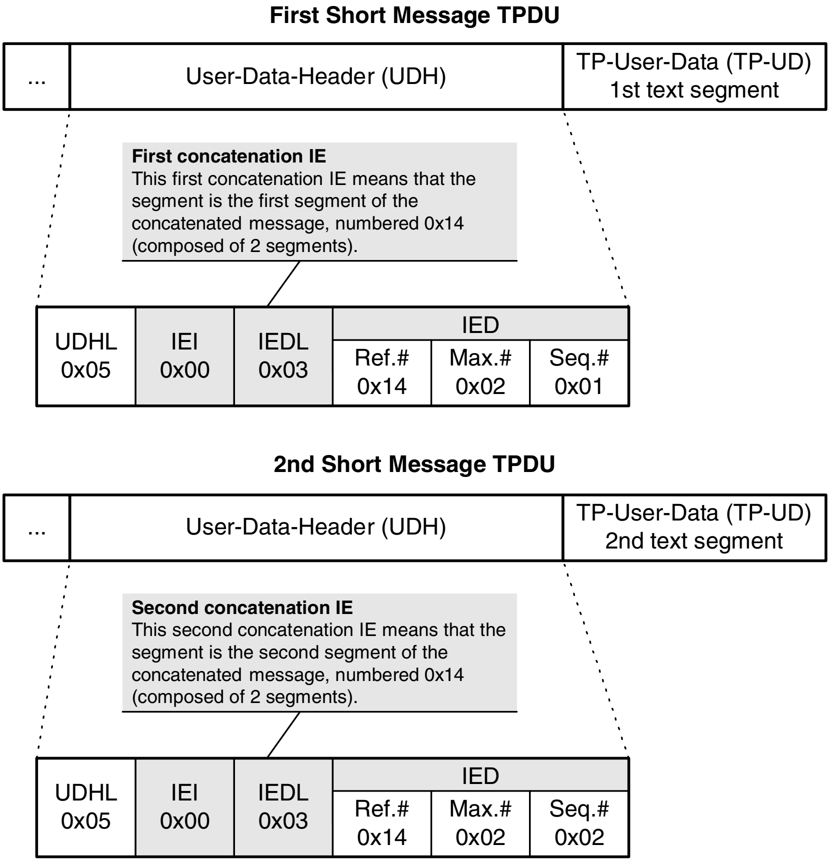


Figure 3.34 IE concatenation/example

Table 3.23 IE/concatenation with 16-bit reference number

IEI	0x08 Concatenation with 16-bit reference number	From Release 99
IEDL	0x04 (4 octets)	
IED	Octets 1 and 2	Concatenated short message reference number. This is a modulo 65,536 number which remains the same for all segments composing a single concatenated message.
	Octet 3	Maximum number of short messages in the concatenated message. This number represents the number of segments composing the concatenated message.
	Octet 4	Sequence number of this segment in the concatenated message. The first segment of a concatenated message has a sequence number of 1. Value 0 is reserved.

Box 3.1 Concatenated message identification
The triplet composed of the concatenation reference number, the message originator address, and the number of message segments represents the key for uniquely identifying a concatenated message. The 16-bit reference number is preferred since it reduces the probability of encountering the conflicting situation where two different segments in the receiving SME have the same message key and the same segment sequence number. In this conflicting situation, the SME faces difficulties in reconstructing the two distinct messages. However, the usage of the 16-bit reference number means that one additional octet is used per segment of a concatenated message. It is not advisable to mix concatenation elements with 8-bit and 16-bit reference numbers in the same concatenated message.

3.15.3 Special SMS Message Indication

This information element is used for updating waiting message indicators in the receiving handset. These indicators inform the subscriber whether or not messages are awaiting retrieval from a server (voice mail, fax, Email, video message, etc.). Depending on the handset capabilities, indicators may additionally inform the subscriber on the number of messages that are waiting. The information element has the structure shown in Table 3.24.

If several indicators are to be updated, then several information elements of this type can be inserted in one short message or in several message segments composing a concatenated message.

In addition to the information element method, three other methods exist for updating message waiting indicators.

- `TP-Data-Coding-Scheme` method: a message waiting indicator can also be updated with the delivery of a message for which a special value has been assigned to the `TP-Data-Coding-Scheme` parameter. With this method, a message waiting indicator can be activated or deactivated on the receiving handset. Unlike the information element method, the number of waiting messages cannot be specified. If the value assigned to the `TP-Data-Coding-Scheme` indicates that the message is to be stored (see also coding groups in Section 3.7.6), then the message may contain some additional text (GSM 7-bit default alphabet or UCS2). Special values to be assigned to the `TP-Data-Coding-Scheme` parameter are structured as shown in Figure 3.35.
- `TP-Originator-Address` method: another method is commonly supported by SMEs. This method consists of assigning a special value to the `TP-Originator-Address` parameter of the message to be delivered. Note that this method has been defined by a group of operators (Orange and T-Mobile (UK), formerly One2one) and has not yet been published by any recognized standardization development organization. This method is defined in the Common PCN Handset Specification (CPHS). 3GPP members can download this specification from the following location:

  `http://www.3gpp.org/ftp/tsg_t/WG3_USIM/TSGT3_00_old_meetings/TSGT3_15/docs/T3-000450.zip`

Table 3.24 IE/special SMS message indication

IEI	0x01 Special SMS message indication	From Release 99 Updated in Release 6
IEDL	0x02 (2 octets)	
IED	Octet 1	**Message indication type and storage** *Message storage:* Bit 7 of octet 1 indicates whether the message shall be stored or not in the MS. Bit 7 is set to 1 if the short message is to be stored and set to 0 otherwise. *Basic indication type:* Bits 0 and 1 of octet 1 indicate which indicator is to be updated. These bits can take the following values: Bit 1 0 — Basic indicator 00 — Voice Message Waiting 01 — Fax Message Waiting 10 — Email Message Waiting 11 — Extended Message Type Waiting *Extended indication type*[1] Bits 6...2 of octet 1 indicate the extended message indication type (11 assigned to Bits 0 and 1). These bits can take the following values: Bit 6. . .2 — Extended indicator 000 00 — No extended message indication 000 01 — Video message waiting other reserved
	Octet 2	**Number of waiting messages** This number represents the number of waiting messages. It can range from 0 to 255. The value 255 for this octet means that 255 or more messages are waiting.

[1] Note that the extended indication type was introduced in Release 6 of [3GPP-23.040].

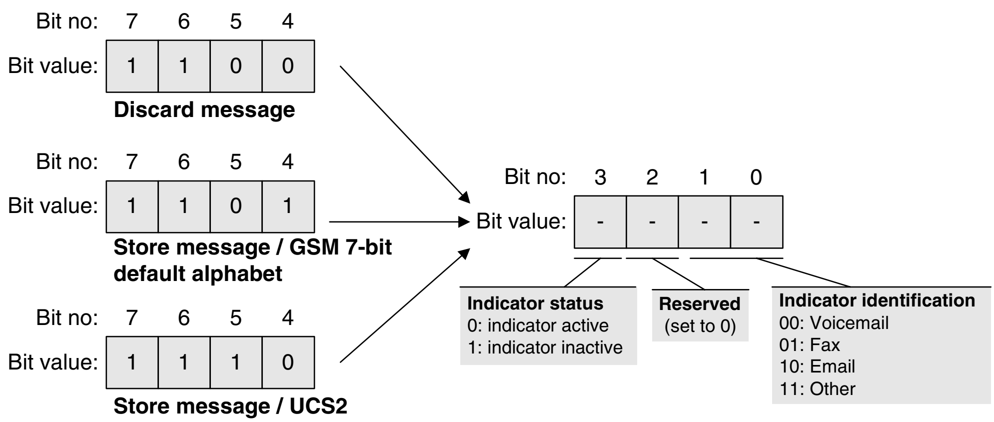


Figure 3.35 Message waiting indicator/`TP-Data-Coding-Scheme` method

- Return call message: a fourth method, to indicate to the subscriber that one or more messages are waiting, consists of sending a return call message (see `TP-Protocol-Identifier` description in Section 3.7.7). In this situation, the message contains a text description indicating that one or more messages are waiting retrieval. Note that with this method, message waiting indicators are usually not updated automatically.

3.15.4 Application Port Addressing

Application port addressing is a feature allowing the routing of a received message to the port of an identified application running in the mobile station. Applications that benefit from such a feature are personal information managers, programs handling over-the-air provisioning of configuration parameters, etc. Application port addressing can be realized using two distinct information elements. The first information element is used for 8-bit address ports whereas the second information element is used for 16-bit address ports. If a concatenated message is to be routed to an application, then the associated application port addressing IE is incorporated in all message segments composing the concatenated message.

For applications with 8-bit address ports, the information element shown in Table 3.25 is used. In the 8-bit address range, the values listed in Table 3.26 are used.

For applications with 16-bit address ports, the information element shown in Table 3.27 is used. In the 16-bit address range, the values listed in Table 3.28 are used.

Table 3.25 IE/application port addressing scheme, 8-bit address

IEI	0x04 Application port addressing scheme, 8 bit address		From Release 99
IEDL	0x02 (2 octets)		
IED	Octet 1	**Destination port** This octet indicates the 8-bit address of the receiving port.	
	Octet 2	**Originator port** This octet indicates the 8-bit address of the sending port.	

Table 3.26 Port number range/8-bit addresses

Port number range		Description
From	to	
0x00 0 (decimal)	0xEF 239 (decimal)	Reserved.
0xF0 240 (decimal)	0xFF 255 (decimal)	Available for allocation by applications.

Table 3.27 IE/application port addressing scheme, 16-bit address

IEI	0x05 Application port addressing scheme, 16-bit address	From Release 99
IEDL	0x04 (4 octets)	
IED	Octets 1 and 2	**Destination port** These octets indicate the 16-bit address of the receiving port.
	Octets 3 and 4	**Originator port** These octels indicate the 16-bit address of the sending port.

Table 3.28 Port number range/ 16-bit address

Port number range		Description
From	to	
0x0000 0 (decimal)	0x3E7F 15,999 (decimal)	See default address ports as defined by IANA[1].
0x3E80 16,000 (decimal)	0x4267 16,999 (decimal)	Available for allocation by applications.
0x4268 17,000 (decimal)	0xFFFF 65,535 (decimal)	Reserved.

[1]IANA port numbers are available at `http://www.iana.org/assignments/port-numbers`

3.15.5 Service Center Control Parameters

This information element is used to convey control instructions in a flexible way. These instructions may be interpreted by the SMSC or the recipient SME. In particular, as part of a message, this information element identifies the set of events for which a status report should be generated by the SMSC. The possible events which can be identified for the generation of a status report are the following:

- A transaction has been completed
- A permanent error when the SMSC is not making anymore transfer attempts
- A temporary error when the SMSC is not making anymore transfer attempts
- A temporary error when the SMSC is still trying to transfer the message.

For the successful generation of the status report by the SMSC or by the receiving entity, the originator SME must also set the `TP-Status-Report-Request` parameter in the submitted message segment. The information element has the structure shown in Table 3.29.

If the original user data header is to be included in the status report (see bit 7 description), then the status report normal UDH is differentiated from the original UDH with the UDH source indicator information element as described in the following section.

Table 3.29 Service center control parameters

IEI	0x06 Service center control parameters	From Release 99
IEDL	0x01 (1 octet)	
IED	Octet 1	**Selective Status Report (SR)** The four least significant bits (bits 3...0) indicate whether or not a specific status report is to be generated as shown below: **Bit 0**/SR for transaction completed. 0: no status report to be generated. 1: status report to be generated. **Bit 1**/SR for permanent error when SMSC is not making any more transfer attempts. 0: no status report to be generated. 1: status report to be generated. **Bit 2**/SR for temporary error when SMSC is not making any more transfer attempts. 0: no status report to be generated. 1: status report to be generated. **Bit 3**/SR for temporary error when SMSC is still trying to transfer the short message. 0: no status report to be generated. 1: status report to be generated. **Bits 4 and 5** are reserved for future use. **Bit 7**/Original UDH This bit indicates whether or not the original user data header is to be included in the status report. The bit is set to 0 if the original UDH is to be included, it is set to 1 otherwise.

If a concatenated message includes service center control parameters, then the service center control parameter information element must be included in each message segment composing the concatenated message.

3.15.6 User-Data-Header Source Indicator

The User-Data-Header (UDH) source indicator information element is used to combine several User-Data-Headers provided by sources such as the originator SME, the recipient SME or the SMSC in a single message segment. A message segment can contain multi-source User-Data-Header information in status reports or in messages used for updating message waiting indicators. The information element has the structure shown in Table 3.30.

Figure 3.36 shows how the UDH Source Indicator can be used for separating various sequences of information elements in the User-Data-Header of a status report.

Table 3.30 IE/user data header source indicator

IEI	0x07 User-Data-Header source indicator	From Release 99
IEDL	0x01 (1 octet)	
IED	Octet 1	**Source indicator** Value 0x01 indicates that the following part has been generated by the original sender (valid in case of a status report). Value 0x02 indicates that the following part has been generated by the original receiver (valid in case of a status report). Value 0x03 indicates that the following part has been generated by the SMSC (valid in any message or status report). Value 0x00 and values ranging from 0x04 to 0xFF are reserved values.

3.15.7 (U)SIM Toolkit Security Header

This group of 16 information elements is used for indicating the presence of a (U)SIM toolkit security header in the `TP-User-Data` after the User-Data-Header. Such an information element is only inserted in the first message segment of a concatenated message. One characteristic of these information elements is that they do not have any payload (value 0 assigned to IEDL and there is no IED). These information elements are structured as shown in Table 3.31.

3.15.8 Wireless Control Message Protocol

This information element is used for exchanging states of applications or protocols between SMEs. This method avoids the unnecessary retransmission of packets in the WAP environment when an application is not available for receiving data. The information element is structured as shown in Table 3.32.

3.15.9 Alternate Reply Address

In the context of SMS, a message reply is typically sent to the address of the message originator. Alternatively, during message submission, the message originator has the capability to indicate that message replies should be sent to an alternate reply address. This is performed by inserting, in the submitted message, a dedicated information element containing the alternate reply address. This dedicated information element is structured as shown in Table 3.33.

Note that for external SMEs, an alternative method is sometimes supported by SMSCs. This method is called the `TP-Originator-Address` substitution method. It consists of

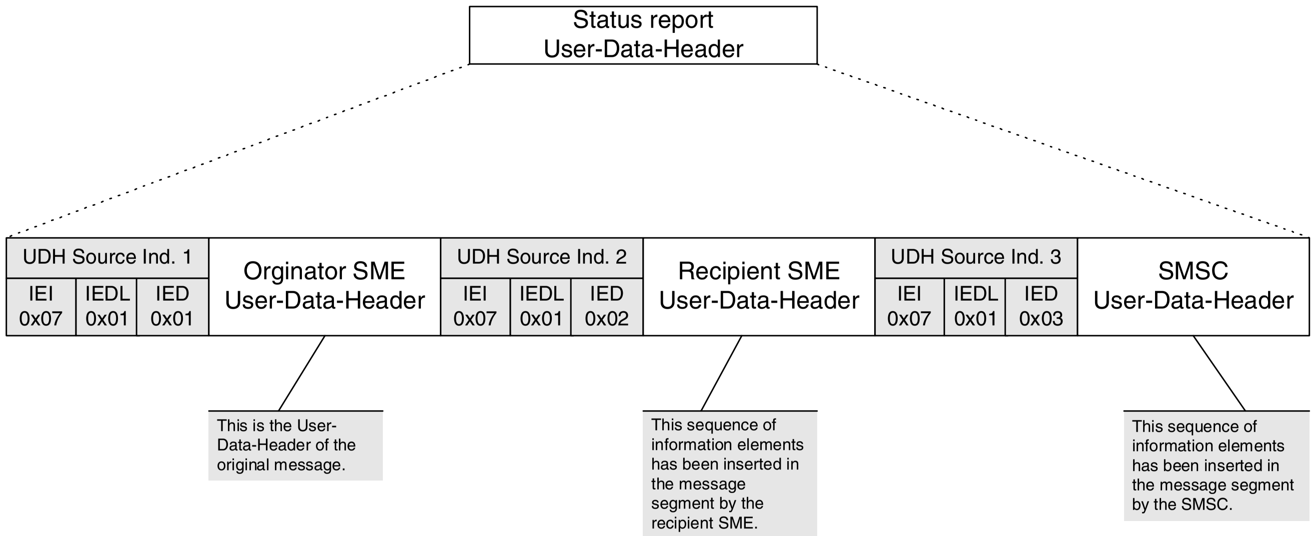


Figure 3.36 UDH source indicator/example

Table 3.31 IE/(U)SIM toolkit security header

IEI	0x70 to 0x7F (U)SIM Toolkit security header	From Release 99
IEDL	0x00 (0 octet - no IED)	

Table 3.32 IE/wireless control message protocol

IEI	0x09 Wireless control message protocol	From Release 99
IEDL	Variable	
IED	Octets 1...n	These octets contain wireless control message protocol information

Table 3.33 IE/alternate reply address

IEI	0x22 Alternate reply address	From Release 5
IEDL	Variable	
IED	Octets 1...n	These octets represent the alternate reply address. This address is formatted as defined in Section 3.9.7.

replacing the value, assigned by the external SME to the `TP-Originator-Address`, by an alternate reply address specified by the external SME during message submission.

3.15.10 Enhanced Voice Mail Notification

A Release 6 extension of the SMS standard allows a voice mail system to deliver mail box related information to mobile users. For this purpose, the information element presented in Table 3.34 has been defined. This information can convey one of the two following commands:

- Notification from the voice mail system that new voice messages have been deposited in the voice mail box and indication of new voice mail box status.
- Notification from the voice mail system that selected voice messages have been deleted from the voice mail box and indicaton of new voice mail box status.

The structure of the IE data when conveying a notification providing information about messages present in the voice mail box is shown in Figure 3.37.

The structure of the IE data when conveying a notification providing information about messages that have been deleted from the voice mail box is shown in Figure 3.38.

Table 3.34 IE/enhanced voice mail notification

IEI	0x23 Enhanced voice mail notification	From Release 6
IEDL	Variable	
IED	Octets 1...n	These octets represent the notification originated from the voice mail system. The notification can include the following fields: • notification type (command) • indication whether the message shall be stored or discarded after processing • whether the voice mail box is full or almost full • address used by the user to access the mail box. For the notification indicating messages available in the voice mail box, the following fields are also included: • number of unread voice messages • one or multiple voice mail notifications (message identification, number of retention days, message priority, address of calling party). For the notification identifying deleted messages, the following fields are also included: • number of deleted voice messages • information about each deleted message (message identification).

3.16 Network Functions for Message Delivery

It may happen that the SMSC is unable to deliver a message to a recipient SME due to some temporary or permanent error conditions. This can occur, for instance, if the mobile device has been powered off, if the device has no more storage capacity to handle the message, or if the recipient subscriber is unknown. For failures due to permanent error conditions, the SMSC does not attempt to deliver the message anymore. For failures due to temporary error conditions, the SMSC may store the message temporarily in a queue and may attempt to deliver the message again. Table 3.35 shows the list of temporary and permanent errors that can be returned to an SMSC for a failed message delivery.

To allow the re-transmission of messages by the SMSC, the network which is serving the recipient SME can maintain a Message Waiting Indication (MWI). This indication enables the serving network to know when the recipient SME has retrieved the capabilities for handling messages. In such a situation, the serving network alerts SMSCs for which a previous message delivery to the recipient SME has failed due to some temporary error conditions. Upon alerting, SMSCs can appropriately re-attempt the transmission of queued messages. The MWI is composed of the following set of parameters, maintained by various network elements (HLR, VLR, and GGSN):

- Address of the recipient SME (MSISDN-Alert) and associated addresses of SMSCs that unsuccessfully attempted to deliver messages to the recipient SME. This information is maintained by the HLR.

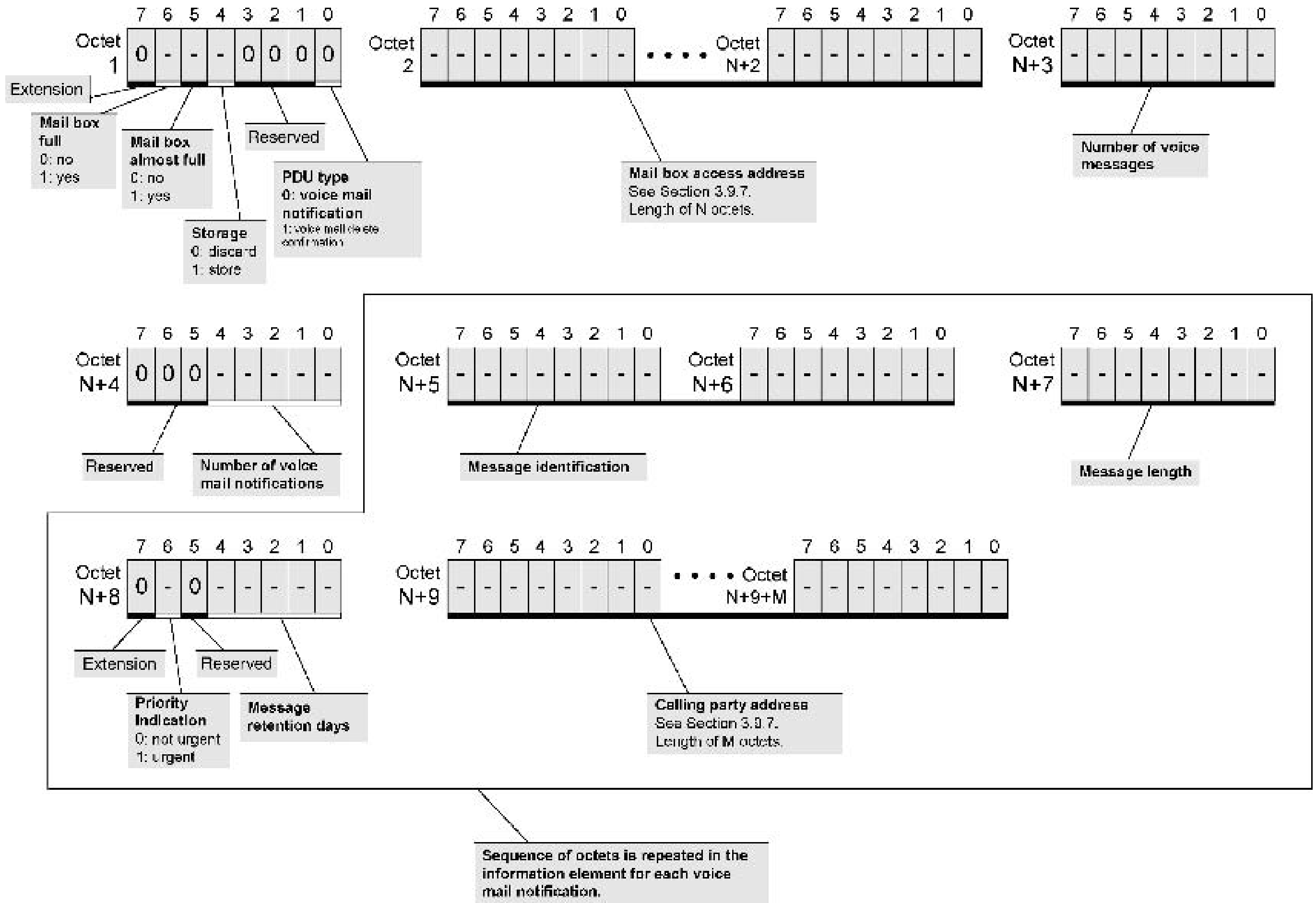


Figure 3.37 IE/enhanced voice mail notification/message information

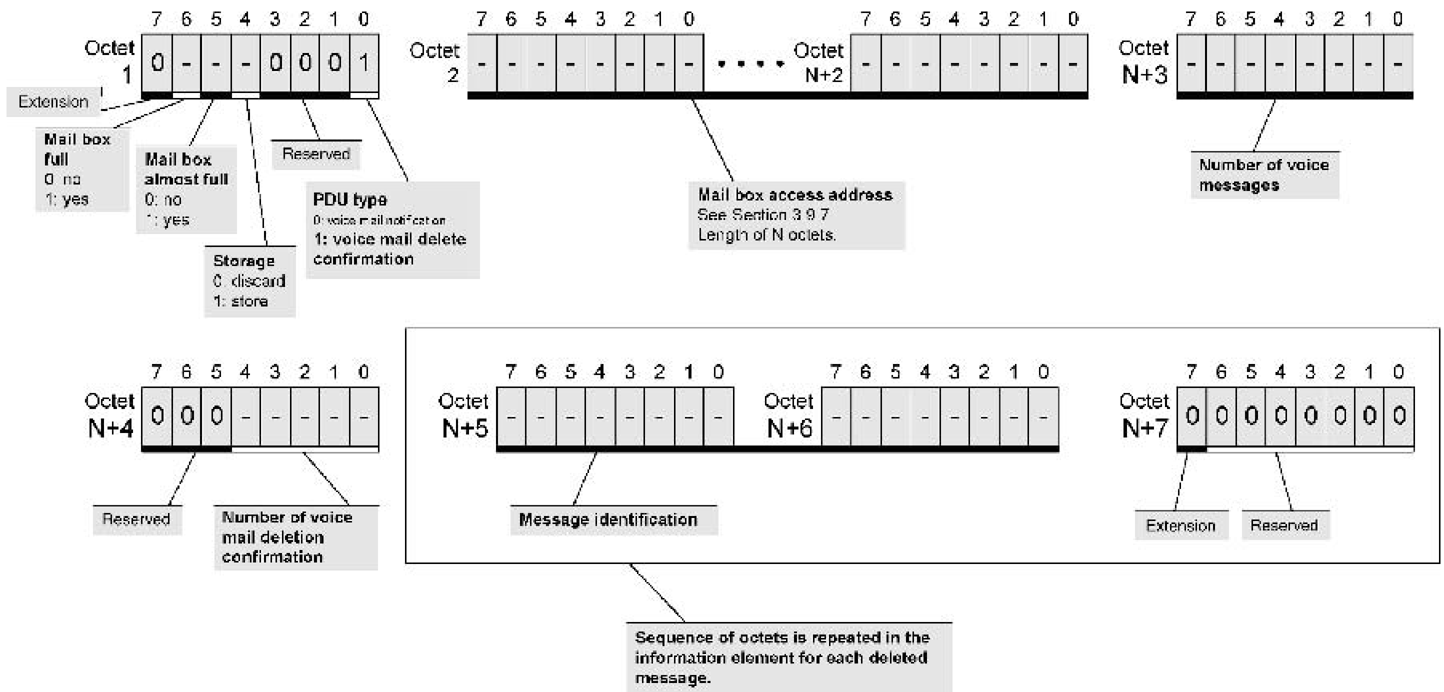


Figure 3.38 IE/enhanced voice mail notification/message deletion

Table 3.35 SMS failure causes

Failure cause	Status
Unknown subscribed	Permanent
Teleservice not provisioned	Permanent
Call barred	Temporary
Facility not supported	Temporary
Absent subscriber	Temporary
MS busy for MT SMS	Temporary
SMS lower layers capabilities not provisioned	Temporary
Error in MS	Temporary
Illegal subscriber	Permanent
Illegal equipment	Permanent
System failure	Temporary
Memory capacity exceeded	Temporary

- The Mobile-station-memory-Capacity-Exceeded-Flag (MCEF) Boolean flag indicates whether or not a message delivery has failed because the storage capacity of the recipient SME has exceeded. This flag is maintained by the HLR.
- In the context of GPRS, the Mobile-station-Not-Reachable-for-GPRS (MNRG) Boolean flag indicates whether or not a message delivery has failed because the recipient SME is absent from the network. This flag is maintained by the HLR or by the SGSN.
- The Mobile-station-Not-Reachable-Flag (MNRF) Boolean flag indicates whether or not a message delivery has failed because the recipient SME is absent from the network. This flag is maintained by the HLR or the VLR.
- The Mobile-station-Not-Reachable-Reason (MNRR) indicates the reason why the recipient SME is absent from the network. This indicator, maintained by the HLR, is either cleared or contains one of the following values: no paging response via the MSC, no paging response via the SGSN, IMSI detached or GPRS detached.

The MWI is updated on signaling events such as network attach, location update, call set-up, etc. After update, if the HLR detects that the associated recipient SME has retrieved the ability to handle messages, then it sends an "Alert-SC" message to all SMSCs that unsuccessfully attempted to deliver messages to the recipient SME. Upon alerting, SMSCs can retry to deliver the message(s) for this particular recipient SME, without delay.

In the situation where the recipient SME rejected a message delivery because of a storage capacity failure, the SME can alert the HLR when it has retrieved the ability to store messages. This is performed by sending a specific message (`RP-SM-MEMORY-AVAILABLE`), at the relay layer, to the HLR. Upon receipt of such a message, the HLR alerts the SMSCs which unsuccessfully attempted to deliver one or more messages to the associated SME.

In addition to the possibility of an alert from the serving network that the recipient SME is available for handling message deliveries, an SMSC can also independently re-attempt the message delivery after an appropriate period of time. A well designed SMSC usually supports separate configurable retry algorithms for each delivery failure cause. With such SMSCs, the operator can configure the delay between two successive delivery attempts and the number of attempts performed by the SMSC. Typically, the delay between successive delivery attempts depends on the cause of the delivery failure and will increase if the causes for message delivery failures remain the same. Overall, the operator configures the SMSC retry procedure in order to achieve the best balance between subscriber service quality and network loading.

3.17 SMSC Access Protocols

SMSC access protocols enable interactions between two SMSCs or interactions between an external SME and an SMSC. Although 3GPP, in technical report [3GPP-23.039], recognizes five main SMSC access protocols, only the four following protocols are widely used: SMPP (by Logica, now LogicaCMG), CIMD (by Nokia), UCP/EMI (by CMG, now LogicaCMG), and the Open Interface Specification (OIS) (by SEMA).

These protocols are proprietary and are usually binary protocols operating over TCP/IP or X.25. SMS Forum, a non-profit organization established by companies willing to promote SMS in the wireless industry, adopted SMPP as its recommended binary access protocol for service centers. In addition, the SMS Forum has published specifications for enabling XML-based interactions between an application and an SMS center (directly or via a messaging gateway):

- *Short Message Application Part* (SMAP) is an XML format for defining requests and responses enabling interactions between the application and the SMS center.
- *Mobile Message Access Protocol* (MMAP) is a SOAP-based protocol for transporting these requests and responses.

This chapter outlines two binary protocols: SMPP from the SMS Forum and the SMSC open interface specification from SEMA Group. Descriptions of SMAP and MMAP are also provided.

3.17.1 SMPP from SMS Forum

The Short Message Peer to Peer (SMPP) was originally developed by Logica (now LogicaCMG) and the protocol has now been adopted by the SMS Forum as an open binary access protocol enabling interactions between external SMEs and service centers developed by different manufacturers.

In order to interact with an SMSC via the SMPP protocol, an external SME first establishes a session. The transport of operation requests over this session is usually performed over TCP/IP or X.25 connections. For TCP/IP, application port 2775 is used for SMPP (this may, however, vary in implementations from different manufacturers). Operations over SMPP sessions can be categorized into the following four groups:

- *Session management*: these operations enable the establishment of SMPP sessions between an external SME and the SMSC. In this category, operations also provide a means of handling unexpected errors.
- *Message submission*: these operations allow external SMEs to submit messages to the SMSC.
- *Message delivery*: these operations enable the SMSC to deliver messages to external SMEs.

- *Ancillary operations*: these operations provide a set of features such as cancellation, query, or replacement of messages.

SMPP is an asynchronous protocol. This means that the external SME may send several instructions to the SMSC without waiting for the result of previously submitted instructions. SMPP itself does not define any encryption mechanism for the exchange of messages between the external SME and the SMSC. However, two methods are recommended for securing interactions. The first method consists of using the well known Secure Socket Layer (SSL) and its standardized form, Transport Layer Security (TLS). The second method consists of allowing an SMPP session to operate over a secure tunnel. In this configuration, the connection is not made directly between the external SME and the SMSC but is relayed between two secure tunnel servers.

3.17.2 SMS Open Interface Specification from Sema Group

Sema Group Telecoms (now part of the Atos Origin group) provides the SMSC open interface specification along with its SMSCs. An external SME interacts with the SMSC with a protocol designed to operate over a variety of interfaces such as X.25, DECnet, and SS7. This SMSC is usually accessed via the general X.25 gateway (either by using a radio Packet Assembler Disassembler or a dedicated link to the SMSC). The SME connected to the SMSC invokes an operation by sending a request to the SMSC. When the SMSC has completed the request, it sends the operation result back to the SME. Alternatively, the SMSC may invoke an operation by sending a request to the SME. When the SME has completed the request, it sends the operation result back to the SMSC. Possible operations are shown in Figure 3.39.

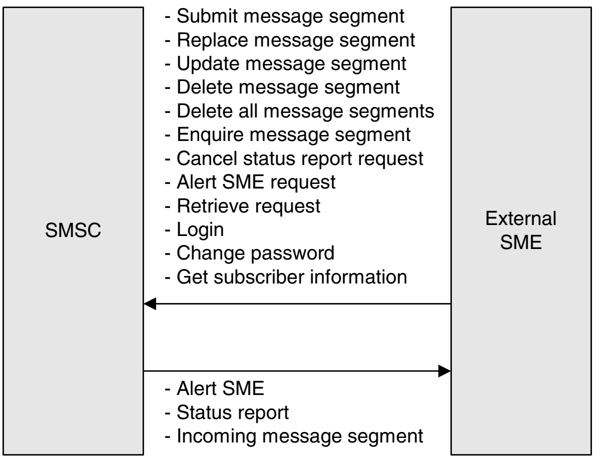


Figure 3.39 SEMA Group service center configuration

An external SME can request the following operations from the service center:

- *Submit message segment*: this operation is for sending a message segment to an SME.
- *Delete message segment*: this command is used for deleting a previously submitted message segment.
- *Replace message segment*: this operation allows the replacement of a previously submitted message segment (which has not been delivered yet) by a new message segment.
- *Delete all message segments*: delete all previously submitted message segments which have not been delivered yet.
- *Enquire message segment*: request the status of a previously submitted message segment.
- *Cancel status report request*: this operation is for canceling a request for the generation of a status report related to the previous submission of a message segment.
- *Alert SME request*: this command allows alerts when a specified SME has registered.
- *Login*: this operation allows users to access to the SMSC remotely.
- *Change password*: this operation allows users to change their password.
- *Get subscriber information*: this operation allows an SME to query the network HLR to determine if a network node is currently serving a mobile station.

The SMSC can request the following operations from the external SME:

- *Alert SME*: this operation is used by the SMSC to indicate to the SME that a mobile station has registered in the network.
- *Status report*: this operation is used by the SMSC to provide a status report to the external SME. The status report indicates the status of the corresponding message segment delivery.
- *Incoming message segment*: this operation is used by the SMSC to provide an incoming message segment addressed to the SME.

3.17.3 MMAP and SMAP

As previously indicated, access protocols initially developed to allow interactions between SMSCs and external SMEs are binary protocols. In addition, SMS Forum[8] has published the specifications for:

- *Short Message Application Part* (SMAP), an XML format for defining requests and responses enabling communications between the application and the SMS center.
- *Mobile Message Access Protocol* (MMAP), a SOAP-based protocol for transporting these requests and responses.

An external SME may communicate directly with an SMSC over MMAP (only if the SMSC has native support for MMAP). Alternatively, an SMS gateway can fit between the external SME and the SMSC. This latter solution allows an easier evolution path from previous proprietary configurations. Such a configuration is shown in Figure 3.40.

The design of SMAP (version 1.0) makes it functionally equivalent to SMPP (version 3.4). For this purpose, SMAP operations are categorized into the four SMPP groups of operations. SMAP is an application protocol independent from underlying transport protocols. However, SMS Forum recommends the use of the SOAP-based protocol MMAP.

[8] http://www.smsforum.net/

Figure 3.40 SMAP/MAP configuration with SMS gateway

Figure 3.41 shows an example of a SMAP operation request formatted in XML. This operation corresponds to a message submission request from an external SME.

The following four operational modes are available with SMAP:

- *Immediate mode*: with this mode, the external SME does not maintain a session with the gateway. Each operation contains the application context. This mode is used for message submissions only.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope
   xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope
         http://www.w3.org/2003/05/soap-envelope">
  <SOAP:Header>

    <MMAP:MMAPHeader SOAP:mustUnderstand="1"
      xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
      xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
            http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">

      <MMAP:ApplicationContext bodyType="Request"
                sourceOperationReference="1138"/>

    </MMAP:MMAPHeader>

  </SOAP:Header>
  <SOAP:Body>

    <SMAP:SubmitRequest
         xmlns:SMAP="http://www.smsforum.net/schemas/smap/v1.0"
         xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.0
              http://www.smsforum.net/schemas/smap/v1.0/smap.xsd">
      <SMAP:ShortMessage>
        <SMAP:Header>
          <SMAP:Destination>
            <SMAP:Number>6666854</SMAP:Number>
          </SMAP:Destination>
        </SMAP:Header>
        <SMAP:Body>
          <SMAP:Text>Hello world!</SMAP:Text>
        </SMAP:Body>
      </SMAP:ShortMessage>
    </SMAP:SubmitRequest>

  </SOAP:Body>
</SOAP:Envelope>
```

SMAP submission request

Figure 3.41 SMAP protocol data unit conveyed over MMAP

- *Client session mode*: with this mode, the external SME first establishes a session with the gateway prior to requesting operations to be processed by the SMSC. The gateway may also establish such a session for message delivery to an external SME.
- *Peer-to-peer session mode*: this mode allows a bi-directional session to be established between the external SME and the SMSC. Message submissions and deliveries can be performed over a single bi-directional session.
- *Batch mode*: with this mode, the gateway receives a set of SMAP operations to be processed from the external SME. The gateway processes each operation in turn and builds a set of results. The set of results is also provided in a batch to the external SME. The batch mode is usually used when an interactive session is not required or would be unsuitable due to timeout issues.

3.18 SIM Application Toolkit

The SIM Application Toolkit (SAT), defined in [3GPP-31.111], defines mechanisms for allowing SIM-hosted applications to interact with the mobile equipment. This includes the following mechanisms:

- *Profile download*: this mechanism allows the mobile equipment to inform the SIM about its capabilities.
- *Proactive SIM*: a proactive SIM can issue commands to the mobile equipment. Section 3.18.1 provides a description of features available to proactive SIMs.
- *Data download to SIM*: it was shown in this chapter that a particular `TP-Protocol-Identifier` value could be used to download data to the SIM. This mechanism is further detailed in Section 3.18.2.
- *Menu selection*: this mechanism allows the (U)SIM to define menu items and to be notified by the mobile equipment when the subscriber has selected one of the menu items.
- *Call control by SIM*: with this mechanism, the SIM establishes a control prior to the establishment of calls by the mobile equipment. This allows the SIM to authorize or reject the call establishment or to modify the parameters of the call to be established.
- *Control of outgoing messages by SIM*: with this mechanism, the SIM performs a control prior to the sending of messages by the mobile equipment. This allows the SIM to authorize or reject the sending of a message.
- *Event download*: this mechanism allows the SIM to provide a set of events to be monitored by the mobile equipment. If an event occurs then the mobile equipment notifies the SIM.
- *Security*: this mechanism ensures data confidentiality, data integrity, and data sender validation.
- *Timer expiration*: the SIM can manage a set of timers running physically in the mobile equipment.
- *Bearer independent protocol*: this mechanism enables the SIM to establish a data connection between the SIM and the mobile equipment and between the mobile equipment and a remote server.

3.18.1 Proactive SIM

Technical specification [3GPP-11.11] defines the protocol of communications between the SIM and the mobile equipment. The protocol is known as the $\mathtt{T} = \mathtt{0}$ protocol (or $\mathtt{T} = \mathtt{1}$ for the USIM). A characteristic of this protocol is that the mobile equipment remains the ‘‘master’’ during the communications and is the element which initiates all commands to the SIM. In this protocol, there is no means for the SIM to issue commands to the mobile equipment. In order to cope with this limitation, the concept of a proactive command was introduced. A SIM making use of proactive commands is known as a proactive SIM. With proactive commands, the SIM is able to issue a command to the mobile equipment by specifying the proactive command in the response to a normal command previously submitted by the mobile equipment to the SIM. Upon receipt of such a response, the mobile equipment executes the specified command and provides the execution results to the SIM as part of a normal command.

3.18.2 SIM Data Download

As shown in Section 3.7.7, two specific values can be assigned to the `TP-Protocol-Identifier` parameter (0x7C and 0x7F) for SIM data download. Upon receipt of a message containing one of these two values, the receiving mobile equipment provides the message to the SIM. The message is provided to the SIM by the use of a SAT command known as an `ENVELOPE` (`SMS-PP DOWNLOAD`) command. The subscriber is not notified of the receipt of the message by the mobile equipment.

3.18.3 SIM Interactions: Example

In order to illustrate the use of SIM proactive commands and SIM data download messages, a simple scenario is described in this section. A SIM application maintains a SIM elementary file in which the subscriber geographical location is regularly updated. For the purpose of collecting statistics on subscriber moves, an application hosted in a remote server (external SME) regularly queries the mobile station for the subscriber location. For this purpose, the external SME submits a ‘‘querying’’ message to the SMSC. The SMSC delivers the message to the recipient mobile station. Upon receipt, the mobile equipment detects that the message is a SIM data download message and therefore provides the message to the SIM as part of an `ENVELOPE SAT` command. The SIM analyzes the message payload and generates an additional message that contains the results of the transaction (subscriber location, e.g., retrieved with a GPS module connected to the mobile equipment). The SIM issues the message to the mobile equipment via the `SEND SHORT MESSAGE` proactive command. Upon receipt of the proactive command, the mobile equipment submits the message to the SMSC which in turn delivers the message to the external SME. Finally, the external SME analyzes the payload of the ‘‘result’’ message and updates a database. The flow of interactions for such a scenario is depicted in Figure 3.42.

3.19 SMS and AT Commands

Technical specification [3GPP-27.005] defines interface protocols for control of SMS functions between the MS and an external Terminal Equipment (TE) via an asynchronous interface. The MS and the TE are connected with a serial cable, an infrared link, or any other similar link as shown in Figure 3.43.

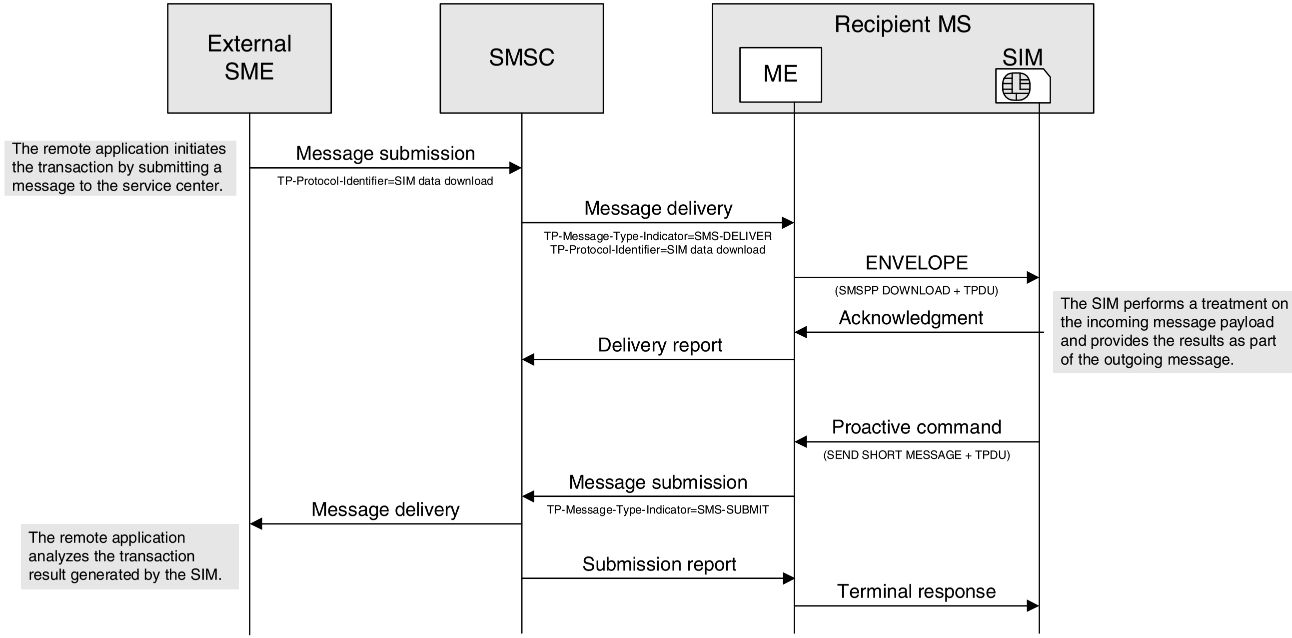


Figure 3.42 Example of interactions between an external SME and a SIM

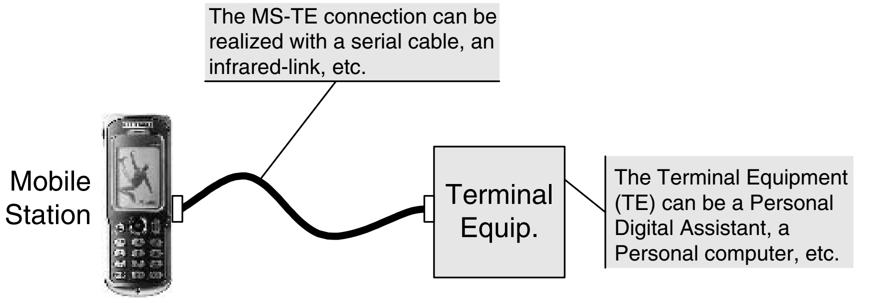


Figure 3.43 MS connection with terminal equipment

Communications between the MS and the TE can be carried out in three different modes:

- *Block mode*: a binary communications protocol including error protection. It is advisable to use this mode if the link between the MS and the TE is not reliable.
- *Text mode*: a character-based protocol suitable for high-level software applications. This protocol is based on AT commands. AT stands for ATtention. This two-character abbreviation is always used to start a command line to be sent from TE to MS in the text mode.
- *Protocol Data Unit (PDU) mode*: a character-based protocol with hexadecimal-encoded binary transfer of commands between the MS and the TE. This mode is suitable for low-level software drivers that do not understand the content of commands.

Regardless of the mode, the TE is always in control of SMS transactions. The TE operates as the "master" and the MS as the "slave". The block mode is a self-contained mode whereas the text and PDU modes are just sets of commands operated from the V.25ter command state or online command state.

3.19.1 AT Commands in Text Mode

This section provides an outline of commands available in the text mode only. In this mode, SMS-related AT commands are categorized in the following four groups:

- General configuration commands allow the terminal equipment to configure the way it wishes to communicate with the mobile station.
- Message configuration commands enable the terminal equipment to consult and update the mobile station SMS settings (service center address, etc).
- Message receiving and reading commands allow the terminal equipment to read messages locally stored in the mobile station and to be notified of incoming messages.
- Message sending and writing commands enable the terminal equipment to create, send, and delete messages in the mobile station.

The list of SMS-related AT commands is given in Table 3.36.

Table 3.36 SMS-related AT commands

	AT command	Description
General configuration commands	+CSMS	Select message service
	+CPMS	Preferred message storage
	+CMGF	Message format
	+CESP	Enter SMS block mode protocol
	+CMS ERROR	Message service failure code
Message configuration commands	+CSCA	Service center address
	+CSMP	Set text mode parameters
	+CSDH	Show text mode parameters
	+CSCB	Select cell broadcast message types
	+CSAS	Save settings
	+CRES	Restore settings
Message receiving and reading commands	+CNMI	New message indications to TE
	+CMGL	List messages
	+CMRG	Read message
	+CNMA	New message acknowledgment
Message sending and writing commands	+CMGS	Send message
	+CMSS	Send message from storage
	+CMGW	Write message to memory
	+CMGD	Delete message
	+CMGC	Send command
	+CMMS	More message to send

3.19.2 AT Command Usage: Example

The example in Figure 3.44 shows how a message can be created in the ME message local store and sent to a recipient.

3.20 SMS and Email Interworking

Interworking between SMS and Email is enabled by allowing the conversion of an SMS message into an Email message, and vice versa. This conversion is performed by the Email

Direction		AT commands	Comments
(TE->MS)	1.	AT+CMGW="+33612121212"	1. Write message (recipient address is +33612121212).
(TE->MS)	2.	> This is a one-line message.^Z	2. Enter message text (end is control-Z).
(MS->TE)	3.	+CMG: 7	3. Message has been stored with index 7.
(MS->TE)	4.	OK	4. Message writing is successful.
(TE->MS)	5.	AT+CMSS=7	5. Send message previously stored.
(MS->TE)	6.	+CMSS: 12	6. Message sent with reference value 12.
(MS->TE)	7.	OK	7. Message sending is successful.
(TE->MS)	8.	AT+CMGD=7	8. Request for message deletion.
(MS->TE)	9.	OK	9. Message has been deleted.

Figure 3.44 AT command usage example

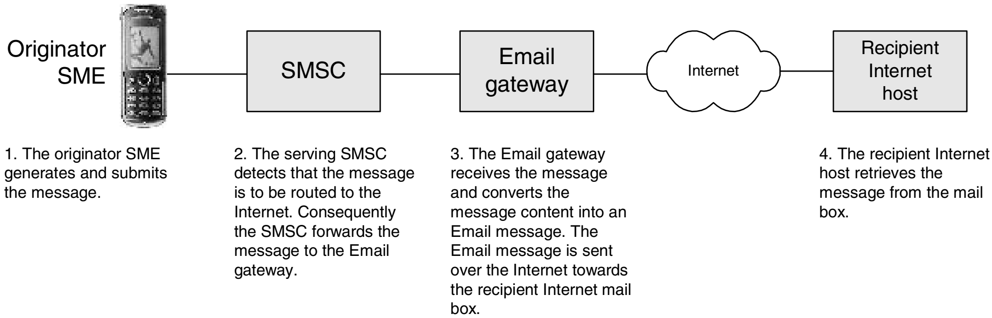


Figure 3.45 Process of sending a message to an Internet user

Table 3.37 TPDU parameters for Internet interworking

TPDU parameter	Value
`TP-Protocol-Identifier`	Internet electronic mail (0x32)
`TP-Destination-Address`	Gateway address

gateway as shown in the architecture depicted in Figure 3.45. An originator SME can indicate that a message has to be delivered to an Internet recipient by setting specific values for the parameters listed in Table 3.37. The process of sending an SMS message to the Internet domain is summarized in Figure 3.45.

The Email gateway may also convert an Email message to an SMS message. The conversion process consists of incorporating the Email message content in the `TP-User-Data` parameter of the SMS message TPDU. For this purpose, two methods have been developed. The first method is a *text-based method* consisting of inserting the Email message (RFC 822 header and footer) directly into the text section of the `TP-User-Data` parameter. The second method, called the *information element-based method*, consists of using a specific information element for separating the Email header from the Email body in the text part of the `TP-User-Data` parameter.

3.20.1 Text-Based Method

With this method, the content of the Email message is directly incorporated in text form in the `TP-User-Data` parameter. The text part representing the Email message content shall comply with the grammar rules listed in Table 3.38.

Fields `<to-address>` and `<from-address>` can take the two following forms:

`user@domain`

or

`User Name <user@domain>`

Table 3.38 Email text format

Email type	Message content
Internet to SMS – without subject – without real name	`[<from-address> <space>] <message>`
SMS to Internet – without subject – without real name	`[<to-address> <space>] <message>`

Internet to SMS
– with subject
– without real name
	`[<from-address>] (<subject>) <message>`
	or
	`[<from-address>] ## <subject># <message>`

SMS to Internet
– with subject
– without real name
	`[<to-address>] (<subject>) <message>`
	or
	`[<to-address>] ## <subject> # <message>`

Internet to SMS – with subject – with real name	`[<from-address>]#<real-name>##[<subject>]#<-message>`
SMS to Internet – with subject – with real name	`[<to-address>]#<real-name>##[<subject>]#<message>`

In this table, the following notation is used:
[] denotes optional fields.
<> delimits fields.
<space> denotes a single space character.

In the latter form, angle brackets are part of the address and are conveyed in the message. A message can contain multiple recipient addresses for the `<to-address>` field. In this case, addresses are separated by a comma:

```
user1@domain1,user2@domain2,user3@domain3
```

According to the grammar rules, the examples shown in Figure 3.46 are valid.

In SMS messages, the character "@" can be replaced by the character "*" and the character "_" (underscore) can be replaced by the character "$."

If the content of the Email message does not fit into one short message, then concatenation may be used. It is advisable to concatenate message segments with one of the concatenation information elements as described in Section 3.15.2. Alternatively, a text-based concatenation mechanism consists of adding the symbol "+" in specific positions in the SMS message. The first message segment contains the Email header as described above and ends with "+." Subsequent message segments start with "+" and end with "+." The last segment starts with a "+" but does not end with a "+." The Email message header is only inserted once in a concatenated message. The example in Figure 3.47 shows three message segments composing an Email message with text-based concatenation.

3.20.2 Information Element-Based Method

Another method for representing the content of an Email message in the `TP-User-Data` parameter consists of using a dedicated information element structured as shown in Table 3.39.

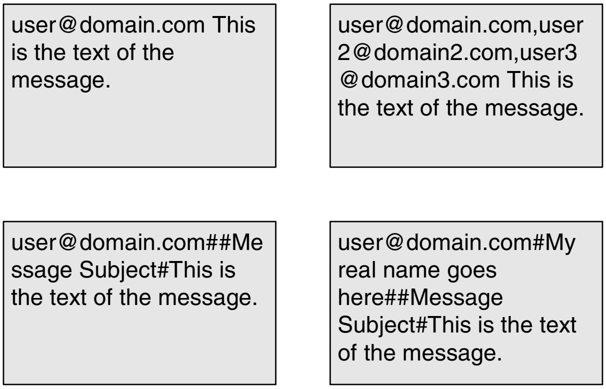


Figure 3.46 Examples of SMS Email messages

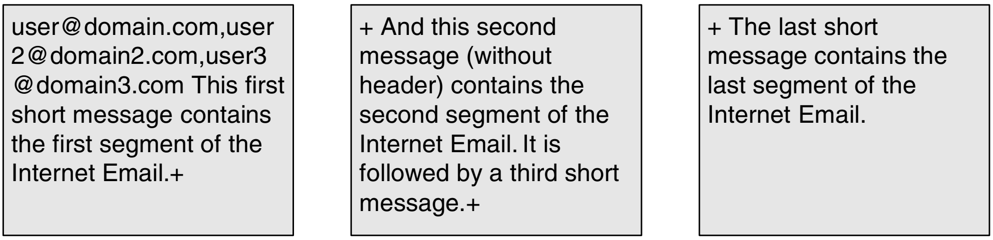


Figure 3.47 Example of a concatenated SMS Email message

The presence of this information element indicates that the text part of the `TP-User-Data` parameter contains an Email header and an optional Email body. The Email header and body composing the Email message are formatted according to the conventions published by the IETF in [RFC-822].

Table 3.39 IE/RFC 822 Email header

IEI	0x20 RFC 822 Email header	From Release 99
IEDL	0x01 (1 octet)	
IED	Octet 1	This octet represents the length of the Email header (or the length of the Email header fraction if used in a concatenated message). This allows to differentiate the Email header from the Email body in the text part of the `TP-User-Data` parameter. The length is expressed in terms of: • number of septets for GSM 7-bit default alphabet. • number of 16-bit symbols for UCS2. • number of octets for 8-bit encoding.

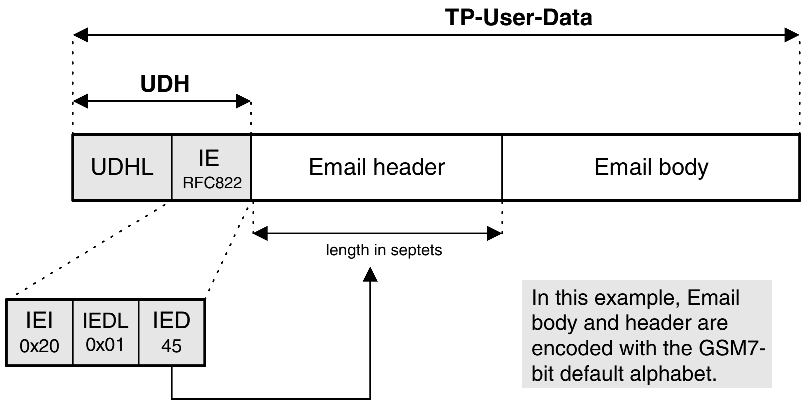


Figure 3.48 Information element RFC 822 Email header

The Email header shall always precede the Email body and both the header and body shall be encoded using the same character set (GSM 7 bit default alphabet, UCS2 or 8-bit data for ASCII).

If the Email message content does not fit into one message segment, then the concatenation mechanism defined in Section 3.15.2 can be used. In this situation, the information element "RFC 822 Email header" is inserted into each message segment composing the concatenated message.

Figure 3.48 shows how this information element can be used for separating a 45-septet Email header from the Email body.

3.21 Index of TPDU Parameters

Table 3.40 provides a list of all TPDU parameters and indicates whether or not the parameter is supported according to the TPDU type. If the parameter is supported, then an indication is given whether the parameter is mandatory or optional.

3.22 Pros and Cons of SMS

The incontestable advantage of the Short Message Service is that is has become a ubiquitous service in most GSM networks. All GSM handsets are capable of supporting the Short Message Service. A message can be sent from almost any GSM network and delivered to any other GSM subscriber attached to the same network, to another network in the same country, or even to a network in another country.

The main drawback of the Short Message Service is that only limited amounts of information can be exchanged between subscribers. In its simplest form, SMS allows 140 octets of information to be exchanged. Concatenation has been introduced to allow longer messages to be transmitted. Another drawback is that only text can be included in messages and this does not allow the creation of messages with content more compelling

Table 3.40 TPDU parameters

Abbreviation	Name	Short message submission *From MS to SMSC*	Submission report *From SMSC to MS*	Short message delivery *From SMSC to MS*	Delivery report *From MS to SMSC*	Status report *From SMSC to MS*	Command *From MS to SMSC*
`TP-CD`	`TP-Command-Data` Length 1 octet	No	No	No	No	No	Optional Variable position
`TP-CDL`	`TP-Command-Data-Length` Length 1 octet	No	No	No	No	No	Mandatory Variable position
`TP-CT`	`TP-Command-Type` Length 1 octet	No	No	No	No	No	Mandatory Octet 4
`TP-DA`	`TP-Destination-Address` Length 2...12 octets	Mandatory From octet 3	No	No	No	No	No
`TP-DCS`	`TP-Data-Coding-Scheme`	Mandatory Variable position	Optional Variable position	Mandatory Variable position	Optional Variable position	Optional Variable position	No
`TP-DT`	`TP-Discharge-Time` Length 7 octets	No	No	No	No	Mandatory Variable position	No
`TP-FCS`	`TP-Failure-CauSe` Length 1 octet (Integer)	No	Mandatory Octet 2	No	Mandatory Octet 2	No	No
`TP-MMS`	`TP-More-Message-to-Send` Length 1 bit	No	No	Mandatory Octet 1 / Bit 2	Mandatory Octet 1 / Bit 2	No	No
`TP-MN`	`TP-Message-Number` Length 1 octet	No	No	No	No	No	Mandatory Octet 5
`TP-MR`	`TP-Message-Reference` Length 1 octet (Integer)	Mandatory Octet 2	No	No	No	No	Mandatory Octet 2
`TP-MTI`	`TP-Message-Type-Indicator` Length 2 bits	Mandatory Octet 1 / Bits 0 and 1	Mandatory Octet 1 / Bits 0 and 1	Mandatory Octet 1 / Bits 0 and 1	Mandatory Octet 1 / Bits 0 and 1	Mandatory Octet 1 / Bits 0 and 1	Mandatory Octet 1 / Bits 0 and 1
`TP-OA`	`TP-Originator-Address` Length 2...12 octets	No	No	Mandatory From octet 2	No	No	No
`TP-PI`	`TP-Parameter-Indicator` Length 1 octet	No	**Positive** Mandatory Octet 2; **Negative** Mandatory Octet 3	No	**Positive** Manda-tory Octet 2; **Negative** Manda-tory From octet 3	Optional Variable position	No
`TP-PID`	`TP-Protocol-Identifier` Length 1 octet	Mandatory Variable position	**Positive** Mandatory Octet 10; **Negative** Optional Octet 11	Mandatory Variable position	**Positive** Optional Octet 11; **Negative** Optional Octet 11	Optional Variable position	Mandatory Octet 3

(*Continued*)

Abbreviation	Name	Short message submission *From MS to SMSC*	Submission report *From SMSC to MS*	Short message delivery *From SMSC to MS*	Delivery report *From MS to SMSC*	Status report *From SMSC to MS*	Command *From MS to SMSC*
`TP-RA`	`TP-Recipient-Address` Length 2...12 octets	No	No	No	No	Mandatory From octet 3	No
`TP-RD`	`TP-Reject-Duplicates` Length 1 bit	Mandatory Octet 1 / Bit 2	No	No	No	No	No
`TP-RP`	`TP-Reply-Path` Length 1 bit	Mandatory Octet 1 / Bit 7	No	Mandatory Octet 1 / Bit 7	No	No	No
`TP-SCTS`	`TP-Service-Center-Time-Stamp` Length 7 octets	Mandatory Variable position	**Positive** Mandatory From octet 3 **Negative** Mandatory From octet 4	No	No	Manda-tory Variable position	No
`TP-SRI`	`TP-Status-Report-Indicator` Length 1 bit	No	No	Optional Octet 1 / Bit 5	No	No	No
`TP-SRQ`	`TP-Status-Report-Qualifier` Length 1 bit	No	No	No	No	Mandatory Octet 1 / Bit 5	No
`TP-SRR`	`TP-Status-Report-Request` Length 1 bit	Optional Octet 1 / Bit 5	No	No	No	No	No
`TP-ST`	`TP-Status` Length 1 octet	No	No	No	No	Mandatory Variable position	No
`TP-UD`	`TP-User-Data` Variable length	Optional Variable position	Optional Variable position	Optional Variable position	Optional Variable position	Optional Variable position	No
`TP-UDHI`	`TP-User-Data-Header-Indicator` Length 1 bit	Optional Octet 1 / Bit 6	Optional Octet 1 / Bit 6	Optional Octet 1 / Bit 6	Optional Octet 1 / Bit 6	Optional Octet 1 / Bit 6	Optional Octet 1 / Bit 6
`TP-UDL`	`TP-User-Data-Length` Length 1 octet (Integer)	Mandatory Variable position	Optional Variable position	Mandatory Variable position	Optional Variable position	Optional Variable position	No
`TP-VP`	`TP-Validity-Period` Length 1 octet or 7 octet	Optional Variable position	No	No	No	No	No
`TP-VPF`	`TP-Validity-Period-Format` Length 2 bits	Mandatory Octet 1 / Bit 3	No	No	No	No	No

than text. Furthermore, the lack of content support for SMS prevents the development of commercial applications based on SMS. To cope with these limitations, an application-level extension of SMS has been introduced in the standard. This extension, known as the Enhanced Messaging Service (EMS), leverages SMS by allowing subscribers to exchange messages containing elements such as melodies, pictures, and animations. At the transfer layer, EMS messages are transported in the same way as SMS messages.

The next chapter provides an in-depth description of the Enhanced Messaging Service.

3.23 Further Reading

[1] G. Peersman and S. Cvetkovic, The Global System for Mobile Communications Short Message Service, *IEEE Personal Communications Magazine*, June, 2000.

4

Enhanced Messaging Service

Without any doubt, the Short Message Service has been a tremendous commercial success. However, SMS traffic growth started to slow down. In 2000, to prevent this slowdown, several mobile manufacturers, mobile operators, and third party vendors decided to give a further breath to SMS by collaborating on the development of an application-level extension of SMS: the Enhanced Messaging Service (EMS). EMS supersedes SMS capabilities by allowing the exchange of rich-media messages containing text with pictures, melodies, animations, etc. Standardization work went on for almost 2 years to define and finalize EMS features. A close analysis of the standardization work and availability of EMS handsets in the market leads to the identification of two sets of EMS features. The first set of features is defined in 3GPP technical specifications Release 99 (with several updates in Release 4 and Release 5) whereas the second set of features is defined in 3GPP technical specifications Release 5. In this book, the first features set is described as *basic EMS* and the second features set is described as *extended EMS*. This chapter provides an in-depth description of the two sets of EMS features.

4.1 Service Description

4.1.1 Basic EMS

Basic EMS allows the exchange of rich-media messages. Note that basic EMS features were mainly introduced in [3GPP-23.040] Release 99, with several updates in Release 4 and Release 5. EMS messages can contain several of the following elements:

- text, with or without formatting (alignment, font size, and style)
- black-and-white bitmap pictures
- black-and-white bitmap-based animations
- monophonic melodies.

One of the characteristics of EMS is that graphical elements (pictures and animations) and melodies are always placed in the text at a specific position: that of a character. This method

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2

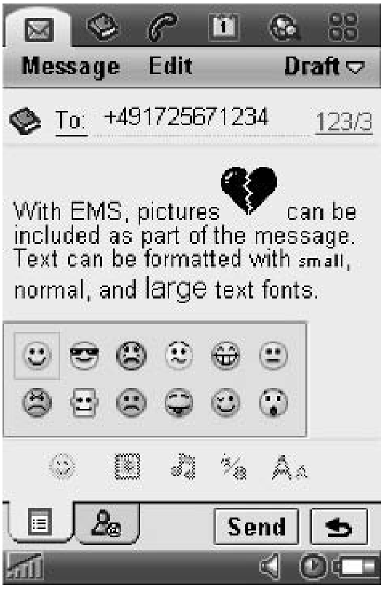


Figure 4.1 Example of rich-media EMS message

looks appropriate for graphical elements. However, the applicability of this positioning method is less appropriate for melodies. Indeed, a melody is rendered by the receiving device when its associated character position in the message text becomes visible to the subscriber.

In basic EMS, an element position is always expressed with a character position in the text of the message segment in which it is contained (relative positioning). In basic EMS, an element has to fit into a single message segment and cannot be positioned in the text part of another message segment. Such an element cannot be segmented and spread over several message segments. Consequently, its maximum size is equal to the maximum size of a message segment payload (<140 octets).

Figure 4.1 shows an example of rich-media message formatted with basic EMS. In addition, for the machine-to-person scenario (also known as content-to-person scenario), basic EMS includes the possibility of realizing a download service with the capability to limit the redistribution of downloaded elements.

EMS is an application-level extension of SMS using the concept of information element, as defined in the previous chapter, to convey EMS elements in messages. Each EMS element is associated with a dedicated information element. The payload of a dedicated information element is specifically structured to represent the corresponding EMS element (sequence of musical notes for a melody, bitmap for a picture, etc.). Information elements dedicated to basic and extended EMS are listed in Section 3.15.1 and fully described in this chapter.

4.1.2 Extended EMS

Basic EMS has many limitations preventing the development of attractive services, in particular for commercial and professional uses. To cope with these limitations, standardization work has been carried out on the development of an additional set of EMS features. This evolutionary step is designated in this book as extended EMS. Like basic EMS, extended EMS is also an application-level extension of SMS that builds on basic EMS by enabling the inclusion of large objects in messages. In addition to elements already supported in basic EMS, extended EMS also supports grayscale and color bitmap pictures and animations, monophonic and polyphonic melodies, vector graphics, etc. For this purpose, a framework has been designed to cope with the object size limitation of basic EMS. Compression of objects is also supported in extended EMS to allow the development of cost-effective services.

4.2 EMS Compatibility with SMS

Two forms of compatibility with SMS were considered for the design of basic EMS: forward compatibility and backward compatibility. In this book, a device is said to be EMS-enabled if it supports at least one EMS feature (basic or extended). An EMS-enabled device is said to be *backward compatible* with an SMS-only device if it is capable of interpreting messages sent from SMS-only devices. All EMS-enabled devices are backward compatible with SMS-only devices. The other way round, an SMS-only device is said to be *forward compatible* with an EMS-enabled device if it supports messages sent from EMS-enabled devices. Existing implementations of SMS-only devices are able to correctly interpret the text part of EMS messages and simply ignore EMS-specific elements such as images, animations, and sounds. It can therefore be said that SMS-only devices are forward compatible with EMS-enabled devices.

Note that some EMS-enabled devices have partial support for the whole set of EMS features. For instance, an EMS-enabled device may support sounds only while another may support sounds, images, and animations. If an EMS-enabled device receives a message containing an element which it is not capable of rendering, then the element is simply ignored by the device. Only the remaining part of the message is presented to the subscriber.

4.3 Basic EMS

Next sections provide a detailed description of the first set of EMS features, covered under the term "Basic EMS."

4.3.1 Formatted Text

Basic EMS allows text formatting instructions to be conveyed as part of the message. The appearance of the text can be formatted on the following aspects:

- alignment (language dependent–default, align left, center, align right)
- font size (normal, small, large)
- style (bold, italic, strikethrough, and underlined).

The information element dedicated to text formatting is structured as shown in Table 4.1.

If the text formatting length is set to 0, then text formatting instructions apply to all text characters from the start position. However, this may be overridden for a text section if a subsequent text formatting information element follows.

Table 4.1 IE/text formatting

IEI	0x0A Text formatting			From Release 99
IEDL	0x03 (3 octets)			
IED	Octet 1	**Start position** This octet represents the position of the first character of the message text to which the text formatting instructions are to be applied		
	Octet 2	**Text formatting length** This octet represents the number of characters to which the text formatting instructions are to be applied		
	Octet 3	**Formatting mode** This octet specifies how the associated text is to be formatted. The structure of the octet is as follows:		
		Bit 1	**Bit 0**	**Alignment**
		0	1	Align left
		0	1	Center
		1	0	Align right
		1	1	Language dependent (default)
		Bit 3	**Bit 2**	**Font size**
		0	0	Normal (default)
		0	1	Large
		1	0	Small
		1	1	Reserved
		Bit 4	**Bold**	
		0	Bold off	
		1	Bold on	
		Bit 5	**Italic**	
		0	Italic off	
		1	Italic on	
		Bit 6	**Underlined**	
		0	Underlined off	
		1	Underlined on	
		Bit 7	**Strikethrough**	
		0	Strikethrough off	
		1	Strikethrough on	

In the conflicting situation where several text formatting information elements apply to the same text section, text formatting instructions are applied in the sequential order of occurrence of corresponding information elements.

Note that this information element has been enhanced in extended EMS with the support of text foreground and background colors. This enhanced information element has an additional octet (octet 4 of IED, value 0x04 is assigned to IEDL). The description of the

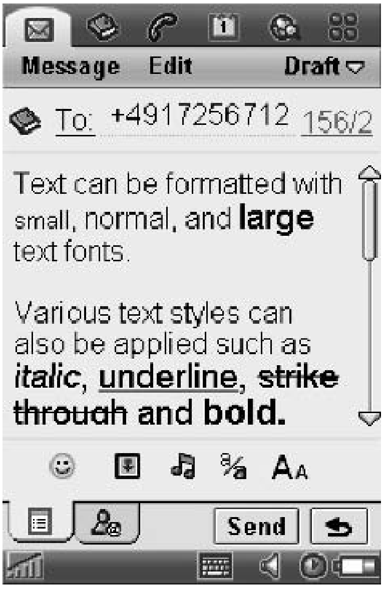


Figure 4.2 Example of message with text formatting instructions

enhanced text formatting information element is given in Section 4.4.18. Figure 4.2 shows a message containing some formatted text. Figure 4.3 presents the content of a TPDU containing text formatting instructions.

4.3.2 Pictures

In basic EMS, three types of black-and-white bitmap pictures can be included in messages as listed in Table 4.2.

The first two types are used for the representation of pictures with predefined dimensions whereas the last type is for the representation of pictures for which dimensions can be configured by the message originator. Each picture format is associated with a dedicated information element for which the identifier is provided in Table 4.2. The bitmap of each picture is encoded in the data part of the associated information element.

4.3.2.1 Large Picture

The information element for a large picture (32×32 pixels) is structured as given in Table 4.3. The bitmap structure for a large picture is shown in Figure 4.4. The size for a large picture (32×32) is 129 octets (including position and bitmap, excluding IEI and IEDL fields).

4.3.2.2 Small Picture

Another type of picture that can be used in basic EMS is for the representation pictures of 16×16 pixels. The associated information element is structured as given in Table 4.4.

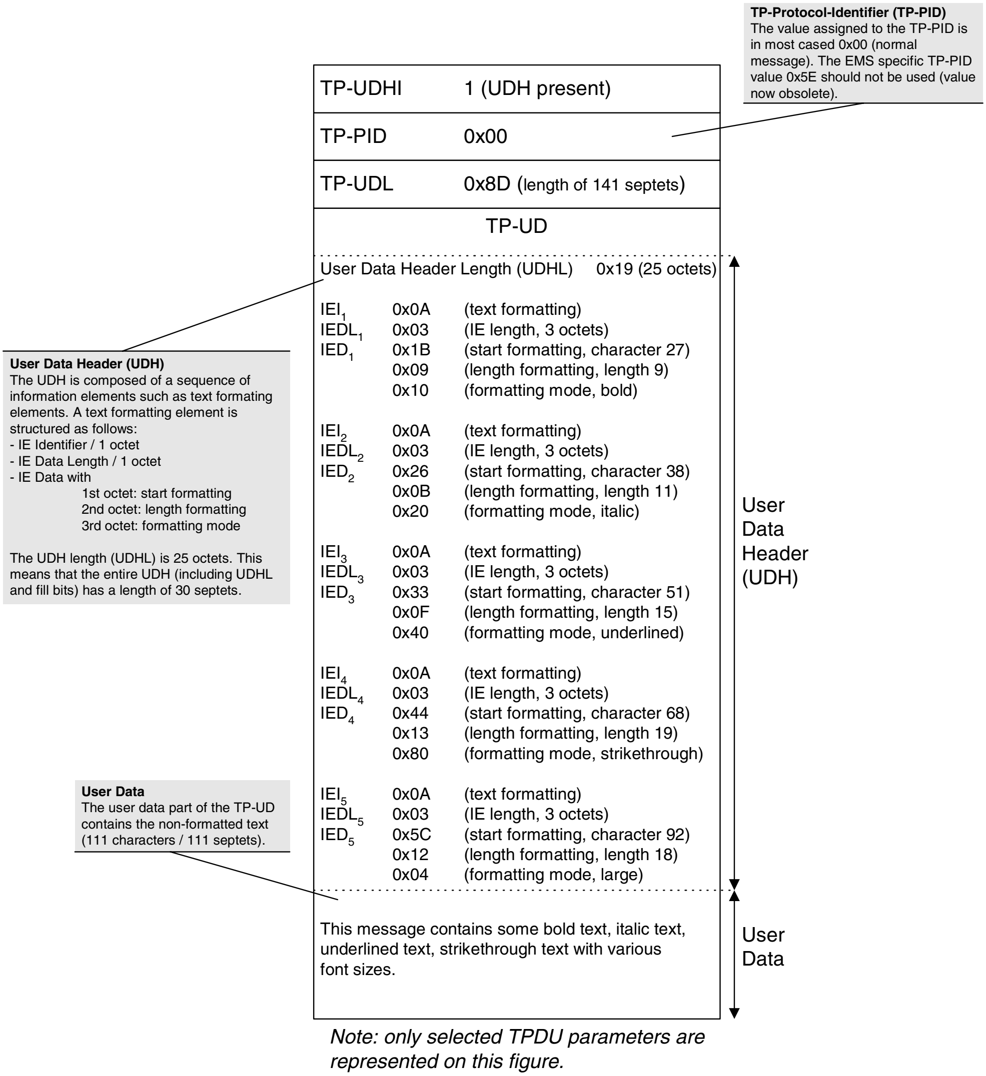


Figure 4.3 Text formatting information element/example

Table 4.2 Basic EMS bitmap pictures

IE identifier	Description	Dimensions	Bitmap size
0x10	Large picture	32 × 32 pixels	128 octets
0x11	Small picture	16 × 16 pixels	32 octets
0x12	Variable-size picture	Variable	Variable

Table 4.3 IE/large bitmap picture

IEI	0x10 Large picture (32 × 32)	From Release 99
IEDL	0x81 (129 octets)	
IED	Octet 1	**Large picture position** This octet represents the position in the text where the large picture is to be placed.
	Octets 2...129	**Large picture bitmap** This sequence of octets represents the bitmap of the large picture. The four first octets are used to encode the first row (top picture row), the four following octets are used to encode the immediately following picture row. For a 32 pixel row, the 8 bits of the first octet represent, respectively, the 8 first pixels (pixels at the extreme left of the picture), the 8 bits of the second octet represent respectively the next 8 pixels and so on. The most significant bit of an octet represents the pixel which is on the left side whereas the least significant bit represents the pixel on the right side. A bit at 0 means that the pixel is white and a bit at 1 means that the pixel is black. The mapping between the sequence of octets and the pixels is shown in Figure 4.4.

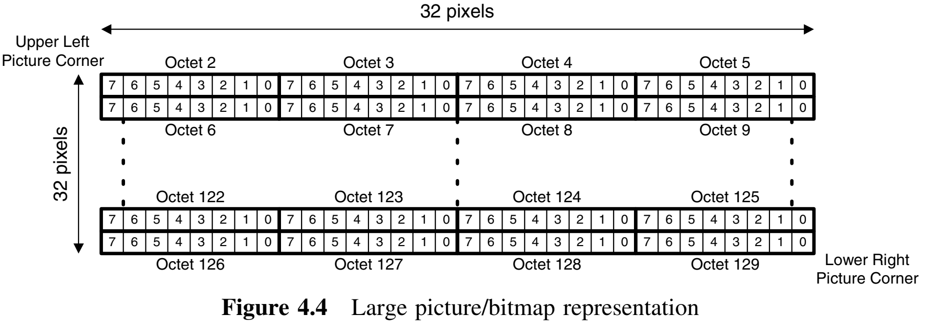


Figure 4.4 Large picture/bitmap representation

Table 4.4 IE/small bitmap picture

IEI	0x10 Small picture (16 × 16)	From Release 99
IEDL	0x21 (33 octets)	
IED	Octet 1	**Small picture position** This octet represents the position in the text where the small picture is to be placed.
	Octets 2...33	**Small picture bitmap** This sequence of octets represents the bitmap of the small picture. The two first octets are used to encode the first row (top picture row), the two following octets are used to encode the immediately following picture row. For a 16 pixel row, the 8 bits of the first octet represent respectively the 8 first pixels (pixels at the extreme left of the picture) and the 8 bits of the second octet represent, respectively, the next 8 pixels. The most significant bit of an octet represents the pixel which is on the left side whereas the least significant bit represents the pixel on the right side. A bit at 0 means that the pixel is white and a bit at 1 means that the pixel is black. The mapping between the sequence of octets and the pixels is shown in Figure 4.5.

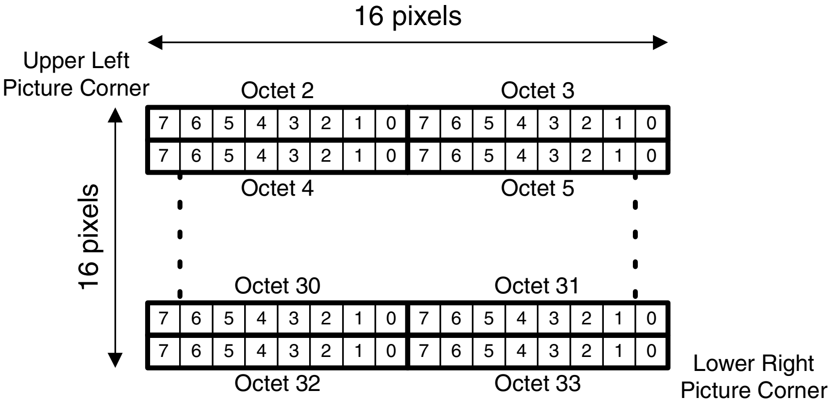


Figure 4.5 Small picture/bitmap representation

The bitmap structure for a small picture is as shown in Figure 4.5. The size for a small picture (16 × 16) is 33 octets (including position and bitmap, excluding IED and IEDL fields). Figure 4.6 shows how a picture of 16 × 16 pixels is encoded.

4.3.2.3 Variable-Size Picture

In addition to small and large pictures (predefined dimensions), a picture with customizable dimensions can also be conveyed as part of a message. The information element dedicated to *variable-size pictures* (width × height) is structured as shown in Table 4.5.

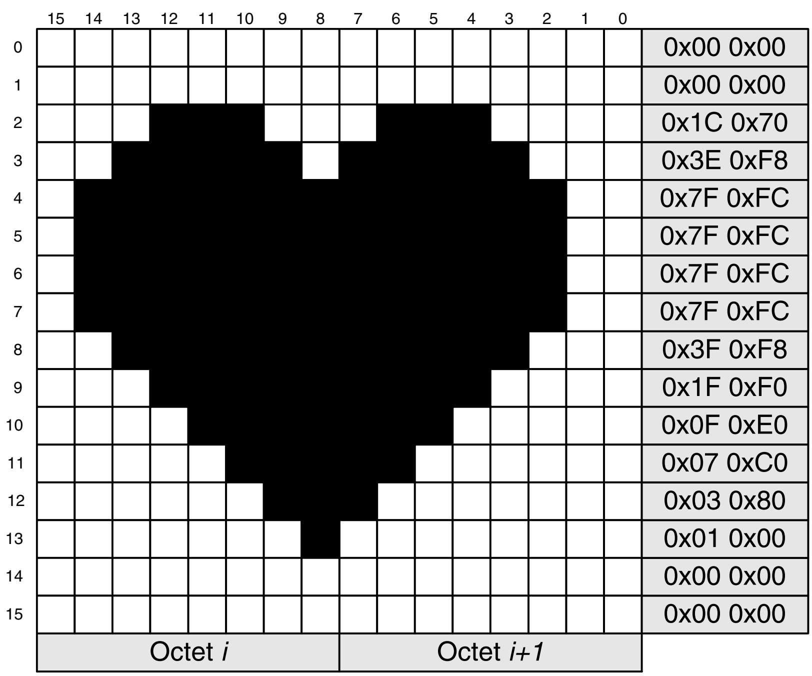


Figure 4.6 Example/small bitmap picture

Table 4.5 IE/variable-size bitmap picture

IEI	0x11 Variable-size picture (width × height)	From Release 99
IEDL	variable	
IED	Octet 1	**Variable-size picture position** This octet represents the position in the text where the variable-size picture is to be placed.
	Octet 2	**Horizontal dimension (width)** This octet represents the picture width. The unit is 8 pixels (for instance 2 represents a width of 16 pixels).
	Octet 3	**Vertical dimension (height)** This octet represents the picture height. The unit is 1 pixel.
	Octets $2 \ldots n$	**Variable-size picture bitmap** This sequence of octets represents the bitmap of the variable-size picture. The bitmap is coded row per row from top left to bottom right. The first octets represent the first pixel row of the picture and so on.

The picture size (including picture position, picture dimensions, and picture bitmap, excluding IEI and IEDL fields), expressed in octets, is calculated as follows:

$$PictureSize(octets) = width(in\ 8\ pixels) \times height(in\ pixels) + 3$$

In basic EMS, a picture can only fit into one message segment at most and considering this limitation, the following conditions must be fulfilled:

$PictureSize \leq 137$ octets for a one segment message.

$PictureSize \leq 132$ octets for a concatenated message (8-bit reference number).

$PictureSize \leq 131$ octets for a concatenated message (16-bit reference number).

Box 4.1 Obsolete method for stitching pictures

3GPP technical specifications Release 99 and Release 4 define a method for segmenting a large picture (larger than a message segment) to cope with the size limitations of basic EMS. With this method, if multiple pictures with the same width are received side by side, then they are stitched together horizontally to form a larger picture. Similarly, if multiple series of stitched pictures of the same resulting width are received, separated by a carriage return only, then the multiple series of pictures are stitched together vertically to form an even larger picture. It has to be noted that very few handsets on the market support this stitching method, therefore it is not recommended to generate content based on this method. The User Prompt Indicator defined in Section 4.3.5 should be used instead. Another alternative is to use extended EMS in which elements do not have the size limitations of basic EMS.

4.3.3 Sounds

With basic EMS, two types of sounds can be represented in messages: predefined and user-defined sounds.

A *predefined sound* is a commonly used sound for which the sound representation (sequence of notes) is included by default in the EMS-enabled device supporting this feature. Consequently, the sequence of notes of the predefined sound does not require to be included as part of the message. A reference to the predefined sound in the message is sufficient. The advantage of using predefined sounds is that the associated information element requires a very limited amount of message space. Available predefined sounds are given below:

- chime high
- chime low
- ding
- claps
- tada
- notify
- drum
- fanfare
- chord high
- chord low.

A *user-defined sound* is a sound for which the sequence of notes is entirely inserted in a message segment. With basic EMS, a user-defined sound is represented in the iMelody format and has a maximum size of 128 octets. The iMelody format is used for representing monophonic sounds/melodies (several notes cannot be rendered at the same time). It has to be noted that 128 octets are usually not enough for defining a sound of more than a few seconds (this explains why the word 'sound' is employed here instead of "melody"). Extended EMS allows the definition of longer monophonic melodies or polyphonic melodies.

4.3.3.1 Predefined Sounds

The information element dedicated to predefined sounds is structured as shown in Table 4.6. Predefined sounds are implemented in a manufacturer-specific manner. This means that these sounds may be rendered differently on devices produced by different manufacturers (polyphonic or monophonic sounds, short or long sounds, etc.).

4.3.3.2 User-Defined Sounds

The information element dedicated to user-defined sounds is structured as shown in Table 4.7.

Table 4.6 IE/predefined sounds

IEI	0x0B Predefined sound	From Release 99
IEDL	0x02 (2 octets)	
IED	Octet 1	**Predefined sound position** This octet represents the position in the text where the predefined sound is to be placed.
	Octet 2	**Predefined sound identification** This octet represents the identification of the predefined sound to be played. The following values can be assigned to this octet: **Id** — **Description** 0 — Chimes high 1 — Chimes low 2 — Ding 3 — TaDa 4 — Notify 5 — Drum 6 — Claps 7 — Fanfare 8 — Chord high 9 — Chord low *Other* — *Reserved*

A user-defined sound consists of a sequence of instructions. Instructions include the execution of notes and the definition of additional information such as composer name, sound name, volume, etc. Such a sequence of instructions is represented in the iMelody format. The BNF grammar of the iMelody format is provided in Appendix D. The iMelody format is a monophonic sound format originally developed by the IrDA Forum in [IRDA-iMelody]. An iMelody sound is composed of an iMelody *sound header*, an iMelody *sound body*, and an iMelody *sound footer* as shown in Figure 4.7.

Table 4.7 IE/user-defined sound

IEI	0x0C User-defined sound	From Release 99
IEDL	Variable length	
IED	Octet 1	**User-defined sound position** This octet represents the position in the text where the user-defined sound is to be placed.
	Octets $2 \ldots n$	**User-defined sound code** Octets $2 \ldots n$ represent the iMelody representation of the user-defined sound.

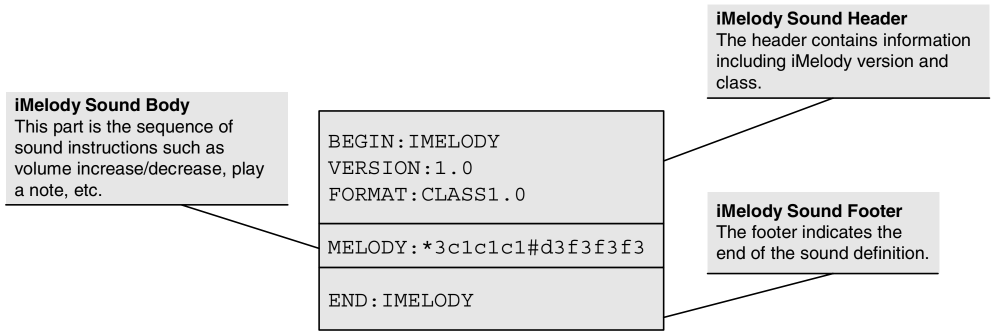


Figure 4.7 iMelody example

iMelody Sound Header

The default alphabet for coding an iMelody is UTF8 (as defined in Appendix C). Each header parameter is coded over one line. The value assigned to a parameter is separated by column from the parameter name and is placed on its right. The presence of a parameter in a sound header is either mandatory or optional. Table 4.8 provides an exhaustive list of sound header parameters.

Box 4.2 Recommendation for the creation of iMelody sounds

In order to ensure that the melody is properly interpreted by the largest number of EMS-enabled devices, it is recommended to specify the header parameters in the melody in the order indicated in the first column of Table 4.8.

iMelody Sound Body

The melody body is composed of at least one command. Five command types can be mixed in order to build the melody. These command types are shown in Tables 4.9–4.13. In addition to the five commands defined in the tables, a sequence of commands can be repeated more than once. For this purpose, the sequence to be repeated is inserted in the iMelody body between a pair of brackets:

```
(<sequence-of-commands> @ <repeat-count> [<volume-modifier>] )
```

where <sequence-of-commands> is the sequence of commands to be repeated, <repeat-count> is the number of times the sequence is to be repeated. It can take values such as “0” for repeat forever, “1” for play once, “2” for play twice, and so on.

Table 4.8 iMelody commands

Parameter name	Description	Status	Example
1 BEGIN	Tag identifying the type of melody. In this case, the iMelody format.	●	BEGIN: IMELODY
2 VERSION	The version of the melody in the format "**x.y**"	●	VERSION: 1.2
3 FORMAT	The class of the melody. The class is represented by concatenating the word "**CLASS**" with a version number in the format "**x.y**."	●	FORMAT: CLASS1.0
4 NAME	The name of the melody as a string of characters.	○	NAME: MyMelody
5 COMPOSER	The name of the melody's composer as a string of characters.	○	COMPOSER: MyName
6 BEAT	The beat value is expressed as a number of beats per minute (bpm). The value to be assigned to this parameter ranges from 25 to 900 (decimal values). This parameter is optional, if not specified, then notes are usually played at 120 bpm.	○	BEAT: 63 (means that a 1/4 note as a duration of 60/63 = 0.95 seconds)
7 STYLE	Three styles can be specified for playing the melody: "**S0**:" natural style (with rests between notes) "**S1**:" continuous style (no rest between notes) "**S2**:" staccato style (shorter notes and longer rest period) If no style is defined in the melody definition, then the natural style is used as the default style.	○	STYLE: S1
8 VOLUME	A volume level on a scale of 16 levels can be specified as part of the sound header. Volume level identifiers range from "**V0**" to "**V15**." "**V0**" means volume off whereas "**V15**" is the maximum volume level. All other values refer to intermediary volume levels and "**V7**" is the default volume level if this parameter has been omitted from the melody definition.	○	VOLUME: V10

Note: status is ● when the associated parameter is mandatory in the melody, otherwise the parameter is optional and the associated status is marked ○.

<volume-modifier> can be assigned values "V+" to increase the volume by one level each time the sequence of notes is repeated, or "V−" to decrease the volume by one level each time the sequence of notes is repeated. This parameter is optional. With iMelody class 1.0, a sequence of commands to be repeated cannot be nested into a sequence of commands which is already designed to be repeated.

Table 4.9 iMelody/note command

Command	Note command
Description	The note command is used to play a note.
Format	The note command is structured as follows: *<note><duration>*[*<duration-specifier>*] where *<note>* can be either a basic note such as “**c**,” “**d**,” “**e**,” “**f**,” “**g**,” “**a**,” or “**b**” a flat note such as “**&d**,” “**&e**,” “**&g**,” “**&a**,” or “**&b**” a sharp note such as “**#c**,” “**#d**,” “**#f**,” “**#g**,” or “**#a**” where *<duration>* can take the following values: “**0**” for full note “**1**” for 1/2 note “**2**” for 1/4 note “**3**” for 1/8 note “**4**” for 1/16 note “**5**” for 1/32 note where *<duration-specifier>* can take the following values: “**.**” for dotted note “**:**” for double-dotted note “**;**” for 2/3 length. The *<duration-specifier>* is an optional parameter in the note command.

Table 4.10 iMelody/octave command prefix

Command	Octave command
Description	The octave command indicates that the immediately following note is associated with a specific octave.
Format	The octave command is a two-character command which takes one of the following values: “***0**” for 55 Hz “***1**” for 110 Hz “***2**” for 220 Hz “***3**” for 440 Hz “***4**” for 880 Hz “***5**” for 1760 Hz “***6**” for 3520 Hz “***7**” for 7040 Hz “***8**” for 14080 Hz
Note	At least one note command is to be inserted between two octave commands. If the octave level has not been specified, notes are played with the default octave level of 880 Hz (value “***4**”).

Table 4.11 iMelody/silence command

Command	Silence command
Description	The silence command is used to insert a silence between two notes.
Format	The silence command is structured as follows: “**r**” *<duration>*[*<duration-specifier>*] where *<duration>* and *<duration-specifier>* can take the same values as the note command *<duration>* and *<duration-specifier>*.
Example	“**r1:**” means a rest for half a double-dotted note.

Table 4.12 iMelody/volume command

Command	Volume command
Description	The volume command is used to change the volume level at which the following notes are going to be played.
Format	The silence command can be assigned one of the following values: "**V1**" to "**V15**" to indicate that the following notes are to be played with, respectively, a low volume level to a high volume level. The silence command can also be assigned the values "**V+**" or "**V−**" to indicate, respectively, one volume level up or one volume level down.

Table 4.13 iMelody/special effect command

Command	Special effect commands
Description	A special effect command is used to light on/off a led on the mobile handset, activate/deactivate the vibrator or light on/off the screen backlight.
Format	Below is the list of special effect commands: "**ledon**" for lighting on a mobile handset led "**ledoff**" for lighting off a mobile handset led "**vibeon**" for activating the vibrator "**vibeoff**" for deactivating the vibrator "**backon**" for lighting on the screen backlight "**backoff**" for lighting off the screen backlight

iMelody Sound Footer

The iMelody sound is always ended by the sequence of characters: "END:IMELODY."

iMelody Versions

At the time of writing, three versions of the iMelody format have been publicly released. Table 4.14 provides a list of the differences between the three iMelody format versions.

Table 4.14 Differences between versions of the iMeldody format

iMelody version	Supported commands
iMelody v1.0	• Definition of header information, body commands, and footer.
iMelody v1.1	• Corrections for relative and absolute volume commands in the iMelody body • Corrections for the support of commands for lighting on and off the screen backlight.
iMelody v1.2	• Correction for the command for executing more than once a group of commands.

Box 4.3 Short iMelody sound (without header and footer)

Headers and footers are mandatory in the iMelody format. However, in the context of EMS, they usually convey information that is not always required by the receiving device. This can be seen as a waste of message space. Devices can sometimes interpret correctly an iMelody composed of the iMelody body only (iMelody header and footer are omitted). Note that such a melody does not conform to the iMelody grammar and may not be interpreted correctly by all EMS-enabled devices. It is, therefore, not recommended to use this shorter form of iMelody sounds unless one can ensure that the receiving device supports it.

Box 4.4 Ambiguity in the iMelody grammar definition

The iMelody specification does not clearly define the signification of the octave prefix/command. It is unclear whether the octave prefix/command should apply only to the immediately following note or to all the notes until another octave prefix/command is encountered. After lengthy discussions between device manufacturers regarding the potential interoperability issue, it was decided to apply the octave prefix only to the immediately following note. Unfortunately, some handsets have been released on the market with the wrong interpretation of the iMelody format.

4.3.4 Animations

With basic EMS, several types of animations can be included in messages as shown in Table 4.15, which lists the size of each animation type (including animation position, predefined animation identification or animation bitmap pictures; excluding IEI and IEDL fields).

4.3.4.1 Predefined Animations

A *predefined animation* is a commonly known animation whose definition is supported by all EMS-enabled handsets. Consequently, the full representation (animated sequence of pictures) of such predefined animations is not transferred as part of messages. Only the identification of the predefined animation is included in messages. As with predefined sounds, the main advantage of using a predefined animation is its low requirement in terms of message space (four octets for the information element including one octet for the reference to the predefined animation). The information element used to insert a predefined animation in a message is structured as shown in Table 4.16.

Table 4.15 Animations in basic EMS

IE identifier	Description	Dimensions	Animation size (octets)
0x0D	Predefined animation	*Not applicable*	2
0x0E	Large user-defined animation	4 pictures(16 × 16 pixels)	129
0x0F	Small user-defined animation	4 pictures (8 × 8 pixels)	33

Table 4.16 IE/predefined animations

IEI	0x0D Predefined animation			From Release 99, new predefined animations added in Release 4
IEDL	0x02 (2 octets)			
IED	Octet 1	**Predefined animation position** This octet represents the position in the text where the predefined animation shall be displayed.		
	Octet 2	**Predefined animation identification** This octet represents the identification of the predefined animation to be displayed. The following values can be assigned to this octet:		
		Id	**Description**	**Introduced in:**
		0	I am ironic, flirty	Release 99
		1	I am glad	Release 99
		2	I am skeptic	Release 99
		3	I am sad	Release 99
		4	Wow !!!	Release 99
		5	I am crying	Release 99
		6	I am winking	Release 4
		7	I am laughing	Release 4
		8	I am indifferent	Release 4
		9	In love/kissing	Release 4
		10	I am confused	Release 4
		11	Tongue hanging out	Release 4
		12	I am angry	Release 4
		13	Wearing glasses	Release 4
		14	Devil	Release 4
		Other	*Reserved*	

The six first predefined animations (id 0–5) are defined in EMS Release 99 [3GPP-23.040]. The nine additional animations (id 6–14) are defined in EMS Release 4. If an EMS-enabled device conforms to EMS Release 99 only, then only the six first predefined animations are rendered.

Predefined animations are implemented in a manufacturer-specific manner. This means that animations may be rendered differently on devices produced by different manufacturers.

4.3.4.2 User-Defined Animations

Unlike predefined animations, *user-defined animations* are animations for which the representation of animated sequences of pictures is included in messages. A user-defined animation consists of a sequence of four pictures with identical dimensions. No timing information can be specified to define the display duration of each picture. Consequently, a handset usually represents an animation by looping over the four pictures. The display duration for each picture is often arbitrarily set to a few hundreds of milliseconds by the device manufacturer. User-defined animations have a maximum size of 128 octets. Furthermore, two types of user-defined animations can be inserted in messages: small user-defined animations (4 pictures 8×8 pixels) and large user-defined animations

Table 4.17 Examples of user-defined animations/reproduced by permission of Alcatel Business Systems

Animation no 1	**Animation no 2**
	
(1) (2) (3) (4)	(1) (2) (3) (4)

(4 pictures 16 × 16 pixels). Examples of user-defined animations are given in Table 4.17. The information element used to insert a large user-defined animation in a message is structured as shown in Table 4.18. The information element used to insert a small user-defined animation in a message is structured as shown in Table 4.19.

In basic EMS, there is no means for defining an animation whose dimensions are configured by the message originator.

Table 4.18 IE/large user-defined animation

IEI	0x0E Large animation (4 pictures 16 × 16 pixels)	From Release 99
IEDL	0x02 (129 octets)	
IEDL	Octet 1	**Large animation position** This octet represents the position in the text where the large animation is to be displayed.
	Octets 2...129	**Large animation definition** These octets represent a sequence of four bitmap pictures as defined in Section 4.3.2. The picture dimensions are 16 × 16 pixels. Octets 2...33 First picture (16 × 16 pixels). Octets 34...65 Second picture (16 × 16 pixels). Octets 66...97 Third picture (16 × 16 pixels). Octets 98...129 Fourth picture (16 × 16 pixels).

Table 4.19 IE/small user-defined animation

IEI	0x0F Small animation (4 pictures 8 × 8 pixels)	From Release 99
IEDL	0x21 (33 octets)	
IED	Octet 1	**Small animation position** This octet represents the position in the text where the small animation is to be displayed.
	Octets 2...33	**Small animation definition** These octets represent a sequence of four bitmap pictures as defined in Section 4.3.2. The picture dimensions are 8 × 8 pixels. Octets 2...9 First picture (8 × 8 pixels). Octets 10...17 Second picture (8 × 8 pixels). Octets 18...25 Third picture (8 × 8 pixels). Octets 26...33 Fourth picture (8 × 8 pixels).

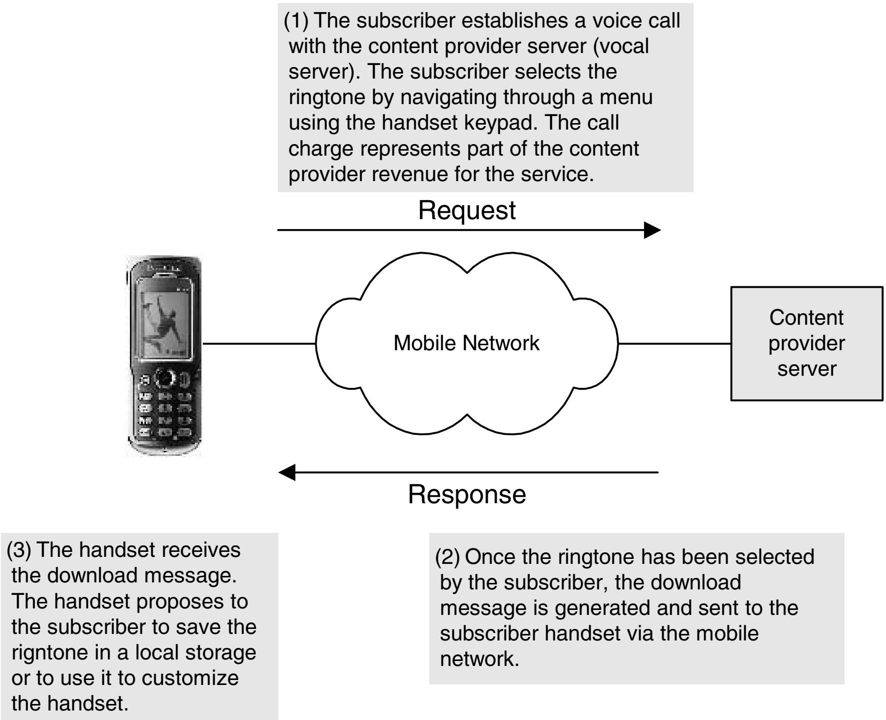


Figure 4.8 Interactions for the download service

4.3.5 User Prompt Indicator

Downloading ringtones and background pictures has become very popular for the customization of handsets. Such a service can rely on SMS/EMS. For this purpose, the object being downloaded is contained in a *download message,* typically composed of several message segments. A simplified network configuration for the support of a download service and the necessary steps to download a ringtone, are shown in Figure 4.8.

Before the introduction of EMS, each manufacturer willing to support a download service had to specify its own proprietary object formats and its own proprietary transport method. Such proprietary download services presented a major drawback: the content of the download message had to be specifically generated according to the characteristics of the receiving device. This made the deployment of generic and widely available download services impossible.

With EMS, a standard transport protocol is used in the form of SMS. Additionally, a set of standard object formats are defined as part of basic EMS. Manufacturers, operators, and content providers quickly realized that EMS can be used appropriately as the building block for the realization of a generic and standard download service. However, basic EMS presents one limitation: it can transport only small objects. This is a strong limitation for a download service, which requires the download of potentially large objects (up to 512 octets). In order to cope with this EMS limitation, a basic application-level object segmentation and reconstruction method has been introduced in basic EMS. This method, allowing the

Table 4.20 IE/user prompt indicator

IEI	0x13 User Prompt Indicator	From Release 4
IEDL	0x01 (1 octets)	
IED	Octet 1	**Number of associated information elements** This octet represents the number of object segments to be concatenated to reconstruct the downloaded object.

transport of large objects, is based on the use of an additional information element: the *User Prompt Indicator* (UPI). The purpose of this information element is twofold:

1. The UPI information element is used to indicate that a message (short message or concatenated message) contains an object being downloaded for the customization of the receiving handset.
2. The UPI information element is also used to indicate that the object being downloaded has been segmented over several message segments and needs to be reconstructed by the receiving device.

The structure of the UPI information element is as shown in Table 4.20.

4.3.5.1 UPI Management

Upon receipt of the download message, the receiving device can propose to the subscriber:

- to save the downloaded object locally in order to be used later for composing new messages;
- to customize the device (change the default ringtone, change switch-on and switch-off animations, etc.).

The set of actions that may be performed on a downloaded object obviously depends on the object format and on the capabilities of the receiving device. The UPI information element is used in the machine-to-person scenario. This information element is usually not supported in the person-to-person scenario. The text that may be placed in a download message is usually not presented to the subscriber. It is, therefore, not recommended to add text as part of a download message.

4.3.5.2 UPI Segmentation and Reconstruction

The UPI reconstruction method can only be applied to melodies (user-defined) and pictures (variable, large, and small).

- *Melody reconstruction*: melody segments are stitched together according to the concatenation index number. Only the header and footer of the first melody segment are kept in the reconstructed melody.

- *Picture reconstruction*: the height of all picture segments shall be identical, otherwise the UPI is ignored. Picture segments are stitched vertically in order to build a larger picture.

Figure 4.9 shows how a large picture can be segmented into three picture slices in a download message.

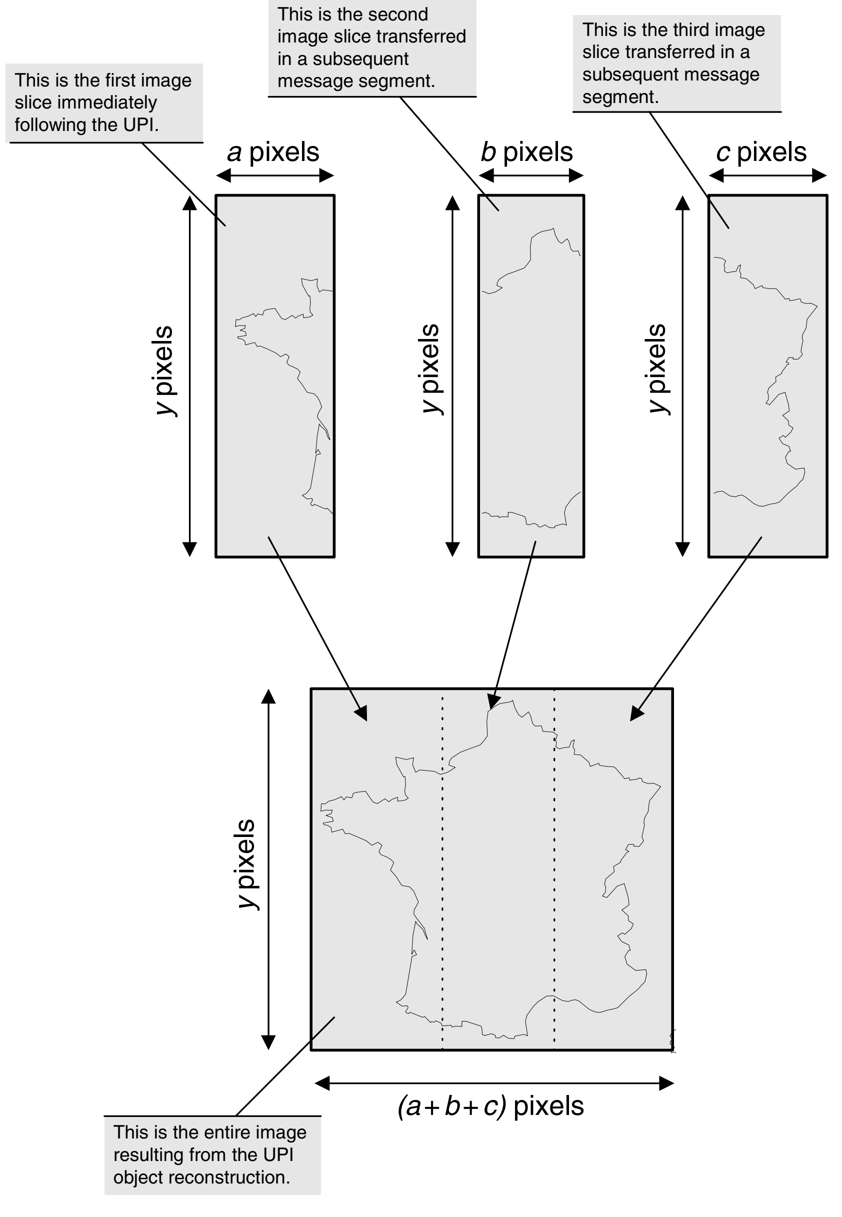


Figure 4.9 UPI picture segmentation (example)

4.3.6 Independent Object Distribution Indicator

The independent *Object Distribution Indicator* (ODI) is used to control the way the receiving SME can redistribute one or more objects contained in an EMS message. The major use case for the independent ODI is to control the redistribution of copyrighted content once it has been received by the subscriber. In the situation where the subscriber purchases copyrighted content, the content provider is able to indicate that one or more objects contained in the message cannot be redistributed via SMS. This means that the message cannot be forwarded by the receiving SME. This also means that objects can be extracted from the message but cannot be reused to compose new messages. For this purpose, information elements representing basic EMS objects, which must be limited for redistribution, are preceded by an independent ODI information element. One single ODI information element can be associated with one or more objects. The structure of the ODI information element is as shown in Table 4.21.

If an object is not associated with any object distribution indicator, then the object is considered as not being limited in its distribution. Although the independent ODI information element was not introduced before Release 5, this information element is considered in this book as a basic EMS feature. This is explained by the fact that this object distribution indicator is used to limit the distribution of basic EMS objects only. The extended EMS defines another method for limiting the distribution of extended EMS objects. In this book, this method is known as the *integrated ODI* and is presented in Section 4.4.1.

Figure 4.10 shows how the redistribution of two objects can be limited with the use of one independent ODI information element.

Table 4.21 Object distribution indicator

IEI	0x17 Object distribution indicator		From Release 5
IEDL	0x02 (2 octets)		
IED	Octet 1	**Number of associated information elements** This octet represents the number of information elements associated with the distribution indication. If set to 0, then the distribution indication applies to all information elements until the end of the message or until another ODI is encountered.	
	Octet 2	**Distribution status** The distribution status octet is coded as follows: **Bit 0** — **Distribution attribute** 0 — associated object(s) may be forwarded. 1 — associated object(s) shall not be forwarded by SMS. **Bit 1...7** — **Reserved for future use** Unused bits shall be set to 0.	

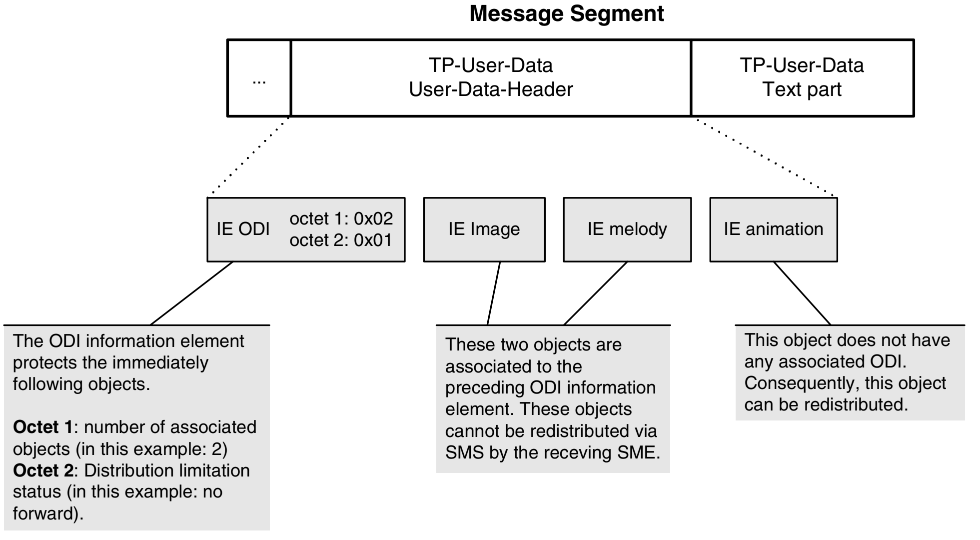


Figure 4.10 ODI/example

4.4 Extended EMS

In order to break the limitations of basic EMS, an extended version of EMS has been developed. Extended EMS features were mainly introduced in [3GPP-23.040] Release 5. In addition to all the features of basic EMS, extended EMS-enabled devices support the following features:

- *A framework for the support of extended objects*: in the framework, an extended object is either an object with a type supported in basic EMS (which no longer has the basic EMS size limitation) or an object of a new type defined in extended EMS. One of the main limitations of basic EMS resides in the impossibility of including large objects in messages. With basic EMS, concatenation can be used for building large messages with many objects but each single object is limited to fit into one message segment. With extended EMS, objects are no longer limited to the size of one message segment but may be segmented and spread over more than one message segment.
- *Compression of objects*: since the extended EMS framework allows large objects to be included in messages, the number of segments per message can become significantly high. In order to allow the development of cost-effective services, a method for compressing extended objects has been introduced in extended EMS.
- *Integrated Object Distribution Indicator* (ODI): it was shown in Section 4.3.6 that an independent ODI could be used for limiting the distribution of basic EMS objects. Similarly, an ODI tag has been integrated into the definition of each extended object. This tag is known, in this book, as the integrated ODI.
- *A new set of objects*: additional object formats are supported for the construction of extended EMS messages. The entire set of supported object formats is given below:

- black-and-white bitmap pictures (also supported in basic EMS)
- 4-level grayscale bitmap pictures
- 64-color bitmap pictures
- black-and-white bitmap animations (also supported in basic EMS)
- 4-level grayscale bitmap animations
- 64-color bitmap animations
- vCard data streams (used to define business cards)
- vCalendar data streams (used to define appointments, reminders, etc.)
- monophonic (iMelody) melodies (also supported in basic EMS)
- polyphonic (MIDI) melodies
- vector graphics.

- *Color formatting for text*: in basic EMS, text formatting is limited to changing the text alignment (left, right, and center), font style (bold, italic, underlined, strikethrough) and font size (small, normal, or large). In addition to these basic features, text background and foreground can also be colored with extended EMS.
- *Hyperlink*: the hyperlink feature allows the association of some text and/or graphical elements (pictures, animations, etc.) with a Uniform Resource Identifier (URI).
- *Capability profile*: many EMS-enabled devices have partial support for basic and extended EMS features. A mechanism in extended EMS allows SMEs to exchange their extended EMS capabilities. This enables SMEs (e.g., an application server) to format the content of messages according to what a specific recipient device is capable of rendering.

4.4.1 Extended Object Framework

The extended EMS framework allows a large object to be segmented and spread over more than one contiguous message segment. For this purpose, User-Data-Header information elements, introduced in Section 3.15, are used to avoid potential impacts on the network infrastructure for the deployment of extended EMS. It also enables compatibility with devices already available on the market.

With this framework, a large object is segmented into several parts and each part is included in the payload of a dedicated information element. Each information element, known as an *extended object information element*, is conveyed in one message segment with other information elements (SMS, basic and extended EMS) and optional text.

The representation of a large bitmap picture with three extended object IEs is shown in Figure 4.11. In the example, the first part of the picture is encoded in the first extended object IE. For a concatenated message, the maximum length of the picture segment in the first extended object is 124 octets.[1] In addition to containing the first picture segment, this first IE extended object also contains the extended object header. This extended object header is composed of the following parameters:

- *Object length*: this is the length of the entire extended object, expressed in octets. The part in the first extended object IE (excluding the extended object header) plus parts in

[1] Calculation: 140 octets (TP-UD size) − 1 octet (UDHL) − 6 octets (concat. IE, 16 bit reference number) − 9 octets (first extended object IE without payload) = 124 octets

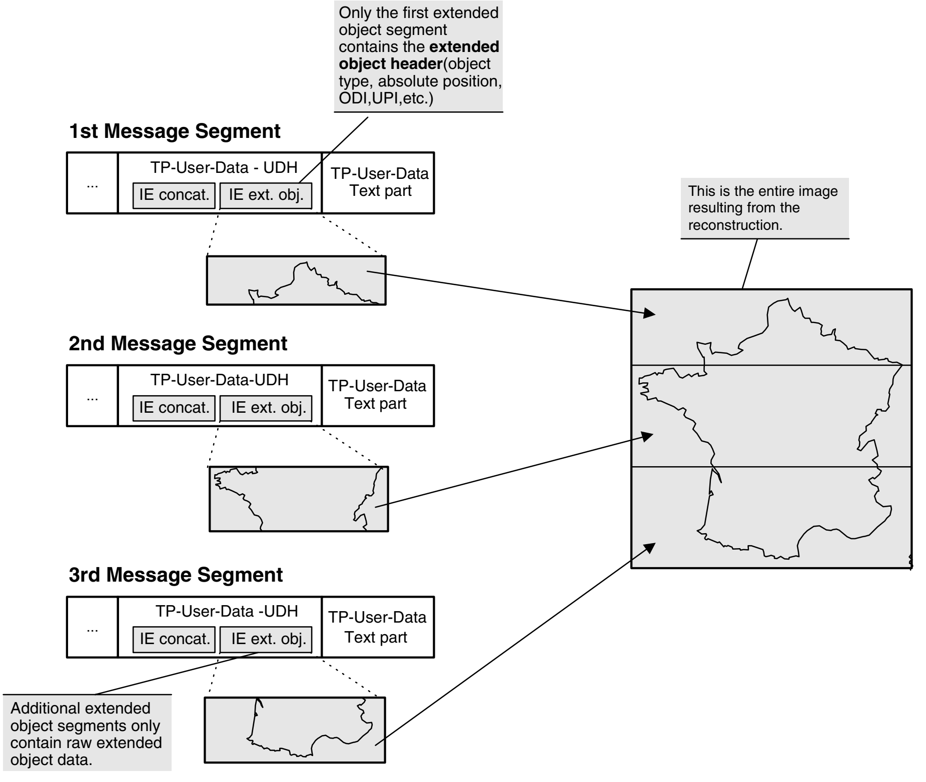


Figure 4.11 Extended object encoding/example

additional extended object IEs are taken into consideration for the calculation of the length. The knowledge of the object length is required by the receiving SME to be able to correlate extended objects with corresponding information elements. This information is required because additional extended object IEs do not refer to any particular object reference number (or object index number).

- *Object handling status*: the object handling status indicates (1) whether or not the associated object is part of a download message and (2) whether or not the object can be redistributed.
- *Object type*: the object type indicates which type of object is being conveyed (picture, animation, etc.).
- *Object position*: because an extended object can be segmented over several message segments, this object position indicates the absolute message text position where the object should be placed. Note that in basic EMS, only relative positions (positions in the text of the message segment where the object is inserted) can be specified.

If the large object cannot be conveyed using one single extended object IE, then additional extended object IEs may be used. For each additional extended object IE, the extended object header is omitted. Consequently, any additional extended object IE contains only raw data from the extended object remaining parts (parts that could not be included in the first extended object IE). Due to this, an additional extended object IE can contain an extended object segment for which the size can reach 131 octets.[2]

The structure of the first extended object IE (including the extended object header) is shown in Table 4.22. Octets 1...7 of this extended object IE represent the extended object header and are not taken into consideration for the object length calculation (see description of octets 2 and 3).

Table 4.22 IE/extended object (first)

IEI	0x14 Extended Object (first)		From Release 5
IEDL	Variable		
IED **Extended object header**	Octet 1	**Reference number** This octet represents the reference number of the extended object. This reference number may be used for re-inserting the object in another part of the message. Each extended object, in a message, shall have a unique reference number.	
	Octets 2 and 3	**Object length** These octets represent the length of the object expressed in octets. This length comprises the part of the object which is included in the first extended object information element (excluding extended object header) plus object parts contained in additional extended object information elements.	
	Octet 4	**Object handling status** This octet indicates how the object should be handled by the receiving SME. **Bit 0** **Integrated ODI** This bit indicates whether or not the object can be redistributed by SMS. 0 Object may be redistributed. 1 Object shall not be redistributed by SMS. If this bit is set to 1, then the associated object is handled as shown in Section 4.3.6. **Bit 1** **User prompt indicator** This bit indicates whether or not the object is part of a download message. 0 Part of a download message. 1 Not part of a download message. If this bit is set to 1, then the object is handled as shown in Section 4.3.5. Unused bits for this octet shall be set to 0.	
	Octet 5	**Extended object type** This octet indicates which type of object is being conveyed as part of this information element. Available object formats are: (dec.) (hex.) Object type — Refer to section 0 0x00 Predefined sound — 4.4.5 1 0x01 iMelody melody — 4.4.6 2 0x02 Black-and-white picture — 4.4.7 3 0x03 4-level grayscale picture — 4.4.8 4 0x04 64-color picture — 4.4.9	

(*Continued*)

[2] Calculation: 140 octets (TP-DU size) − 1 octet (UDHL) − 6 octets (concat. IE, 16-bit reference number) − 2 octets (extended object IE without payload and without extended object header) = 131 octets

Table 4.22 (*Continued*)

IEI	0x14 Extended Object (first)			From Release 5
IEDL	Variable			
	5	0x05	Predefined animation	4.4.10
	6	0x06	Black-and-white animation	4.4.11
	7	0x07	4-level grayscale animation	4.4.12
	8	0x08	64-color animation	4.4.13
	9	0x09	vCard object	4.4.14
	10	0x0A	vCalendar object	4.4.15
	11	0x0B	Vector graphic (WVG)	4.4.17
	12	0x0C	Polyphonic melody (MIDI)	4.4.16
	255	0xFF	Capability return	4.4.20
		Other	*Reserved*	
	Octets 6 and 7	**Extended object position** These octets represent the absolute character position in the message where the object is placed.		
	Octets 8...*n*	**Extended object definition** These octets represent the entire object definition or the first part of the object definition. If more than one message segment are required to convey the object, then the first object part is inserted in this information element and the remaining part(s) of the object is/are inserted in subsequent message segments (inserted as part of other extended object information elements 0x14).		

In the situation where more than one extended object IE have to be used to convey an extended object, the concatenation IE 16-bit reference number shall be used. The use of this concatenation IE, compared with the 8-bit reference number concatenation IE, helps reduce the probability of receiving conflicting concatenated message segments (see Box 3.1).

If the extended object is too large to fit into one message segment, then the first part of the object is inserted into a first extended object IE as defined in Table 4.22. The remaining part(s) of the extended object is/are inserted into subsequent message segments in additional extended object IEs as defined in Table 4.23. Note that the first and additional extended object information elements have the same IE identifier (value 0x14).

Table 4.23 IE/extended object (additional)

IEI	0x14 Extended object (additional)		From Release 5
IEDL	Variable		
IED	Octets 1 . . . *n*	These octets represent an additional extended object segment. Unlike the first extended object IE, an additional extended object IE does not contain an extended object header (reference number, object length, etc.).	

Box 4.5 Recommendation for a maximum size for extended EMS messages

A receiving SME is capable of interpreting a message composed of up to eight message segments. A message composed of more than eight message segments might not be interpreted correctly by all receiving SMEs. It is, therefore, highly recommended to limit the number of segments per message to eight.

Box 4.6 Recommendation for the use of integrated and independent ODIs

Two ODIs are available: an independent ODI (basic EMS) and an integrated ODI (extended EMS). The integrated ODI is part of the extended object header and the independent ODI is managed via an independent IE which is associated with one or more objects. If the object for which the distribution is limited is an extended object, then it is recommended to use the integrated ODI (resource gain). If the object for which the distribution is limited is not an extended object (basic EMS objects), then the use of the independent ODI is necessary. In this latter case, SMEs that do not support the ODI concept just ignore the independent ODI IE but interpret associated objects. In this situation, SMEs are able to freely distribute objects for which the message originator requested a limited distribution.

Figure 4.12 shows the encoding of a large black-and-white picture conveyed as part of a concatenated message composed of two segments.

4.4.2 Extended Object Reuse

In the extended object framework, each extended object is associated with a unique[3] *extended object reference number* (see extended object header). This reference number can be used for reinserting the associated extended object in other part(s) of the message. In order to perform this, an additional IE, known as the reused extended object IE, has been introduced in extended EMS. Figure 4.13 shows a message where a first picture is defined once and reused twice subsequently in other positions in the message.

The reused extended object IE has the structure shown in Table 4.24. The absolute character position of the original extended object is overwritten by the position indicated as part of the reused extended object IE.

Box 4.7 Recommendations for the reuse of extended objects

It is recommended to define the object to be reused in the first segment(s) of the message with the extended object IE and to reuse the defined extended object in the same or next message segments (according to the message concatenation index).

[3] The extended object reference number is unique in the message only.

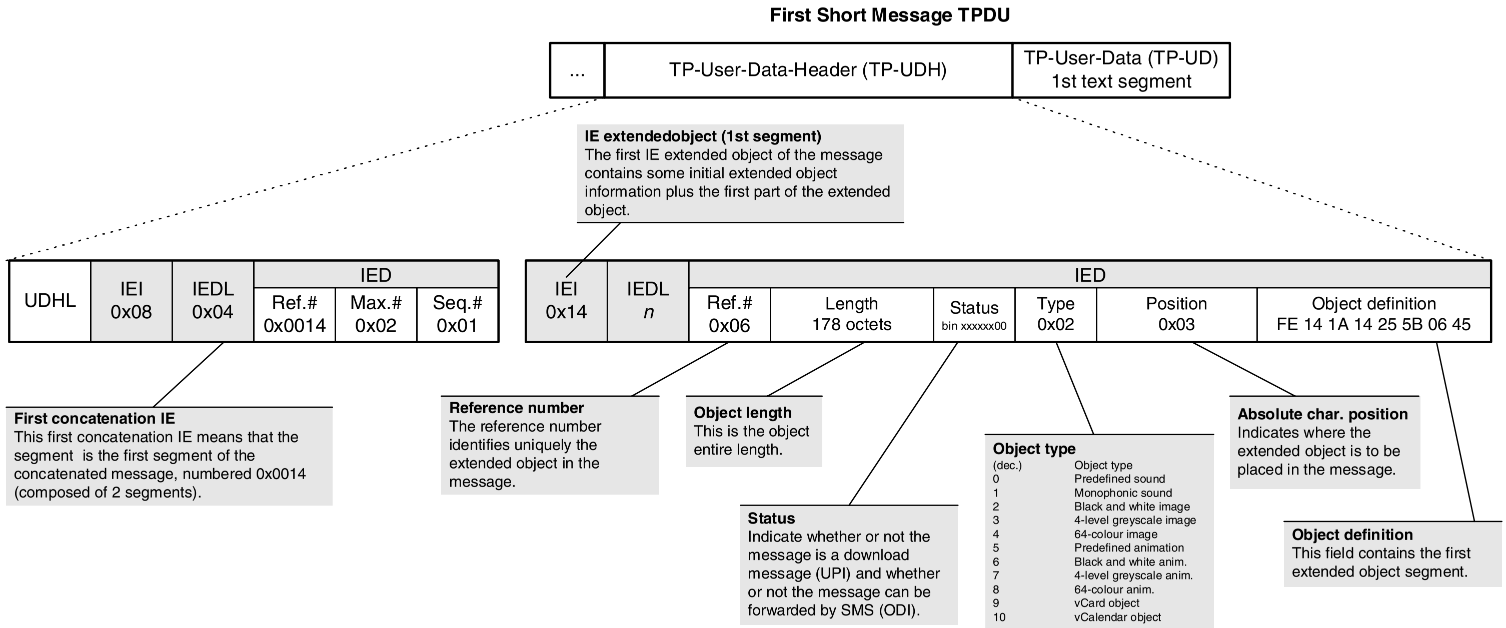


Figure 4.12 (*Continued*)

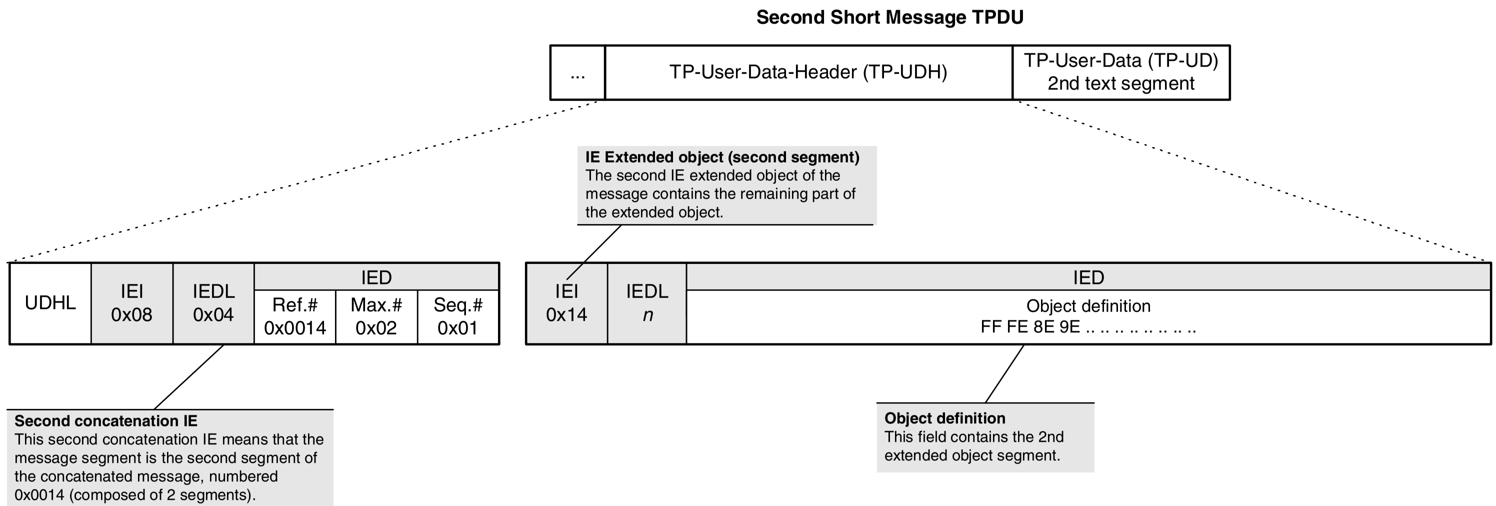


Figure 4.12 Extended object encoding

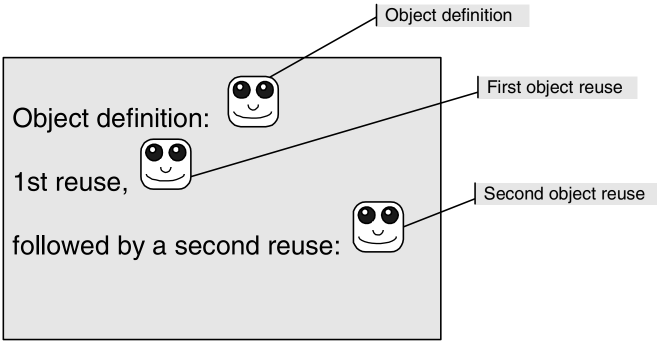


Figure 4.13 Example/object reuse

Table 4.24 IE/reused extended object

IEI	0x15 Reused extended object		From Release 5
IEDL	0x03 (3 octets)		
IED	Octet 1	**Reference number** This octet represents the reference number of the already-defined object.	
	Octets 2 and 3	**Absolute character position** These octets represent the absolute character position in the message where the reused extended object shall be placed.	

An extended object defined in a message can be reused in any part of the message in which it has been defined. However, an extended object cannot be reused, with the reused extended object IE, in other messages.

4.4.3 Compression of Extended Objects

Extended objects can be significantly larger than objects supported in basic EMS. In order to avoid an explosion of the number of message segments exchanged between SMEs, a support of compression/decompression has been introduced in extended EMS. Note that compression applies to extended objects only. Basic EMS elements cannot be decompressed/compressed unless they are encoded as extended objects. Several extended objects may be compressed together into a single compressed stream. Compressing several objects together usually achieves a better compression ratio.

4.4.3.1 Compressed Stream Structure

The compression mechanism introduced in extended EMS has similarities with the way uncompressed extended objects are encoded using the extended object IE. Compression is managed by a dedicated information element: the *compression control IE*. If a compressed stream is too large to fit into one message segment, then the compressed stream may be segmented in several parts and each part is conveyed into a single compression control IE. Each compression IE is inserted in a message segment along with other information elements and optional text. If several compression control IEs are used to convey a large compressed stream, then only the first compression IE contains the compression control header. Additional compression control IEs only contain raw compression data. The compression control header contains the following information:

- *Compression method*: this field identifies the compression method, which has been used to compress extended objects, along with optional compression settings. At the time of writing, the only compression/decompression method supported in extended EMS is based on LZSS compression principles.
- *Compressed bytestream length*: this is the length of the entire compressed bytestream (including the part in the first compression control IE plus part(s) in additional compression control IEs). This compressed bytestream length is required for the receiving SME to be able to correlate compressed streams with corresponding compression control IEs (more than one compressed bytestream may be present in a message). This information is necessary because additional compression control IEs do not refer to a particular compressed stream reference/index number.

Note that the compression/decompression method introduced in extended EMS is not used to compress the text part of the message: an SME, which does not support the extended EMS compression/decompression method, is still able to interpret the text part of a message containing one or more compressed bytestreams.

The first compression control IE is structured as shown in Table 4.25. If the compressed stream is too large to fit into one message segment, then the first part of the compressed stream is conveyed in the first message segment and remaining parts are conveyed in subsequent message segments with the information element shown in Table 4.26.

Octets 1...3 of the compression control IE represent the compression control header and are not taken into consideration for the calculation of the compressed bytestream length (see description of octets 2 and 3).

Box 4.8 Recommendation for a maximum size for messages containing compressed extended objects

A receiving SME is always able to interpret messages of an uncompressed size of up to eight message segments. A message with one or more compressed streams for which the uncompressed size is over eight message segments may not be interpreted correctly by all receiving SMEs. It is, therefore, highly recommended to limit the number of segments per message to eight (before compression). Note that decompression is mandatory for devices supporting extended EMS objects. On the other hand, compression is optional.

Figure 4.14 shows the encoding of a compressed stream containing three extended objects conveyed in a concatenated message composed of two message segments.

Table 4.25 IE/compression control (first)

IEI		0x16 Compression control (first)	From Release 5
IEDL		Variable	
Compression control header	Octet 1	**Compression method** This octet informs on the method, which has been used for compressing the bytestream. **Bits 3...0 compression method** 0000 LZSS compression 0001...1111 reserved **Bits 7...4 compression parameters** Value 0000 (binary) is assigned to this parameter for the LZSS compression scheme. Other values are reserved.	
	Octets 2 and 3	**Compressed bytestream length** These octets represent the length of the compressed stream (excluding compression control header). The compressed stream may expand over several message segments.	
IED	Octets 4...*n*	**Compressed bytestream definition** These octets represent the entire compressed bytestream definition or the first part of the compressed bytestream definition. If more than one message segments are required to represent the compressed bytestream, then the first compressed bytestream part is inserted in this information element and remaining part(s) of the compressed bytestream is/are inserted in subsequent message segments (inserted in other compression control information elements 0x16). The uncompressed bytestream consists of a sequence of extended objects or reused extended objects (IEI and IED only excluding IEDL). Note that only extended objects and reused extended objects can be compressed using this method. Basic EMS objects cannot be compressed with this method.	

Table 4.26 IE/compression control (additional)

IEI	0x16 Compression control object (additional)		From Release 5
IEDL	Variable		
IED	Octets 1 ... *n*	These octets represent an additional part of a compressed stream. Unlike the first compression control IE, this IE does not contain a compression control header.	

4.4.3.2 Compression and Decompression Methods

With extended EMS, the only compression method supported so far is derived from the LZSS compression principle.[4] LZSS-derived algorithms are often called dictionary-based compression methods. These methods compress a stream by adding, in the corresponding compressed stream, references to previously defined octet patterns rather than repeating

[4] The Lempel–Ziv–Storer–Szymanski (LZSS) compression principle is a modified version of the LZ77 compression principle proposed by Storer and Szymanski.

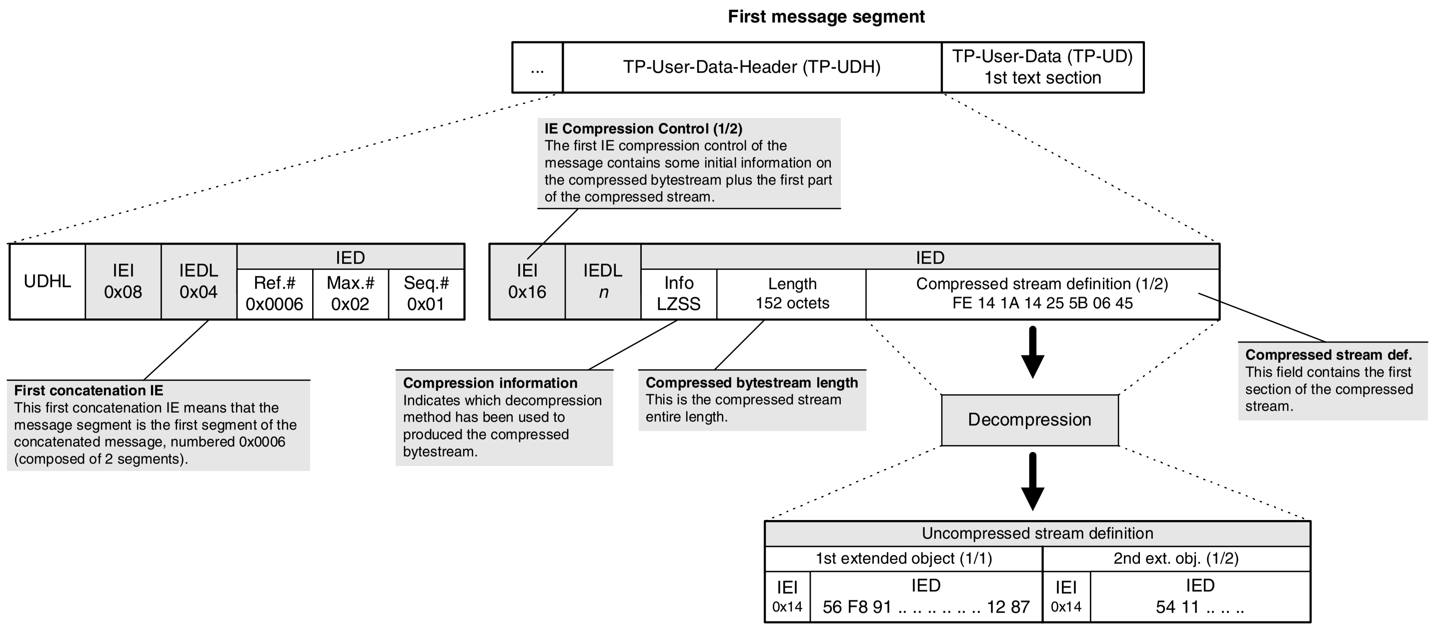


Figure 4.14 (*Continued*)

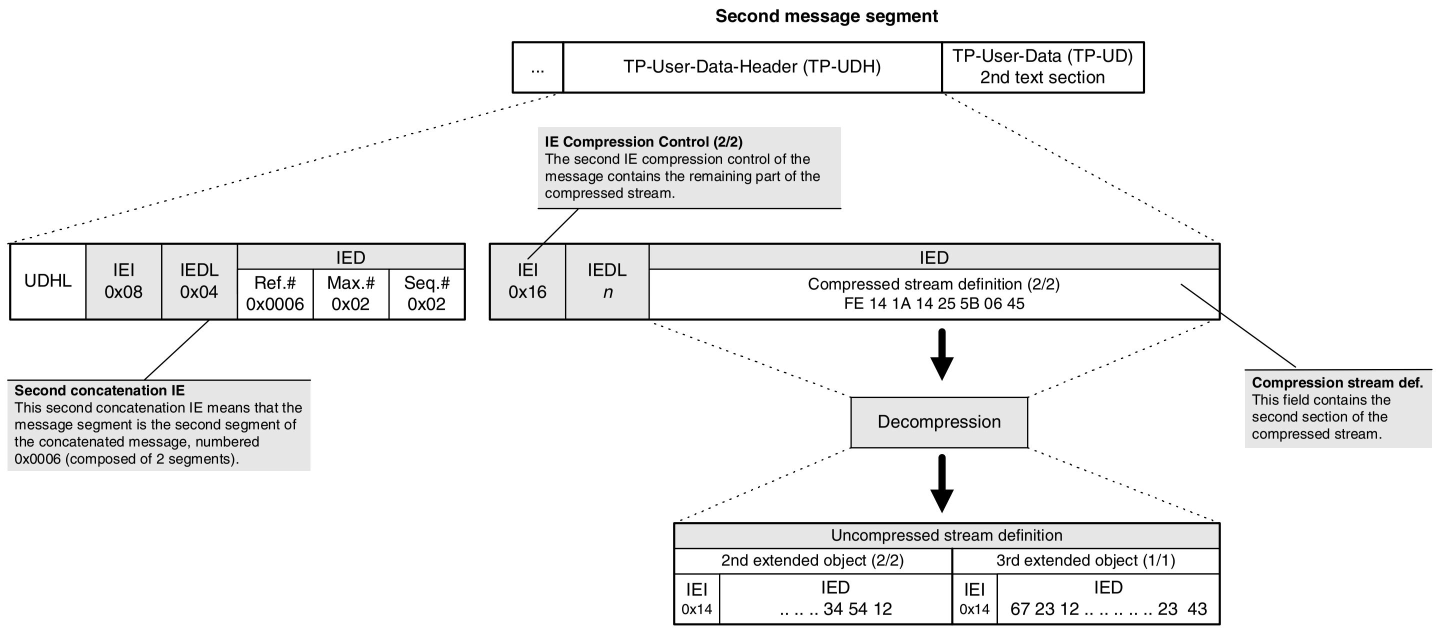


Figure 4.14 Compression control encoding

them. The compression ratio for such dictionary-based methods is, therefore, proportional to the frequency of occurrences of octet patterns in the uncompressed stream.

After compression, a compressed bytestream is composed of two types of elements: data blocks and block references. A *data block* contains an uncompressed block of octets. On the other hand, a *block reference* is used to identify a sequence of octets in the uncompressed stream in order to repeat the identified sequence simply by referring to it. Figure 4.15 shows an example of a compressed bytestream structure.

The data block is composed of a length and a payload. The length indicates the payload length in octets. The payload contains a sequence of up to 127 octets. The structure of a data block is shown in Figure 4.16. The block reference is composed of a repeated block length followed by a block location offset. The block reference is structured as shown in Figure 4.17.

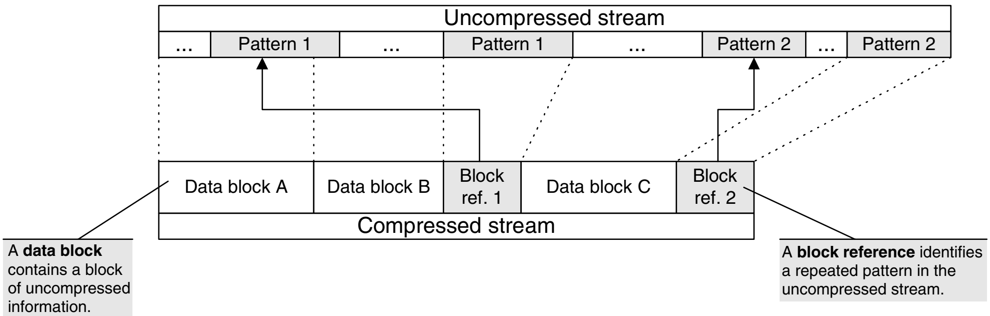


Figure 4.15 Structure of a compressed stream

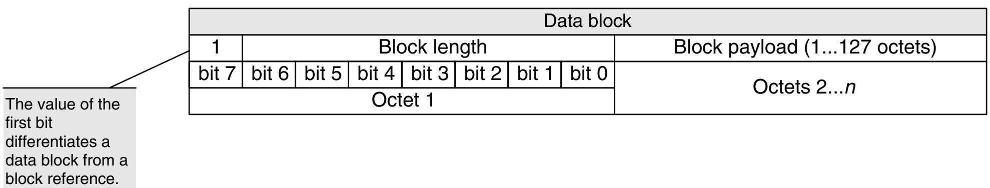


Figure 4.16 Structure of a data block

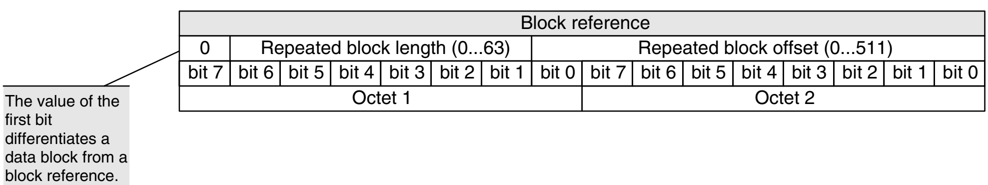


Figure 4.17 Structure of a block reference

Figure 4.18 Decompression process

4.4.3.3 Decompression Method

The decompression method consists of generating an output uncompressed stream from an input compressed stream as shown in Figure 4.18. The input compressed stream is composed of a compression control header followed by a sequence of data and reference blocks.

At the beginning of the decompression process, the output uncompressed stream is empty. A pointer starts from the beginning of the input compressed stream. All data blocks and block references are interpreted in their sequential order of occurrence in order to build the output uncompressed stream. Once a data block or block reference has been interpreted, the pointer moves to the next data block or block reference until the end of the compressed stream is reached. If the pointer is located on a data block, the block payload is extracted and appended at the end of the output uncompressed stream. If the pointer is located on a block reference, then a block of octets is identified in the output uncompressed stream and is appended at the end of the output uncompressed stream. The block to be repeated is the one, which has the size specified in the reference block (repeated block length) and which is located at a specified offset from the end of the output uncompressed stream, as specified in the reference block (repeated block offset).

4.4.3.4 Compression Method

The compression method consists of identifying repeated patterns of octets in the input uncompressed stream and inserting associated reference blocks in the output compressed stream as shown in Figure 4.19.

Octets that cannot be compressed in the input compressed stream are inserted as data blocks in the output compressed stream. The input uncompressed stream is scanned with a pointer from the start to the end, octet by octet. The current reading position in the input uncompressed stream is the octet designated by the pointer as shown in Figure 4.20.

For each octet pointed by the pointer in the uncompressed stream, the following process is performed:

In the sliding back window, the compression process looks for the longest octet pattern matching the sequence of octets starting at the beginning of the sliding front window. Only

Figure 4.19 Compression process

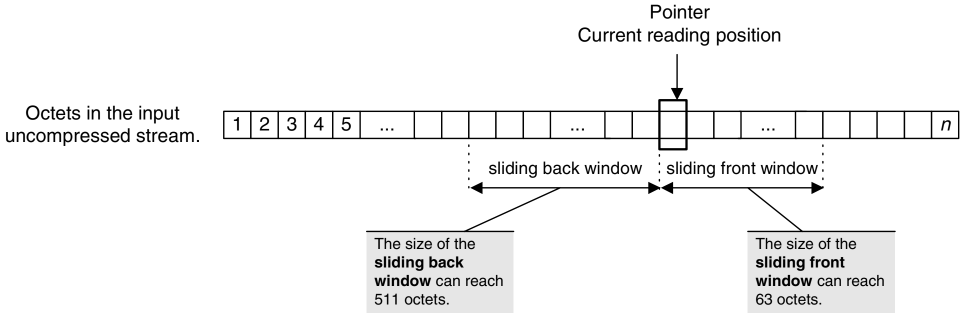


Figure 4.20 Elements of the compression process

patterns with a size over two octets and less than or equal to 63 octets are considered. At this stage two alternatives are possible:

- If *no matching pattern* is found, then the octet in the current reading position in the input uncompressed stream is appended at the end of the output compressed stream. For this purpose, if a data block is available at the end of the output compression stream, then the octet is appended to the data block payload (only if the payload has not reached the maximum length of 127 octets). Otherwise a new data block is inserted at the end of the output compressed stream (with the octet as only payload). The pointer of the input uncompressed stream moves to the next octet and the process is iterated at the new reading position.
- If *a matching pattern is found*, then a reference block is constructed according to the matching pattern characteristics. The reference block length is the length of the matching pattern and the reference block offset is equal to the number of octets between the current reading position and the beginning of the matching pattern in the input uncompressed stream. After construction, the reference block is appended at the end of the output compressed stream. The pointer of the input uncompressed stream moves to the octet that immediately follows the matching pattern in the input uncompressed stream and the process is iterated at the new reading position.

The compression process terminates when the pointer of the input uncompressed stream reaches the end of the stream.

The 3GPP technical specification [3GPP-23.040] defining the compression and decompression methods is provided with a set of test vectors. These test vectors enable implementers to test implementations.

It is difficult to estimate the compression ratio of the compression scheme. This ratio really depends on the frequency of occurrences of repeating patterns in the uncompressed stream. However, Table 4.27 shows the compression ratios for five of the test vectors provided with the [3GPP-23.040] technical specification.

4.4.4 Extended Objects

With the extended EMS framework, a new set of objects can be supported by extended EMS-enabled devices. The object type, in Table 4.28, is assigned to the "extended object type"

Table 4.27 Compression/test vectors

Test vector name	Uncompressed size (octets)	Compressed size (octets)	Compression ratio (%)
Black-and-white picture 64 × 62 pixels	498	448	10.04
Black-and-white picture 64 × 62 pixels	383	383	0
Grayscale picture 54 × 54 pixels	747	610	18.34
Color picture 54 × 54 pixels	2205	1566	28.98
Black-and-white animation 64 × 64 pixels (4 frames)	5652	2223	60.67
Black-and-white animation 16 × 16 pixels (4 frames)	148	94	36.49

Table 4.28 List of extended objects

Type	Description	
0x00	Predefined sound	
0x01	iMelody sound	
0x02	Black-and-white picture	
0x03	4-level grayscale picture	(new)
0x04	64-color picture	(new)
0x05	Predefined animation	
0x06	Black-and-white animation	
0x07	4-level grayscale animation	(new)
0x08	64-color animation	(new)
0x09	vCard object	(new)
0x0A	vCalendar object	(new)
0x0B	Vector graphics (WVG)	(new)
0x0C	Polyphonic melody (MIDI)	(new)
0x0B ... 0xFE	*Reserved for future use*	
0xFF	Capability information	(new)

parameter (octet 5) of the associated extended object information element (extended object header). In this table, an annotation indicates whether the format is new in extended EMS, or if it is already available in basic EMS.

4.4.5 Predefined Sounds

Predefined sounds are already available in basic EMS. Another way of including a predefined sound in a message consists in inserting it as an extended object. The information element representing a predefined sound as an extended object is shown in Table 4.29. User Prompt Indicator (UPI) and ODI flags do not apply to a predefined sound. Values assigned to these flags are ignored by receiving SMEs.

Table 4.29 IE/predefined sound

IEI		0x14 (extended object) Predefined sound	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1*	*Reference number*
		Octets 2 and 3	*Object length*
		Octet 4	*Object handling status (UPI + ODI)*
		Octet 5	*Extended object type* **0x00 (hex) Predefined sound**
		Octets 6 and 7	*Extended object position*
	Extended object definition	Octet 8	**Predefined sound number** This octet represents the predefined sound number as defined below: **Id** **Description** 0 Chimes high 1 Chimes low 2 Ding 3 TaDa 4 Notify 5 Drum 6 Claps 7 Fanfare 8 Chord high 9 Chord low *Other* *Reserved*

Box 4.9 Recommendation for the use of predefined sounds

There are two ways of inserting a predefined sound in a message: as a basic EMS predefined sound or as an extended EMS predefined sound. With basic EMS, the associated information element has a length of 4 octets (including IEI, IEDL, and IED). With extended EMS, the predefined sound information element has a length of 10 octets (without compression). There is a clear resource gain in using the basic EMS predefined sound rather than the extended predefined EMS sound. Furthermore, compared with the extended EMS predefined sound, a predefined sound in the basic EMS format is likely to be interpreted by a larger number of devices. Predefined sounds in the basic EMS format should therefore always be preferred.

4.4.6 iMelody

iMelody sounds are already available in basic EMS. An alternative method for including such sounds in a message consists of inserting them as extended objects. The information element representing an iMelody sound as an extended object is shown in Table 4.30.

Table 4.30 IE/iMelody melody

IEI		0x14 (extended object) iMelody melody	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1*	*Reference number*
		Octets 2 and 3	*Object length*
		Octet 4	*Object handling status (UPI + ODI)*
		Octet 5	*Extended object type* ***0x01 (hex)*** ***iMelody melody***
		Octets 6 and 7	*Extended object position*
	Extended object defination	Octet 8...*n*	**iMelody melody definition** These octets represent the melody in the iMelody format as described in Section 4.3.3.2.

Unlike basic EMS, an iMelody sound inserted in a message as an extended object may have a length greater than 128 octets. Because sounds in extended EMS may have long length, they are known as *melodies*. An iMelody melody, in the extended EMS format, may also benefit from compression.

4.4.7 Black-and-White Bitmap Picture

Compared with black-and-white bitmap pictures in basic EMS, dimensions of extended EMS bitmap pictures can reach a maximum of 255×255 pixels.

The information element representing a black-and-white bitmap picture as an extended object is shown in Table 4.31. Octets 1...7 of this information element contain the generic extended object header. Octets 8...*n* contain the picture-specific information (dimensions and bitmap). If the picture cannot be encoded in one message segment, then additional extended object IEs are needed as shown in Section 4.4.1. A picture, in the extended EMS format, may be compressed.

Table 4.32 presents the bitmap size and the number of message segments required to convey black-and-white pictures (without compression).

4.4.8 Grayscale Bitmap Picture

As for black-and-white pictures, the dimensions of 4-level grayscale bitmap pictures can reach a maximum of 255×255 pixels. In the uncompressed form, the state of each pixel of the bitmap picture is represented with 2 bits. It is, therefore, highly recommended to compress grayscale pictures. The corresponding extended object IE is structured as shown in Table 4.33.

Octets 1...7 of this information element contain the generic extended object header. Octets 8...*n* contain the picture-specific information (dimensions and bitmap). If the picture cannot be encoded in one message segment, then additional extended object IEs are needed as shown in Section 4.4.1. Table 4.34 presents the bitmap size and the number of message segments required to convey grayscale pictures (without compression).

Table 4.31 IE/black-and-white bitmap picture

IEI		0x14 (extended object) Black-and-white bitmap picture	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1*	*Reference number*
		Octets 2 and 3	*Object length*
		Octet 4	*Object handling status (UPI + ODI)*
		Octet 5	*Extended object type* ***0x02 (hex) Black-and-white impage***
		Octets 6 and 7	*Extended object position*
	Extended object definition	Octet 8	**Horizontal dimension (width)** Octet representing the picture width. Range: 1...255, value 0 is reserved (decimal).
		Octet 9	**Vertical dimension (height)** Octet representing the picture height. Range: 1...255, value 0 is reserved (decimal).
		Octets 10...*n*	**Black-and-white picture bitmap** These octets represent the bitmap of the picture from top left to bottom right. Unlike basic EMS, this bitmap does not necessarily have a horizontal length which is a multiple of 8. There is no fill bits between the encoding of each row of pixels. Fill bits are inserted, if needed, in the last octet. The most significant bit of each pixel represents the leftmost pixel. Each pixel state is encoded with 1 bit: 0 Pixel is white 1 Pixel is black

Table 4.32 Black-and-white picture sizes

Dimensions (pixel × pixel)	Bitmap size (octets)	Number of message segments
16 × 16	32	1
32 × 32	128	2
64 × 64	512	4

4.4.9 Color Bitmap Picture

As for black-and-white and grayscale pictures, dimensions of color bitmap pictures can reach a maximum of 255 × 255 pixels. In the uncompressed form, the state of each pixel of the bitmap picture is represented with 6 bits. It is, therefore, highly recommended to compress color bitmap pictures. The corresponding extended object IE is structured as shown in Table 4.35.

Octets 1...7 of this information element contain the generic extended object header. Octets 8...*n* contain the picture-specific information (dimensions and bitmap). If the picture cannot be encoded in one message segment, then additional extended object IEs are needed as shown in Section 4.4.1.

Table 4.33 IE/4-level grayscale bitmap picture

IEI		0x14 (extended object) 4-level grayscale bitmap picture	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1* *Octets 2 and 3* *Octet 4* *Octet 5* *Octets 6 and 7*	*Reference number* *Object length* *Object handling status (UPI + ODI)* *Extended object type* ***0x03 (hex) 4-level grayscale picture*** *Extended object position*
	Extended object definition	Octet 8	**Horizontal dimension (width)** Octet representing the image width. Range: 1...255, value 0 is reserved (decimal).
		Octet 9	**Vertical dimension (height)** Octet representing the image height. Range: 1...255, value 0 is reserved (decimal).
		Octets 10...*n*	**Grayscale picture bitmap** These octets represent the bitmap of the picture from top left to bottom right. Unlike basic EMS, this bitmap does not necessarily have a horizontal length which is a multiple of 8. There is no fill bits between the encoding of each row of pixels. Fill bits are inserted, if needed, in the last octet. Each pixel state is encoded with 2 bits: 00 Pixel is black 01 Pixel is dark gray 10 Pixel is light gray 11 Pixel is white.

Table 4.34 Grayscale picture sizes

Dimensions (pixel × pixel)	Bitmap size (octets)	Number of message segments
16 × 16	64	1
32 × 32	256	3
64 × 64	1024	8

Each pixel state of the color picture is represented by three pairs of bits. These three pairs of bits represent the quantities of red, blue, and green for each pixel as shown in Table 4.36.

Table 4.37 presents the bitmap size and the number of message segments required to convey color pictures (without compression).

4.4.10 Predefined Animation

Predefined animation is already available in basic EMS. Another way of including a predefined animation in a message consists of inserting it as an extended object. The

Table 4.35 IE/64-color bitmap picture

IEI		0x14 (extended object) 64-color bitmap picture	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1* *Octets 2 and 3* *Octet 4* *Octet 5* *Octets 6 and 7*	*Reference number* *Object length* *Object handling status (UPI + ODI)* *Extended object type* ***0x04 (hex)*** ***64-color picture*** *Extended object position*
	Extended object definition	Octet 8	**Horizontal dimension (width)** Octet representing the image width. Range: 1...255, value 0 is reserved (decimal).
		Octet 9	**Vertical dimension (height)** Octet representing the image height. Range: 1...255, value 0 is reserved (decimal).
		Octets 10...*n*	**Color picture bitmap** These octets represent the bitmap of the picture from top left to bottom right. Each pixel state is represented with 6 bits (64 colors) as shown in Table 4.36. As for other large object pictures, fill bits are only used on the last octet of the bitmap, if required.

Table 4.36 Color picture/color code, RGB(2,2,2)

Red		Green		Blue	
MSB Bit 5	LSB Bit 4	MSB Bit 3	LSB Bit 2	MSB Bit 1	LSB Bit 0

Table 4.37 Color picture sizes

Dimensions (pixel × pixel)	Bitmap size (octets)	Number of message segments
16 × 16	192	2
32 × 32	768	6
64 × 64	3072	24

information element representing a predefined animation as an extended object is shown in Table 4.38.

UPI and ODI flags do not apply to predefined animations. Values assigned to these flags are ignored by receiving SMEs. The analysis, provided in Box 4.9, regarding the choice of formats (basic EMS or extended EMS) for inserting predefined sounds in a message, also applies to predefined animations. Consequently, predefined animations should rather be inserted in a message as basic EMS objects rather than as extended EMS objects.

Table 4.38 IE/predefined animation

IEI		0x14 (extended object) Predefined animation	From Release 5
IEDL		Variable	
	Extended object header	*Octet 1* *Octets 2 and 3* *Octet 4* *Octet 5* *Octets 6 and 7*	*Reference number* *Object length* *Object handling status (UPI + ODI)* *Extended object type* ***0x05 (hex) Predefined animation*** *Extended object position*
IED	**Extended object definition**	Octet 8	**Predefined animation number** This octet represents the predefined animation number as defined below: **Id** — **Description** 0 — I am ironic, flirty 1 — I am glad 2 — I am skeptic 3 — I am sad 4 — Wow !!! 5 — I am crying 6 — I am winking 7 — I am laughing 8 — I am indifferent 9 — In love/kissing 10 — I am confused 11 — Tongue hanging out 12 — I am angry 13 — Wearing glasses 14 — Devil *Other* — *Reserved*

4.4.11 Black-and-White Animation

Compared with basic EMS black-and-white animations (maximum dimensions: four frames of 16 × 16 pixels), black-and-white animations in extended EMS can take various forms (up to 255 frames with dimensions of up to 255 × 255 pixels). The frame display time can be configured (same frame display time for the entire animation). The frame display time ranges from 100 ms to 1.6 s. The number of animation repetitions can also be specified. The number of repetitions can be unlimited or specified in the range 1–15. The corresponding extended object IE is structured as shown in Table 4.39.

Octets 1...7 of this information element contain the generic extended object header. Octets 8...*n* contain the animation specific information. If the animation cannot be conveyed in one message segment, then additional extended object IEs are needed as shown in Section 4.4.1.

4.4.12 Grayscale Animation

Grayscale animations are not available in basic EMS. In extended EMS, the frame display time can be specified (same frame display time for the entire animation). The frame display

Table 4.39 IE/black-and-white animation

IEI		0x14 (extended object) Black-and-white animation		From Release 5
IEDL		Variable		
IED	**Extended object header**	*Octet 1*	*Reference number*	
		Octets 2 and 3	*Object length*	
		Octet 4	*Object handling status (UPI + ODI)*	
		Octet 5	*Extended object type* ***0x06 (hex)*** ***Black-and-white animation***	
		Octets 6 and 7	*Extended object position*	
	Extended object definition	Octet 8	**Horizontal dimension (width)** Octet representing the animation width. Range: 1...255, value 0 is reserved (decimal).	
		Octet 9	**Vertical dimension (height)** Octet representing the animation height. Range: 1...255, value 0 is reserved (decimal).	
		Octet 10	**Number of frames in the animation** Range: 1...255, value 0 is reserved.	
		Octet 11	**Animation control** This octet indicates the number of times the animation is to be repeated (from 1 to 15 repetitions or unlimited repetition). It also indicates the display time for each frame. The octet is structured as follows: **Bits 7...4** **Frame display time** 0000 (0.1 s) 0110 (0.7 s) 1100 (1.3 s) 0001 (0.2 s) 0111 (0.8 s) 1101 (1.4 s) 0010 (0.3 s) 1000 (0.9 s) 1110 (1.5 s) 0011 (0.4 s) 1001 (1.0 s) 1111 (1.6 s) 0100 (0.5 s) 1010 (1.1 s) 0101 (0.6 s) 1011 (1.2 s) **Bits 3...0** **Repeat count** 0000 Unlimited repetition 0001 1 repetition ... n repetitions ($1 < n < 15$) 1111 15 repetitions	
		Octets 12...*n*	**Animation frame bitmap(s)** These octets represent *n* bitmaps in the format defined in Section 4.4.7. At the end of each frame representation fill bits may be inserted, if required, to allow the following frame to start on an octet boundary.	

time ranges from 100 ms to 1.6 s. The number of animation repetitions can also be specified. The number of repetitions can be unlimited or specified in the range 1–15. The corresponding extended object IE is structured as shown in Table 4.40.

Octets 1...7 of this information element contain the generic extended object header. Octets 8...*n* contain the animation specific information. If the animation cannot be encoded

Table 4.40 IE/grayscale animation

IEI		0x14 (extended object) 4-level grayscale animation	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1*	*Reference number*
		Octets 2 and 3	*Object length*
		Octet 4	*Object handling status (UPI + ODI)*
		Octet 5	*Extended object type* ***0x07 (hex)*** ***4-levels grayscale animation***
		Octets 6 and 7	*Extended object position*
	Extended object definition	Octet 8	**Horizontal dimension (width)** Octet representing the animation width. Range: 1...255, value 0 is reserved (decimal).
		Octet 9	**Vertical dimension (height)** Octet representing the animation height. Range: 1...255, value 0 is reserved (decimal).
		Octet 10	**Number of frames in the animation** Range: 1...255, value 0 is reserved (decimal).
		Octet 11	**Animation control** This octet indicates the number of times the animation is to be repeated (from 1 to 15 repetitions or unlimited repetition). It also indicates the display time for each frame. The octet is structured as follows: **Bits 7...4** **Frame display time** 0000 (0.1 s) 0110 (0.7 s) 1100 (1.3 s) 0001 (0.2 s) 0111 (0.8 s) 1101 (1.4 s) 0010 (0.3 s) 1000 (0.9 s) 1110 (1.5 s) 0011 (0.4 s) 1001 (1.0 s) 1111 (1.6 s) 0100 (0.5 s) 1010 (1.1 s) 0101 (0.6 s) 1011 (1.2 s) **Bits 3...0** **Repeat count** 0000 Unlimited repetition 0001 1 repetition ... n repetitions ($1 < n < 15$) 1111 15 repetitions
		Octets 12...n	**Animation frame bitmap(s)** These octets represent n bitmaps in the format defined in Section 4.4.8. At the end of each frame representation fill bits may be inserted, if needed, to allow the following frame starts on an octet boundary.

in one message segment, then additional extended object IEs are needed as shown in Section 4.4.1.

4.4.13 Color Animation

Color animations are not available in basic EMS. In extended EMS, the frame display time can be specified (same frame display time for the entire animation). The frame display time ranges from 100 ms to 1.6 s. The number of animation repetitions can also be specified. The number of repetitions can be unlimited or limited in the range 1–15. The corresponding extended object IE is structured as shown in Table 4.41.

Octets 1...7 of this information element contain the generic extended object header. Octets 8...*n* contain the animation specific information. If the animation cannot be encoded in one message segment, then additional extended object IEs are needed as shown in Section 4.4.1.

Table 4.41 IE/color animation

IEI		0x14 (extended object) 64-color animation	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1*	*Reference number*
		Octets 2 and 3	*Object length*
		Octet 4	*Object handling status (UPI + ODI)*
		Octet 5	*Extended object type* ***0x08 (hex)*** ***64-color animation***
		Octets 6 and 7	*Extended object position*
	Extended object definition	Octet 8	**Horizontal dimension (width)** Octet representing the animation width. Range: 1...255, value 0 is reserved (decimal).
		Octet 9	**Vertical dimension (height)** Octet representing the animation height. Range: 1...255, value 0 is reserved (decimal).
		Octet 10	**Number of frames in the animation** Range: 1...255, value 0 is reserved (decimal).
		Octet 11	**Animation control** This octet indicates the number of times the animation is to be repeated (from 1 to 15 repetitions or unlimited repetition). It also indicates the display time for each frame. The octet is structured as follows: **Bits 7...4** **Frame display time** 0000 (0.1 s) 0110 (0.7 s) 1100 (1.3 s) 0001 (0.2 s) 0111 (0.8 s) 1101 (1.4 s) 0010 (0.3 s) 1000 (0.9 s) 1110 (1.5 s) 0011 (0.4 s) 1001 (1.0 s) 1111 (1.6 s) 0100 (0.5 s) 1010 (1.1 s) 0101 (0.6 s) 1011 (1.2 s) **Bits 3...0** **Repeat count** 0000 Unlimited repetition 0001 1 repetition ... n repetitions $(1 < n < 15)$ 1111 15 repetitions
		Octets 12...*n*	**Animation frame bitmap(s)** These octets represent *n* bitmaps in the format defined in Section 4.4.9. At the end of each frame representation fill bits may be inserted, if required, to allow the following frame to start on an octet boundary.

Table 4.42 Animation sizes

	Frame dimensions (pixel × pixel)	Bitmap size (4 frames) (octets)	Number of message segments
Black-and-white	16 × 16	128	2
	32 × 32	512	4
	64 × 64	2048	16
Grayscale	16 × 16	256	3
	32 × 32	1024	8
	64 × 64	4096	32
Color	16 × 16	768	6
	32 × 32	3072	24
	64 × 64	12,288	94

Table 4.42 shows the bitmap size and the number of message segments required to insert various types of 4-frame animations in extended EMS messages (without compression).

Box 4.10 Recommendation for a maximum object size

Although a message could theoretically be composed of 255 message segments, one can seldom find devices that can interpret messages composed of more than eight message segments. To ensure that a message can be rendered properly on the largest number of devices, the number of message segments should not exceed eight (uncompressed size).

4.4.14 vCard Data Stream

The vCard format is used for representing electronic business cards [IMC-vCard]. This format is already widely used with Personal Digital Assistants and is becoming the de facto format for the exchange of electronic business cards over infrared links. It is also becoming common to attach a vCard data stream, as a signature, to an Email message. A vCard data stream contains basic contact details such as last name, first name, postal and electronic addresses, phone, mobile, and fax numbers, etc. It may also contain more complex data such as photographs, company logos, or audio clips. The information element enabling the inclusion of a vCard data stream in a message is structured as shown in Table 4.43.

A vCard data stream is a collection of one or more properties. A property is composed of a property name and an assigned value. In a vCard data stream, a set of properties can be grouped to preserve the coupling of the properties' meaning. For instance, the properties for a telephone number and a corresponding comment can be grouped together. Note that a vCard data stream is composed of a main vCard object, which can also contain nested vCard objects. Properties for a vCard object are included between the two delimiters `BEGIN:V-CARD` and `END:VCARD`. A property (name, parameters, and value) takes the following form:

```
<PropertyName> [;<PropertyParameters>] : <PropertyValue>
```

Table 4.43 IE/vCard

IEI		0x14 (extended object) vCard data stream (version 2.1)	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1* *Octets 2 and 3* *Octet 4* *Octet 5* *Octets 6 and 7*	*Reference number* *Object length* *Object handling status (UPI + ODI)* *Extended object type* ***0x09 (hex) vCard object*** *Extended object position*
	Extended object definition	Octets 9...*n*	**Definition of the vCard data stream** These octets represent the sequence of instructions composing the vCard data stream. Instructions are represented in a text format using the UTF8 encoding (8-bit aligned).

A property can expand over more than one line of text. Property names in a vCard data stream are case insensitive. The property name, along with an optional grouping label, must appear as the first characters on a line. In a data stream, individual text lines are separated by a line break (ASCII character 13 followed by ASCII character 10). Note that long lines of text can be represented with several shorted text lines using the "RFC 822 folding technique." Two types of groups are allowed in the vCard format: vCard object grouping and property grouping. With vCard object grouping, several distinct vCard objects are grouped together whereas in property grouping, a collection of properties within a single vCard object are grouped.

Several generic parameters can be used for the properties of a vCard data stream. The encoding of a property value can be specified with the `ENCODING` property parameter. Values that can be assigned to this parameter are `BASE64`, `QUOTED-PRINTABLE`, or `8BIT`. Similarly, the character set can be specified with the `CHARSET` property parameter. Values that can be assigned to this parameters are `UTF-8`, `ASCII`, and `ISO-8859-8`. The default language for a vCard data stream is US English (`en-US`). This default configuration for a property value can be overridden with the `LANGUAGE` property parameter. Values that can be assigned to this parameter are identified in [RFC-1766]; e.g., `en-US` or `fr-CA`. The value to be assigned to a property is usually specified inline (`INLINE`) as part as the property definition, as shown above. However, the value to be assigned to a property can also be located outside the vCard data stream. The location of the property value is specified with the `VALUE` parameter. Values that can be assigned to this parameter are `INLINE` (default) and `URL` (for an external value accessible via a uniform resource locator).

Figure 4.21 shows a vCard data stream containing one vCard object.

The properties given in Table 4.44 can be used for the definition of a vCard data stream. The only two properties that should always be present in a vCard data stream (version 2.1) [IMC-vCard] are the name (`N`) and the version (`VERSION`).

```
BEGIN:VCARD
VERSION:2.1
UID:6666-601
N:Le Bodic, Gwenael
TEL;HOME:+49-1-23-45-67-89
TEL;CELL:+49-6-23-45-67-89
ORG:Vodafone;Research and Development
AGENT:
BEGIN:VCARD
VERSION:2.1
UID:6666-602
N:Tayot, Marie
TEL;WORK;FAX:+49-1-23-45-67-90
END:VCARD
END:VCARD
```

Figure 4.21 vCard format/example

Table 4.44 vCard properties

Property name	Description	Example
`FN`	Formatted name (include honorific prefixes, suffixes, titles, etc.)	`FN:Dr. G. Le Bodic`
`N`	Name (structured name of the person, place, or thing). For a person, the property value is a concatenation of family name, given name, etc.	`N:Le Bodic, Gwenael`
`PHOTO`	Photograph. The photograph is represented as a picture. Parameters of this property is the picture format such as: • `GIF`: Graphics interchange format • `BMP`: Bitmap format • `JPEG`: Jpeg format etc.	
`BDAY`	Birthday. The value assigned to this property represents the birthday of a person.	`BDAY:1973-07-06`
`ADR`	Delivery address. Property parameters: • `DOM`: Domestic address • `INTL`: International address • `POSTAL`: Postal address • `PARCEL`: Parcel delivery address • `HOME`: Home delivery address • `WORK`: Work delivery address The structured property value is a concatenation of the post office address, extended address, street, locality, region, postal code, and country.	`ADR;HOME:P.O. Box 6;16 Wood street;London; UK;SW73NH`

`LABEL`	Delivery label. The property value represents a postal delivery address. Unlike the previous property, the value assigned to this property is not structured. The property is associated with the same parameters as the ones associated with the `ADR` property.	`LABEL;INTL;ENCODING=`
		`QUOTED-PRINTABLE:-`
		`P.O. Box 6=0D=0A`
		`16 Wood street=OD=0A`
		`London=0D=0A`
		`United Kingdom=0D=0A`
		`SW73NH`

(*Continued*)

Table 4.44 (*Continued*)

Property name	Description	Example
TEL	Telephone. Property parameters: • PREF: Preferred number • WORK: Work number • HOME: Home Number • VOICE: Voice number • FAX: Facsimile number • MSG: Messaging service number • CELL: Cellular number • PAGER: Pager Number • BBS: Bulletin board service number • MODEM: Modem number • CAR: Car-phone number • ISDN: ISDN number • VIDEO: Video phone number	TEL;PREF;CELL:+33-6-12-13-14-15
EMAIL	Electronic mail. Main property parameters are (list is not exhaustive): • INTERNET: Internet address • AOL: America online address • CIS: Computer information service • TELEX: Telex number • X400: X.400 number	EMAIL;INTER-NET:gwenael@lebodic.net
MAILER	Mailer. The value assigned to this property indicates, which mailer is used by the person associated with the vCard object.	MAILER:ccMail 2.2
TZ	Time zone. The value assigned to this property indicates the offset from the Co-ordinated Universal Time (UTC). In the corresponding examples, the time zone is, respectively, −8 hours (Pacific standard time) and −5 hours (Eastern standard time).	TZ:−08:00 or TZ:-0500
GEO	Geographic position. The value assigned to this property indicates a global position in terms of longitude and latitude (in this order).	GEO:37.24,−17.85
TITLE	Title. The value assigned to this property indicates the job title, functional position, or function of the person associated with the vCard object.	TITLE:Architect, Research and Development
ROLE	Business category. The value assigned to this property indicates the role, occupation, or business category of the person associated with the vCard object.	ROLE:Executive
LOGO	Logo (e.g., company logo). The logo is represented as a picture. Parameters of this property is the picture format such as: • GIF: Graphics interchange format • BMP: Bitmap format • JPEG: Jpeg format etc.	LOGO;ENCODING=BASE64;-TYPE=GIF:R01GOdhfg

(*Continued*)

Table 4.44 (*Continued*)

Property name	Description	Example
AGENT	Agent. The value assigned to this property is another vCard object that provides information about the person acting on behalf of the person associated with the main vCard object of the vCard data stream.	AGENT: BEGIN:VCARD VERSION:2.1 N:Tayot;Marie TEL;CELL:+1-612-253-3434 END:VCARD
ORG	Organization name and unit. The structured property value is a concatenation of the company name, company unit, and organizational unit.	ORG:Vodafone;Marketing
NOTE	Comment.	NOTE;ENCODING=QUOTED-PRINTABLE:Do not use the mobile phone number after 9:00 p.m.
REV	Last revision of the vCard object.[1]	REV:2002-12-23T21:12:11Z
SOUND	Sound. The value assigned to this property specifies a sound annotation associated with the vCard object. One property parameter indicates the sound format: • WAVE: WAVE format • PCM: PCM format • AIFF: AIFF format	
URL	Uniform Resource Locator (URL). The value assigned to this property represents a URL which refers to additional information related to the vCard object.	URL:http://www.lebodic.net
UID	Unique identifier. The value assigned to this property indicates a globally unique identifier for the vCard object.	UID:6666666-123456-34343
VERSION	Version. The value assigned to this property indicates the version of the vCard format to which complies the vCard writer.	VERSION:2.1
KEY	Public key. Property parameters: X509: X.509 public key certificate. PGP: PGP key.	

[1]See definition of date value in Section 4.4.15.

4.4.15 vCalendar Data Stream

The vCalendar format is used to represent items generated by calendaring and scheduling applications [IMC-vCalendar]. As for the vCard format, the vCalendar format is widely used with Personal Digital Assistants and is becoming the de facto format for the exchange of calendaring and scheduling information. A vCalendar data stream is composed of one or more objects of types—event and todo. An *event* is a calendaring and scheduling object representing an item in a calendar. A *todo* is a calendaring and scheduling object

Table 4.45 IE/vCalendar

IEI		0x14 (extended object) vCalendar data stream (version 1.0)	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1*	*Reference number*
		Octets 2 and 3	*Object length*
		Octet 4	*Object handling status (UPI + ODI)*
		Octet 5	*Extended object type* ***0x0A (hex)*** ***vCalendar data stream***
		Octets 6 and 7	*Extended object position*
	Extended object definition	Octets 9...*n*	**Definition of the vCalendar data stream** These octets represent the sequence of instructions composing the vCalendar data stream. Instructions are represented in a text format using the UTF8 encoding (8-bit aligned).

representing an action item or assignment. As for a vCard data stream, the vCalendar data stream is composed of a collection of properties with a main vCalendar object including nested vCalendar objects (event and todo objects). Properties for a vCalendar object are included between the two delimiters `BEGIN:VCALENDAR` and `END:VCALENDAR`. Similarly, properties of event and todo objects are included between the pairs of delimiters, `BEGIN: VEVENT/END:VEVENT` and `BEGIN:VTODO/END:VTODO`, respectively. The information element enabling the inclusion of a vCalendar data stream in a message is structured as shown in Table 4.45.

The format of a vCalendar property is the same as that of a vCard property. Generic vCard parameters (`ENCODING`, `CHARSET`, `LANGUAGE`, and `VALUE`) can also be used for the configuration of properties in vCalendar data streams.

For a property representing a date, the value to be assigned to the property is structured as follows:

```
<year> <month> <day> T <hour> <minute> <second> <type designator>
```

For instance, 9:30:00 a.m. on April 20, 2002 local time is formatted as

```
20020420T093000
```

The same date/time in UTC-based time is formatted as

```
20020420T093000Z
```

Values assigned to vCalendar properties can also refer to time periods. Such period values are formatted as shown in Figure 4.22.

A value representing a time period is prefixed with the “P” character. After the P character, optional blocks represent the time period in terms of years, months, and days. Optionally, the value can also contain a time component starting with the character “T” and followed by optional blocks refining the time duration in terms of hours, minutes, and

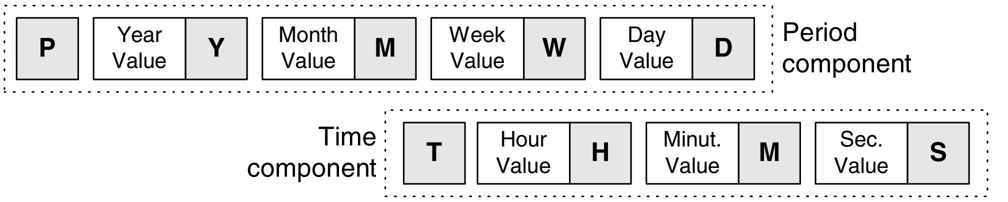


Figure 4.22 vCalendar/period format

seconds. For instance, a period of 1 year and 2 weeks is represented by the following character string:

```
P1Y2W
```

A period of 15 minutes is represented by the following character string:

```
PT15M
```

A vCalendar object can characterize recurring events. A recurrence rule is a character string with the format shown in Figure 4.23.

A recurrence rule starts with a character which identifies the event frequency (“D” for daily, “W” for weekly, “M” for monthly, and “Y” for yearly). This is followed by an interval indicating how often the event repeats (daily, every fourth day, etc.). Optionally, the interval can be followed by one or more frequency modifiers, followed by an optional end date or duration. The following list shows examples of daily and weekly recurrence rules:

`D1 #10`	Daily for ten occurrences
`D1 20020420T093000Z`	Daily until 20th April 2002
`W2 #0`	Every other week, forever.

Two types of monthly recurrence rules are available in the vCalendar format: the by-position rule and the by-day rule. The *by-position* rule allows weekdays in the month to be specified in relation to their occurrence in the month (e.g., 2nd Sunday of the month). The *by-day* rule allows specific days to be specified (e.g., 6th day or 3rd day). A by-position

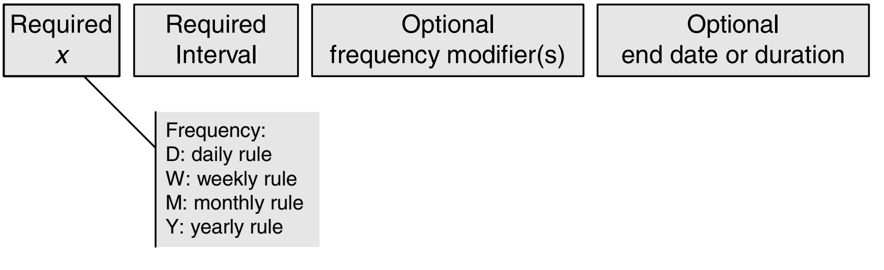


Figure 4.23 vCalendar/recurrence rule

rule starts with the prefix "MP" whereas a by-day rule starts with the prefix "MD." The following list shows examples of monthly recurrence rules:

`MP1 1+ TU #6`	Monthly on the first Tuesday for six occurrences
`MP2 2+ TU 1− SU #2`	Every other month on the second Tuesday and last Sunday for two occurrences
`MD1 1 1− #3`	Monthly on the first and last day for three occurrences.

Two types of yearly recurrence rules are available in the vCalendar format: the by-month rule and the by-day rule. The by-month rule allows specific months to be specified (e.g., yearly in July). The by-day rule allows specific days to be specified (e.g., 206th day of the year). A by-month rule starts with the prefix "YM" whereas a by-day rule starts with the prefix "YD." The following list shows examples of yearly recurrence rules:

`YM1 6 7 #2`	Yearly in June and July for two occurrences
`YD5 206 #3`	Every fifth year on the 206th day for three occurrences
`YD3 2 20050601T000000Z`	Every third year on the 2nd day until the 1st June 2005.

Table 4.46 shows the properties that can be used for the definition of a vCalendar data stream. The only property that should always be present in a vCalendar data stream (version 1.0) is the version (VERSION).

Table 4.47 lists the properties that can be used for the definition of event and todo objects within a vCalendar data stream.

Table 4.46 vCalendar properties

Property name	Description	Example
DAYLIGHT	Daylight saving. The value assigned to the property indicates the daylight saving of the application that generated the vCalendar data stream. The first example indicates that the application does not observe any daylight savings time (value is FALSE). The second example indicates that daylight savings time is observed and that the summer time is 6 hours behind UTC (summer time is from 7/4/2002 to 27/10/2002).	DAYLIGHT:FALSE or DAYLIGHT:TRUE;-06;20020407T025959;20021027T010000T010000;EST;EDT
GEO	Geographic position. The value assigned to this property indicates a global position in terms of longitude and latitude (in this order).	GEO:37.24,−17.85
PRODID	Product identifier.	PRODID:Manufacturer PIM
TZ	Time zone. The value assigned to this property indicates the offset from the Co-ordinated Universal Time (UTC). In the corresponding examples, the time zone is, respectively, −8 hours (Pacific standard time) and −5 hours (Eastern standard time).	TZ:−08:00 or TZ:−0500
VERSION	Version. The value assigned to this property indicates the version of the vCalendar format to which complies the vCalendar writer.	VERSION:1.0

Table 4.47 vCalendar properties/event and todo objects

Property name	Description	Example
ATTACH	Attachment. The attachment can be a document to be reviewed or a list of actions for a todo object.	ATTACH;VALUE=URL:file://lebodic.net/todo.ps
ATTENDEE	Attendee to a group event or a person associated with a todo object. Optional parameters for this property are • ROLE (possible values are ATTENDEE, ORGANIZER, OWNER, and DELEGATE / default is ATTENDEE) • STATUS (possible values are ACCEPTED, NEEDS ACTION, SENT, TENTATIVE, CONFIRMED, DECLINED, COMPLETED, and DELEGATED / default is NEEDS ACTION). RSVP (possible values are YES or NO / default is NO). EXPECT (possible values are FYI, REQUIRE, REQUEST, and IMMEDIATE / default is FYI).	ATTENDEE;ROLE=OW-NER;STATUS=COMPLETED:gwenael@lebodic.net
AALARM	Audio reminder. Sound to be played to notify the corresponding event. An additional property parameter is the sound format (see possible values for the SOUND property in the vCard format). This property value is composed of the run time/start time, snooze time, repeat count, and audio content.	AALARM;TYPE=WAVE;VA-LUE=URL:20020706-T114500;;;file:///notify.wav
CATEGORIES	Categories. More than one category value may be assigned to this property. Possible values are APPOINTMENT, BUSINESS, EDUCATION, HOLIDAY, MEETING, MISCELLANEOUS, PERSONAL, PHONE CALL, SICK DAY, SPECIAL OCCASION, TRAVEL, and VACATION.	CATEGORIES:APPOINTMENT;BUSINESS
CLASS	Classification. The value assigned to this property indicates the accessibility of the associated information. Possible values are PUBLIC, PRIVATE, and CONFIDENTIAL. Default value is PUBLIC.	CLASS:CONFIDENTIAL
DCREATED	Date/time of creation for a todo or event object.	DCREATED:20020421T192500
COMPLETED	Date/time of completion for a todo object.	COMPLETED:20020421T112000
DESCRIPTION	Description.	DESCRIPTION:Meeting part of the workshop.

(*Continued*)

Table 4.47 (*Continued*)

Property name	Description	Example
DALARM	Display reminder. This property value is composed of the run time/start time, snooze time, repeat count, and display string.	DALARM:20020421T080000;PT5M;3;Wake up!
DUE	Due date/time for a todo object.	DUE:20020421T192500
DTEND	End date/time for an event object.	DTEND:20020421T192500
EXDATE	Exception date/times. This property defines a list of exceptions for a recurring event.	EXDATE:20020402T010000Z;20020403T010000Z
XRULE	Exception rule. This property defines a list of exceptions for a recurring event by specifying an exception rule. In the corresponding example: except yearly in June and July forever.	XRULE:YM1 6 7 #0
LAST-MODIFIED	Last modified.	LAST-MODIFIED:20020421T192500Z
LOCATION	Location.	LOCATION:Conference room
MALARM	Mail reminder. This property contains an Email address to which an event reminder is to be sent. This property value is composed of the run time/start time, snooze time, repeat count, Email address, and note.	MALARM:20020421T000000;PT1H;2;gwenael@le-bodic.net;Do not forget the meeting.
RNUM	Number recurrences. This property defines the number of times a vCalendar object will reoccur.	RNUM:6
PRIORITY	Priority. Value 0 is for an undefined priority, value 1 is the highest priority, value 2 is for the second highest priority, and so on.	PRIORITY:2
PALARM	Procedure reminder. A procedure reminder is a procedure, or application that will be executed as an alarm for the corresponding event. This property value is composed of the run time/start time, snooze time, repeat count, and procedure name.	PALARM;VALUE=URL:20020421-T100000;PT5M;1;file:///le-bodic.net/notify.exe
RELATED-TO	Related-to. This property allows the definition of relationships between vCalendar entities. This is performed by using globally unique identifiers as specified by the UID parameter.	RELATED-TO:6666-87-90
RDATE	Recurrence date/times. This property defines a list of date/times for a recurring vCalendar object.	RDATE:20020421T010000Z;20020422T010000Z
RRULE	Recurrence rule. This property defines a recurring rule for a vCalendar object. The corresponding example indicates that the corresponding event occurs weekly on Tuesday.	RRULE:W1 TU
RESOURCES	Resources. Possible values for this property are CATERING, CHAIRS, COMPUTER, PROJECTOR, VCR, VEHICLE, etc.	RESOURCES:VCR;CHAIRS

(*Continued*)

Table 4.47 (*Continued*)

Property name	Description	Example
SEQUENCE	Sequence number. This property defines the instance of the vCalendar object in a sequence of revisions.	SEQUENCE:3
DTSTART	Start date/time for an event object. This property defines the date and time when the event will start.	DTSTART:20020421T235959
STATUS	Status. This property defines the status of the related vCalendar object. Possible values for this property are ACCEPTED, NEEDS ACTION, SENT, TENTATIVE, CONFIRMED, DECLINED, COM-PLETED, and DELEGATED. Default value is NEEDS ACTION.	STATUS:ACCEPTED
SUMMARY	Summary. This property defines a short summary (or subject) for the vCalendar object.	SUMMARY:Debriefing
TRANSP	Time transparency. This property defined whether the event is transparent to free time searches. Possible values are: 0: object will block time and will be factored into a free time search. 1: object will not block time and will not be factored into a free time search. other: implementation specific transparency semantics.	TRANSP:0
URL	Uniform Resource Locator (URL). The value assigned to this property represents a URL, which refers to additional information related to the vCalendar object.	
UID	Unique identifier. The value assigned to this property indicates a globally unique identifier for the vCalendar object.	UID:66666-32432-4532

According to the vCalendar format v1.0 [IMC-vCalendar], properties that should always be supported for an event object are category (CATEGORIES), description (DESCRIPTION), start date/time (DTSTART), end date/time (DTEND), priority (PRIORITY), status (STATUS), and summary (SUMMARY).

According to the vCalendar format v1.0 [IMC-vCalendar], properties that should always be supported for a todo object are category (CATEGORIES), date/time completed (COMPLETED), description (DESCRIPTION), due date/time (DUE), priority (PRIORITY), status (STATUS), and summary (SUMMARY).

Figure 4.24 gives an example of a vCalendar data stream. The data stream contains an event object only. Figure 4.25 gives an example of a vCalendar data stream. The data stream contains a todo object only.

```
BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VEVENT
UID:6666-542-2813
CATEGORIES:MEETING
STATUS:TENTATIVE
PRIORITY:1
DTSTART:20020422T030000Z
DTEND:20020422T050000Z
SUMMARY:Debriefing
DESCRIPTION:Analysis of solutions and decision
CLASS:PRIVATE
END:VEVENT
END:VCALENDAR
```

Figure 4.24 vCalendar format (event)/example

```
BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VTODO
UID:6666-542-2814
CATEGORIES:BUSINESS
STATUS:COMPLETED
COMPLETED:20020422T030000Z
DUE:20020423T050000Z
PRIORITY:1
SUMMARY:Analyse solution no2
DESCRIPTION:Conduct purchasing and software analyses
CLASS:PRIVATE
END:VTODO
END:VCALENDAR
```

Figure 4.25 vCalendar format (todo)/example

4.4.16 MIDI Melody

In basic and extended EMS, it is possible to insert user-defined melodies in the form of iMelody objects. The iMelody format is a text-based monophonic format that has many limitations and consequently does not allow the definition of sophisticated melodies such as those currently available on most personal computers. Extended EMS allows such sophisticated melodies to be included in messages. These melodies are represented in the format defined in the *Musical Instrument Digital Interface* (MIDI). The information element representing a MIDI melody as an extended object is shown in Table 4.48.

This book does not provide a full description of MIDI. Full MIDI specifications can be obtained from [MMA-MIDI-1].

Table 4.48 IE/polyphonic MIDI melody

IEI		0x14 (extended object) Polyphonic MIDI melody	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1* *Octets 2 and 3* *Octet 4* *Octet 5* *Octets 6 and 7*	*Reference number* *Object length* *Object handling status (UPI + ODI)* *Extended object type* ***0x0C (hex)*** ***MIDI melody*** *Extended object position*
	Extended object definition	Octets 8... *n*	**MIDI melody** These octets represent the melody in the MIDI format as described in [MMA-MIDI-1] or [MMA-SP-MIDI].

4.4.16.1 Introduction to MIDI

Since the release of MIDI 1.0 in 1982, MIDI has become the most widely used synthetic music standard among musicians and composers. The MIDI standard encompasses not only the specifications of the connector and cable for interconnecting MIDI-capable devices, but also the format of messages exchanged between these devices. Only the format of messages concerns extended EMS melodies. Melodies in the MIDI format are represented by a sequence of instructions that a sound synthesizer can interpret and execute. In an EMS-enabled device, this MIDI sound synthesizer may be implemented either as a software-based synthesizer or as a hardware MIDI chipset. Instructions rendered by the sound synthesizer are in the form of MIDI messages. For instance, a MIDI message can instruct the synthesizer to use a specific instrument or to play a given note.

4.4.16.2 MIDI Messages

A MIDI melody is defined as a sequence of MIDI messages. These MIDI messages can be seen as a set of instructions telling the sound synthesizer embedded in the EMS device how to play a melody. For instance, MIDI instructions represent events such as an instrument key being pressed, a foot pedal being released, or a slider being moved. MIDI instructions are mainly directed at one of the 16 logical channels of the sound synthesizer, as shown in Figure 4.26. An instrument can be dynamically assigned to a given channel. Only one instrument can be assigned to a channel at a time. After this assignment, each note to be played by the synthesizer on the channel is played using the selected instrument.

The *polyphony* of a sound synthesizer refers to its ability to render several notes or voices at a time. Voices can be regarded as units of resources required by the synthesizer to produce sounds according to received MIDI messages. Complex notes for given instruments may

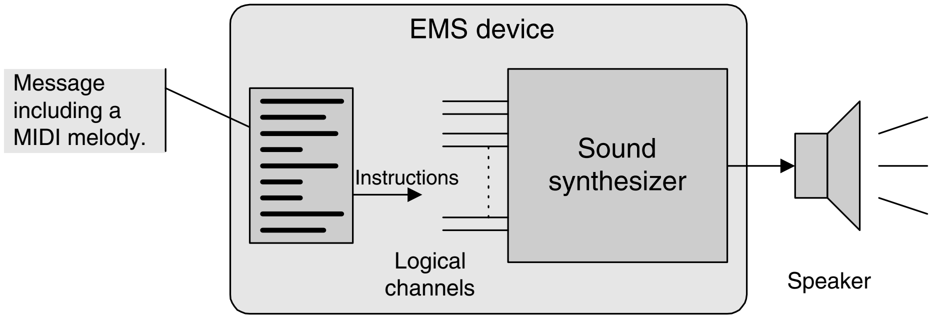


Figure 4.26 Sound synthesizer

require more than one synthesizer voice to be rendered. Early synthesizers were monophonic, meaning that they were able to render only one note at a time. A sound synthesizer with a polyphony of 16 notes, for instance, is able to render up to 16 notes simultaneously.

A sound synthesizer is said to be *multitimbral* if it is able to render notes for different instruments simultaneously. For instance, a sound synthesizer able to render a note for the piano and a note for the saxophone at the same time is a multitimbral synthesizer.

Several types of MIDI messages have been defined and are grouped as shown in Figure 4.27.

Any MIDI message is composed of an initial status octet (also known as status byte) along with one or two data octets (also known as data bytes). The structure of a MIDI message is shown in Figure 4.28.

MIDI messages are divided into channel messages and system messages:

- A *channel message* is an instruction that applies to a selected channel. The identification of the channel, to which a MIDI message is to be applied, is specified as part of the message status octet. Two types of channel messages have been defined: channel mode messages and channel voice messages. *Channel mode messages* affect the way the sound synthesizer generates sounds according to musical information received on one of its

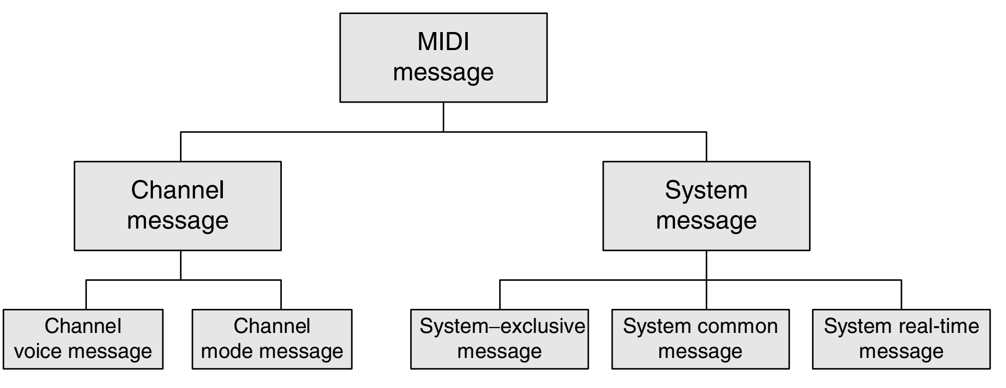


Figure 4.27 MIDI messages types

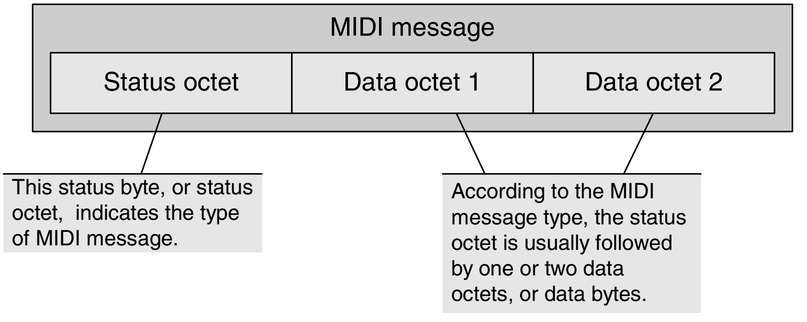


Figure 4.28 Structure of a MIDI message

logical channels. A *channel voice message* contains the musical information that is to be rendered on one specific logical channel. This includes the instructions shown in Table 4.49.

- Unlike channel messages, a *system message* is not an instruction to be applied to a specific channel. A system message is either a system real time message or a system exclusive message. A *system real time message* is used for synchronizing several clock-based MIDI modules together. This message is usually ignored by EMS sound synthesizers. A *system exclusive message* (sysex) allows manufacturers to specify proprietary instructions to be interpreted by their sound synthesizers.

4.4.16.3 General MIDI and MIDI 2.0

After the introduction of early MIDI-capable devices, interoperability problems rapidly occurred while using the MIDI 1.0 format. These problems were due to the lack of a common understanding between manufacturers regarding the identification of notes and instruments. To cope with these issues, the manufacturer Roland proposed an addendum to

Table 4.49 MIDI instructions

Channel voice message	Description
Program change	The program change command is used to specify the instrument, which is to be used for rendering notes on a specific logical channel.
Note on Note off	The rendering of a note is carried out by providing two note-related events to the sound synthesizer: note on and note off. The note on command indicates that a particular note should be rendered whereas the note off command indicates that the note being rendered should be released.
Control change	The control change command indicates to the synthesizer that a key (e.g., pedal, wheel, lever, switch, etc.), is activated. The activation of the key affects the way notes are rendered (e.g., modulation, sustain, etc.).
After touch	This message is used to modify the note being played (after-pressure key).
Pitch bend change	This message is used for altering pitch.

the MIDI 1.0 format in the form of the General MIDI (GM) format. GM supplements MIDI 1.0 by identifying a set of 128 instruments (also known as patches) and a set of notes. Instruments are classified into 16 families as follows:

- piano
- chromatic percussion
- organ
- bass
- ensemble
- reed
- synth lead
- synth effects
- percussive
- guitar
- solo strings
- brass
- pipe
- synth pad
- ethnic
- sound effects.

GM has now been adopted as part of MIDI 2.0.

4.4.16.4 Transport of MIDI Melodies

In theory, MIDI messages can be sent in real time to the synthesizer. With EMS, the MIDI melody is first conveyed as part of a message and later rendered by the sound synthesizer when requested by the subscriber. For this purpose, additional timing information needs to be associated with MIDI messages in order to tell the synthesizer when to play the melody notes. To achieve this, timing information along with MIDI messages are formatted as *Standard MIDI Files* (SMFs).

A melody formatted as a Standard MIDI File belongs to one of the following three SMF formats:

- *SMF format 0*: with this format, all MIDI messages are stored in one single track.
- *SMF format 1*: with this format, MIDI messages are stored as a collection of tracks.
- *SMF format 2*: this format is seldom supported by sound synthesizers.

4.4.16.5 Scalable Polyphony MIDI and 3GPP Profile

Available MIDI synthesizers have various rendering capabilities. Some synthesizers only support a low level of polyphony while others support a high level of polyphony. With EMS the synthesizer is embedded in a mobile device, which usually has limited processing capabilities. In most cases, device resources available to the MIDI synthesizer for rendering sounds depend on the state of other applications being executed in the device. To cope with devices with various capability levels, the MIDI format has evolved to support the concept of *scalable polyphony*. This evolved MIDI format is known as the Scalable Polyphony MIDI (SP-MIDI) [MMA-SP-MIDI]. Scalable polyphony consists of indicating, in the melody, what instructions can be dropped without significant quality degradation when the receiving device is running short of resources. For this purpose, the scalable polyphony information is provided as part of a specific system-exclusive message known as a Maximum Instantaneous Polyphony (MIP) message. The scalable polyphony information indicates the note usage and priority for each MIDI channel.

For instance, a melody composer can generate a MIDI melody that can be best rendered with a maximum polyphony of 16 but can still be rendered by synthesizers supporting a maximum polyphony of 10 or 8 by dropping low-priority instructions.

Regarding the use of SP-MIDI, the MIDI Manufacturers Association (MMA), in cooperation with the Association of Music Electronics Industries (Japan), has defined an SP-MIDI profile to be supported by devices with limited capabilities (e.g. mobile devices). This profile, known as the 3GPP profile for SP-MIDI [MMA-SP-MIDI], provides a means of ensuring interoperability between devices supporting a level of note-polyphony ranging from 5 to 24. This profile identifies MIDI messages and SP-MIDI features, which shall be supported by mobile devices for the sake of interoperability.

4.4.16.6 Recommendation for the Creation of MIDI Melodies

Note that EMS-enabled devices may have various levels of support for MIDI. Simple devices can have support for MIDI 1.0 only, whereas more sophisticated devices could support SP-MIDI. If SP-MIDI is supported, then the device is required to support at least the 3GPP profile for SP-MIDI.

It is important to reduce the resources required to transport MIDI melodies as part of the messages. The size of melodies can be significantly reduced by applying the following recommendations:

- use the SMF format which provides the smallest melody size
- use running status
- use one instrument per track
- use one tempo only for the melody
- use beat, instead of SMPTE, to set the tempo
- remove all controller messages from the melody.

Avoid the use of the following options:

- sequence numbers
- embedded text
- sequence/track name
- instrument name
- lyric
- synchronization markers
- cue point
- MIDI channel prefix
- sequence specific settings.

Due to limitations of EMS devices, content creators should not expect a full support of the following MIDI instructions:

- MIDI message for channel pressure
- MIDI message for pitch bend
- individual stereophonic (pan)
- MIDI message master volume.

Content creators can expect any EMS device supporting MIDI melodies to support a level of polyphony of at least six notes. Consequently, the MIP message of a SP-MIDI melody should specify at least how to render a melody with a device supporting a polyphony of six notes only.

4.4.17 Vector Graphics

In basic and extended EMS, dimensions of bitmap pictures are somewhat limited because of the significant amount of resources required to convey them as part of messages. The bitmap format is not always the most resource efficient way of representing images. A better way of representing images, composed of simple geometrical elements (lines, circles, etc.), is to use a *vector graphic* format. Unlike bitmap formats, a vector graphic format represents an image by identifying geometrical elements composing the image.

For instance, a circle in an image is represented in a bitmap format by a matrix of states for bitmap pixels. The pixel state is white or black, a grayscale level, or a color for, respectively, black-and-white, grayscale, and color bitmap pictures (e.g., a pixel is represented with a 6 bit, state for 64-color bitmap pictures). The same image represented with a vector graphic format consists in indicating which geometrical elements are composing the image. Characteristics of these graphical elements are provided such as the radius, color, and line width for a circle.

It may happen that an image needs to be scaled up or down from its original size to be rendered correctly on the receiving device. For such scaling operations, a clear advantage of the vector graphic format is that the representation of the image on the screen of the receiving device can be recalculated dynamically. Performing such a scaling operation with a bitmap image usually leads to a problem with pixelization where the quality of the image presentation is significantly degraded.

The vector graphic format is well suited for line drawings such as hand-written maps and Asian characters.[5] For this purpose, it is possible, in extended EMS, to insert vector graphic images in messages by using a format called *Wireless Vector Graphics* (WVG) [3GPP-23.040] (from Release 5). An additional interesting feature of WVG is that the format can also be used for the representation of animated images. Three different methods have been defined for inserting a WVG image in an EMS message and are as follows:

- A character-size WVG image is an image with the same height as that of text characters in which it is placed. It is represented with a dedicated information element (not represented as an extended object). This is an appropriate way of inserting basic line drawings as part of the text.

[5] Asian characters can also be encoded with the UCS2 alphabet. However, it is observed that predictive text input mechanisms for complex languages, such as Asian languages, are difficult to design. Vector graphics do offer opportunities for developing more user-friendly built-in features or accessories (touch-screen displays, etc.) for entering text in complex languages with EMS devices.

Table 4.50 IE/character-size WVG image

IEI	0x19 Character-size WVG image	From Release 5
IEDL	Variable	
IED	Octet 1 Octets 2. . .*n*	**WVG image position** This octet represents the position in the text where the WVG image shall be displayed. **WVG image definition** These octets represent the definition of the image in the WVG format.

- A configurable-size WVG image as an independent information element is usually used for representing, in the message, an image for which the dimensions can be configured by the message originator (e.g., a geographical map).
- A configurable-size WVG image as an extended object is also used for representing an image for which the dimensions can be configured. The advantage of representing the image as an extended object is that the image representation can expand over several message segments and compression may be applied to reduce the amount of message space required to transport/store the image.

4.4.17.1 Character-Size WVG Image

A character-size WVG image is used for representing an image for which the height is equal to the height of text characters in which it is placed. For this purpose, a dedicated information element is used for inserting the image in a message. The information element is structured as given in Table 4.50.

The width of a character-size WVG image may be shorter than, equal to, or greater than the width of text characters. The width of the image is determined according to the aspect ratio specified in the image definition and according to the text character height.

4.4.17.2 Configurable-Size WVG Image with Independent Information Element

A WVG image can also be included in a message in the form of a configurable-size image. For this purpose, the information element shown in Table 4.51 can be used. With this method, the image dimensions are specified as part of the image definition.

Table 4.51 IE/configurable-size WVG image

IEI	0x18 Configurable-size WVG image	From Release 5
IEDL	Variable	
IED	Octet 1 Octets 2. . .*n*	**WVG image position** This octet represents the position in the text where the WVG image shall be displayed. **WVG image definition** These octets represent the definition of the image in the WVG format.

Table 4.52 IE/configurable-size WVG image (extended object)

IEI		0x14 (extended object) Configurable-size WVG image		From Release 5
IEDL		Variable		
IED	**Extended object header**	*Octet 1*	*Reference number*	
		Octets 2 and 3	*Object length*	
		Octet 4	*Object handling status (UPI + ODI)*	
		Octet 5	*Extended object type* ***0x0B (hex) WVG image***	
		Octets 6 and 7	*Extended object position*	
	Extended object definition	Octets 8...*n*	**WVG image definition** These octets represent the definition of the image in the WVG format.	

4.4.17.3 Configurable-Size WVG Image as an Extended Object

A method for inserting a WVG image in a message consists of incorporating the image in the form of an extended object. The information element representing a WVG image as an extended object is shown in Table 4.52.

4.4.17.4 WVG Format Definition

The WVG format is a compact vector graphic format. In comparison with the bitmap representation, images can be represented in the WVG format by a limited amount of message space. However, additional processing capabilities are usually required from the EMS device to be able to render vector graphics (integer and floating point calculations). The following WVG features enable very compact image representations:

- *Linear and non-linear coordinate systems*: graphical elements forming a WVG image can be represented using two types of coordinate systems: linear and non-linear systems. Both coordinate systems can be used in the same image and the selection of one against the other is carried out with the objective of minimizing the overall image size.
- *Bit packing*: the representation of numerical values is efficiently compressed by representing these values over a varying number of bits (from 1...16 bits).
- *Global and local envelopes*: local envelopes inserted in a global envelope are used for providing a finer coordinate system for a local zone of the image only.
- *Variable resolution*: different resolutions can be used for representing graphical element coordinates, sizes, angles, etc. The benefit of using a variable resolution resides in the reduction of the number of bits necessary for the representation of numeric values (dimensions, lengths, etc.).
- *Color palettes*: a color palette is provided for an image. This reduces the number of bits necessary to identify colors associated with graphical elements.

- *Default values*: many values characterizing graphical elements and animations can be omitted. If these values are omitted in the definition of the image, then default values are used instead.

An image represented in the WVG format is composed of an image header and an image body. The WVG image header contains elements such as general information (version, author name, date), color palette and default colors, codec parameters (coordinate systems, etc.), and animation settings (looping mode and frame timing). The WVG image body contains graphical elements such as:

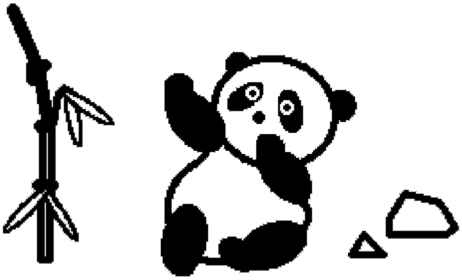
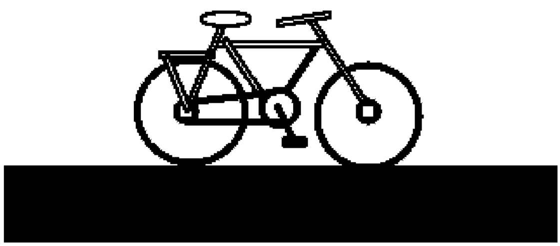
- ellipses
- rectangles
- animations elements
- text elements
- polylines including:
  - simple line polylines
  - circular polylines
- grouping elements
- reuse elements
- frame elements
- local elements
- arbitrary polygons:
  - regular polygons
  - star shaped polygons
  - regular grid elements.

Table 4.53 shows three images. For each image, the size is given for the image represented in the WVG format and compared with the size of the picture represented in the corresponding bitmap format. Note that the two first images are black-and-white images whereas the third image is a color picture (no animation). The size of an image represented in the WVG format is independent of the original image dimensions.

4.4.18 Color for Text Formatting

In Section 4.3.1, a feature for formatting text was presented for basic EMS. This feature allows the text of a message to be formatted on the following aspects: text alignment (left, right, and center), font style (bold, italic, underlined, and strikethrough), and font size (small, normal, and large). In extended EMS, this feature is extended for supporting text foreground and background colors. For this purpose, one octet has been added to the structure of the information element dedicated to text formatting. The structure, which originally had a payload size (IED length) of three octets, now has an additional fourth octet that indicates text foreground and background colors. The evolved information element has the structure shown in Table 4.54.

Table 4.53 Examples of WVG images

Image	WVG size	Bitmap size
 Reproducted by permission of Bijitec[1]	186 octets	Black-and-white bitmap picture 32 × 32 pixels 128 octets 64 × 64 pixels 512 octets
 Reproducted by permission of Bijitec	97 octets	
 Reproduced by permission of Bijitec	208 octets	Color bitmap picture 32 × 32 192 octets 64 × 64 3072 octets

[1]http://www.bijitec.com/

Note that a receiving device supporting the evolved information element should also support the basic version of this information element. A receiving device supporting only the basic information element for text formatting should also be able to interpret partly the extended version of this information element (usually by ignoring the value assigned to the fourth octet of the information element payload).

Table 4.54 IE/text formatting with color

IEI	0x0A Text formatting (with support for color)	From Release 4 Updated in Release 5
IEDL	0x04 (4 octets)	
IED	Octet 1	**Start position** as defined in Section 4.3.1.
	Octet 2	**Text formatting length** as defined in Section 4.3.1.
	Octet 3	**Formatting mode** as defined in Section 4.3.1.
	Octet 4	**Text and background colors** This octet represents the text and background colors to be applied to the identified text section. If this octet is omitted (can be determined by checking the value assigned to the IEDL field), then the information element does not specify any color for text or background. This octet is structured as follows: Bits 7...4 Text foreground color Bits 3...0 Text background color The list of possible colors is the following: 0000 (black) 1000 (gray) 0001 (dark gray) 1001 (white) 0010 (dark red) 1010 (light red) 0011 (dark yellow) 1011 (light yellow) 0100 (dark green) 1100 (light green) 0101 (dark cyan) 1101 (light cyan) 0110 (dark blue) 1110 (light blue) 0111 (dark magenta) 1111 (magenta)

4.4.19 Hyperlink

With extended EMS, an originator SME can include a hyperlink in a message. A hyperlink is composed of a hyperlink title and a URI. In its simplest form, a hyperlink title is just a short text. It can also be a combination of several graphical elements such as text, images, animations, etc. The URI points to a location where additional information, associated with the hyperlink title, can be retrieved. A dedicated information element allows the inclusion of a hyperlink in a message. The dedicated information element is structured as shown in Table 4.55.

Note that this information element only indicates the position of the hyperlink in the message and the length of the hyperlink title and URI. Elements representing the hyperlink

Table 4.55 IE/hyperlink

IEI	0x21 Hyperlink	From Release 5
IEDL	Variable	
IED	Octets 1 and 2	**Absolute hyperlink position** These two octets indicate the absolute position of the hyperlink in the entire text of the message.
	Octet 2	**Hyperlink title length** This octet represents the length of the hyperlink title expressed in number of characters (integer representation).
	Octet 3	**URL length** This octet represents the length of the URI expressed in number of characters (integer representation).

title and URI are conveyed in the text part of the message and are not conveyed as part of the information element itself. This allows earlier SMEs (Release 99/4) that do not support the hyperlink dedicated information element, to still be able to present separately the hyperlink title and the hyperlink URI to the subscriber. In this situation, no graphical association is made between the hyperlink title and the hyperlink URI. If the hyperlink information element is supported by the SME, the SME graphically associates the hyperlink title with the URI. For instance, the hyperlink URI may be hidden and the hyperlink title underlined. In this case, the SME can provide a means for the subscriber to select one of the hyperlinks contained in the message. If the subscriber selects and activates the hyperlink title, then the SME may offer the possibility of launching the microbrowser with the hidden hyperlink URI or alternatively, the SME may offer the possibility of saving the hyperlink as a local bookmark.

The hyperlink in the text part of the message is a character string composed of the concatenation of the hyperlink title, a space character, and the hyperlink URL. All elements (text, image, animation, etc.) for which the position is in the following range are part of the hyperlink title:

[Absolute hyperlink position ... Absolute hyperlink position + hyperlink title length]

Additionally, all characters for which the position is in the following range are part of the hyperlink URL:

[Absolute hyperlink position + hyperlink title length + 1... Absolute hyperlink position + hyperlink title length + 1 + hyperlink title length]

Figure 4.29 shows an example of a hyperlink contained in a message.

4.4.20 Exchange of Capability Information

With extended EMS, devices can exchange capability information. The availability of capability information allows an originator SME to format a message according to whatever the receiving SME is capable of rendering. The capability information indicates which

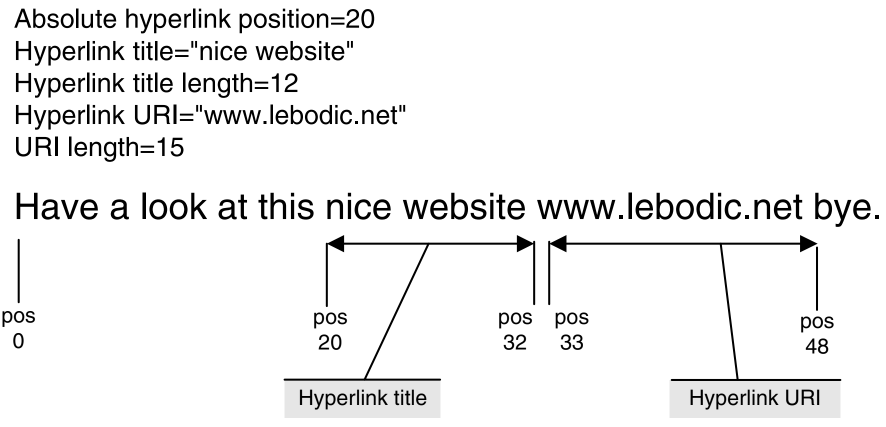


Figure 4.29 Hyperlink information element/example

extended object formats are supported by the associated SME. The method for exchanging capability information consists of the following four successive steps:

1. *Capability request in the message submission*: the originator SME initiates the exchange of capability information by inserting a capability request information element in a message submitted to the SMSC. The message may also contain other information elements, text, and EMS elements. The originator SME shall mark the message for automatic deletion (see Section 3.7.6).
2. *Capability request in the message delivery*: the SMSC delivers the submitted message (with the capability request information element) to the receiving SME in the normal way.
3. *Capability return in the delivery report*: the receiving SME analyzes the content of the delivered message and identifies the request for returning its capability information. In response, the receiving SME inserts a specific extended object information element in the corresponding delivery report. This specific extended object information element informs on the capabilities of the receiving device. After analysis, the initial message may be discarded by the receiving SME without presentation to the subscriber.
4. *Capability return in the status report*: the SMSC receives the delivery report. The extended object contained in the report is copied to the associated status report and sent to the originator SME which initiated the exchange of capability information.

Figure 4.30 shows the four steps leading to the exchange of capability information between a mobile device and an application server.

The originator SME initiates the exchange of capability information by inserting a dedicated information element in a submitted message (step 1). The information element is conveyed to the recipient SME as part of the delivered message (step 2). In this book, this dedicated information is known as the *capability request information element* (also known as the extended object data request command in 3GPP technical specifications). This information element is structured as shown in Table 4.56.

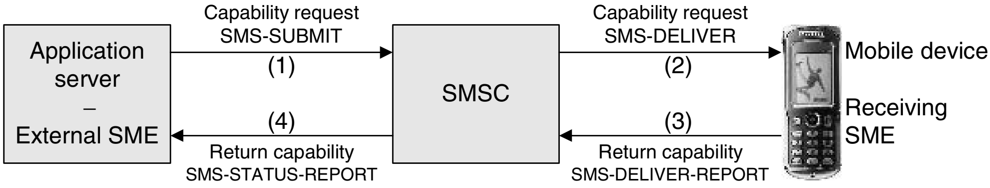


Figure 4.30 Exchange of capability information

Table 4.56 IE/capability request

IEI	0x1A Capability request (a.k.a. Extended object data request command)	From Release 5
IEDL	0x00 (0 octet–no IED)	

In response to the message requesting the provision of capability information, the receiving SME inserts a dedicated information element in the corresponding delivery report (step 1). This dedicated extended object is conveyed to the message originator as part of a status report (step 4). This dedicated capability return extended object is structured as shown in Table 4.57. The value 0 shall be assigned to all unused or reserved bits.

4.4.21 Guidelines for the Creation of Extended Objects

In some situations, an element (sound/melody, image, or animation) may be conveyed in a message as a basic EMS element or as an extended EMS element. Making the assumption that extended EMS devices also support basic EMS features, the following considerations should be taken into account when creating EMS messages:

- If *compression is not supported* by the device generating the message, then it is always better to convey the element as a basic EMS element. In this situation, the element represented in the basic EMS format can be interpreted by the largest audience (basic and extended EMS-enabled devices).
- If *compression is supported* by the device generating the message, then the two following solutions are possible:
  – The ability to reach the largest audience is considered more important than the minimization of the number of segments composing the message. In this situation, conveying the element as a basic EMS element is preferred (compression will not be used).
  – The minimization of the number of segments composing the message is considered more important than the ability to reach the largest audience. In this situation, conveying the element as an extended EMS element is preferred (compression will be used).

Table 4.57 IE/capability return

IEI		0x14 (extended object) Capability return a.k.a. Data format delivery request	From Release 5
IEDL		Variable	
IED	**Extended object header**	*Octet 1* *Octets 2 and 3* *Octet 4* *Octet 5* *Octets 6 and 7*	*Reference number* *Object length* *Object handling status (UPI + ODI)* *Extended object type* ***0x08 (hex) Capability return*** *Extended object position*
	Extended object defination	Octet 8	This octet indicates whether or not extended objects numbered 0–7 are supported by the SME. Bit 0: predefined sounds Bit 1: iMelody sound/melody Bit 2: black-and-white bitmap image Bit 3: grayscale bitmap image Bit 4: color bitmap image Bit 5: predefined animation Bit 6: black-and-white animation Bit 7: grayscale animation Value 1 assigned to a bit means that the associated extended object is supported, otherwise the associated extended object is not supported by the SME.
		Octet 9	This octet indicates whether extended objects numbered 8–12 are supported by the SME. Bit 8: color animation Bit 9: vCard data stream Bit 10: vCalendar data stream Bit 11: vector graphic (WVG) Bit 12: polyphonic melody (MIDI) Bits 13...15: reserved for future use Value 1 assigned to a bit means that the associated extended object is supported, otherwise the associated extended object is not supported by the SME.
		Octets 10. . .*n*	Reserved for future use.

4.5 Pros and Cons of EMS

The Enhanced Messaging Service (EMS) did not really achieve expectations in terms of market uptake. Basic EMS is supported by many handsets but Extended EMS is only supported by a few of them. The window of market opportunity for EMS was small between the success of SMS and the emergence of the Multimedia Messaging Service (MMS). Now that MMS is gaining global support from the industry, it is likely that EMS will never succeed as a widely supported messaging service.

However, EMS may still represent an appropriate service for some specific market needs or for markets where services such as MMS are not available yet. For a network supporting SMS, the availability of EMS-enabled handsets is the only condition for making EMS available to mobile users; no further network investment is required. Unlike EMS, MMS requires significant network investments for mobile network operators in order to make the service a reality. MMS is described in the two next chapters.

4.6 Further Reading

Tutorial on MIDI and Music Synthesis, the MIDI Manufacturers Association, available from http://www.midi.org/about-midi/.

5

Multimedia Messaging Service: Service and Architecture

The Multimedia Messaging Service (MMS) can be seen as the "best of the breed" of several messaging services such as the Short Message Service (SMS), the Enhanced Messaging Service (EMS), and the Internet mail. The first SMS short text message is believed to have been exchanged in 1992. More than 10 years later, MMS multimedia messages propagate over radio channels of major mobile networks. In between, EMS, designed as a rich media extension of SMS, attempted to penetrate the market but without reaching the expected success. More than 100 operators have deployed MMS. Initial MMS offerings were limited to basic messaging features but now more advanced features are proposed for mobile users. Advanced features build-up from basic messaging functions to improve the user experience, from photo messaging to video messaging. MMS is still in its infancy and still has to meet the expectations of the mass market.

From 2002, a first MMS deployment has led to the roll-out of the service in many countries in all continents. This initial market opportunity for MMS relied mainly on the availability of color-screen phones with digital camera and the introduction of packet-based communications in mobile networks. Early MMS implementations allowed a mobile user to exchange multimedia messages not only with other mobile users but also with Internet users. Multimedia messages range from simple text messages to sophisticated messages comprising a slideshow composed of text, images, and audio clips. The roots of the Multimedia Messaging Service lie in the text-based Short Message Service and the Internet electronic mail. Indeed, features already supported by these services have not been forgotten in MMS. MMS supports the management of reports (delivery and read reports), message classes, and priorities and group sending. In addition, MMS differs from other messaging services with its multimedia capabilities, its support for Email and phone number addressing modes, its efficient transport mechanism, and flexible charging framework. From a marketing perspective, the first MMS deployment was mainly regarded as the "photo messaging" service for the mass market of mobile users. It is too early to speak about an overwhelming commercial success for this initial market positioning for MMS. Operators have widely adopted the

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2

service but interoperability issues are still to be solved and penetration of MMS phones has to grow in order to allow a mass adoption of the service by mobile subscribers.

During deployments of initial MMS solutions, open standards for MMS were evolving to enable future service evolutions and solving early interoperability issues. Consequently, improved MMS solutions are now emerging in the market. They leverage initial MMS implementations with the support of new features and new media formats. Certain MMS solutions already support the exchange of sophisticated media objects such as video clips and vector graphics. This will progressively lead to the transport and storage of larger messages. In the context of MMS, the concept of Multimedia Message Box (MMBox) will ease the management of large messages by allowing the storage of multimedia messages in network-based user personal stores (e.g., message boxes, online photo albums, etc.). The wide-scale deployment of these new features is still to be accomplished by network operators. This deployment faces interesting technical and marketing challenges. This chapter and the following one attempt to address some of them.

This chapter places MMS in the patchwork of existing messaging services. It identifies the key success enablers for MMS and compares MMS features with those offered by other messaging services. It then provides a description of the different use cases for MMS and explains how multimedia message can be designed.

5.1 MMS Success Enablers

The commercial introduction of MMS started in March 2002. The future success of MMS is believed to rely on the following five main enablers:

- *Availability and penetration of MMS phones*: mobile users require MMS-enabled phones for composing and sending multimedia messages. Availability of phones is less critical for message reception and viewing since, with message transcoding in the network side, users are often able to send messages to Internet users (via Email) and to users of legacy handsets (non-MMS phones with support of SMS and/or WAP browser). However, a certain market penetration of MMS-enabled phones is required to enable significant revenues. The Global Mobile Suppliers Association[1] believes that a penetration of at least 30% is necessary for MMS to succeed. In 2002, MMS started with a very limited number of MMS phones. At the time of writing, several hundreds of MMS phone models were available, and this figure is increasing at an impressive rate. MMS phones require the support of color screens and are often shipped with a built-in digital still or video camera. Obviously, these multimedia phones are relatively expensive to produce but the mass production of MMS-enabled phones leads to an economy of scale, and this further facilitates increasing the market penetration of these devices.
- *Device interoperability*: the introduction of any new telecommunications service in a multi-vendor environment is often subject to equipment interoperability issues until the service gains a certain level of maturity. Such an interoperability issue occurs, for instance, when two manufacturers of communicating devices interpret a standard differently. In the context of MMS, the number of standards and the number of manufacturers offering solutions are high; therefore, the interoperability risk is proportionally high.

[1] http://www.gsacom.com

Although the MMS standards have been designed with greatest care, too many options sometimes lead to the development of devices conforming to the standard, but which do not interoperate in an efficient manner.

- *Service interworking*: in the context of MMS, service interworking refers to the ability to exchange messages between messaging environments under control of distinct providers (e.g., mobile network operators). This typically requires the establishment of interconnect agreements covering commercial and technical aspects. Initially, service interworking for MMS was seldom ensured. This made the exchange of multimedia messages among subscribers belonging to different MMS environments impossible. At the time of writing, national interworking (i.e., interworking between mobile networks serving in the same country) has been enabled whereas international interworking is still to be accomplished on a global scale.
- *Ease of use*: "snapshot or record and send." The use of MMS should be as easy as this. No time for clicking through complex phone menu items. The use of MMS with the mobile device should be facilitated with dedicated buttons and simplified options, and message sending should be realized with a minimum of track point clicks. Besides the man-machine-interface issues, another cornerstone to achieve ease of use is the availability of pre-configuration methods for MMS settings. This encompasses the storage of default MMS settings during the device manufacturing process, the storage of settings in the SIM (or USIM), or the provisioning of settings over the air (e.g., settings are sent dynamically from the network to the device).
- *Added value for the end-user*: the user should perceive significant added value using MMS compared to other messaging systems such as SMS or Email. Added value of MMS includes its multimedia capabilities, an efficient message transport mechanism, the support of various addressing modes, and management of reports (e.g., delivery and read reports). Added value is also provided by enabling mobile users to enjoy new types of information, entertainment, and other value-added services, such as goal alerts, weather forecasts, etc., in a spam-free environment.

Compared to other messaging services, MMS is in its infancy. Much hype has surrounded MMS, but the service still has to prove that it can fulfill the five success enablers as described above. MMS has the key advantage of having full support from the major market players of the mobile communications industry. Indeed, in a mobile phone market where the penetration rate is high, MMS is an opportunity for device manufacturers to replace the legacy voice-centric phones by selling new sophisticated multimedia phones. Operators regard MMS as the revenue-generating service that is appropriately scaled for recent investments in terms of packet-based transport technologies (e.g., GPRS) and is giving an initial outlook of further multimedia services that are coming with the roll-out of 3G networks. MMS bridges the once closed mobile communications world with the Internet domain, opening the door to the deployment of compelling services by innovative Value-Added Service (VAS) providers. Without any doubt, the entire industry has great expectations for the future of MMS. The future will tell if the initial hype will convert into commercial success.

5.2 Commercial Availability of MMS

Telenor from Norway was the first operator to launch MMS in Europe in March 2002. This initiative was followed by Vodafone D2 (April 2002), Westel Hungary (April 2002), Telecom

Italia Mobile (May 2002), Orange UK (May 2002), Swisscom (June 2002), Orange France (August 2002), T-Mobile Germany/Austria (summer 2002), T-Mobile UK (June 2002), Vodafone UK (summer 2002), Telefonica Moviles Spain (September 2002), and others.

Outside Europe, China Hong Kong CSL launched MMS in March 2002 and was followed, shortly afterwards, by other local operators. In the United States, AT&T Wireless launched MMS in June 2002. In Singapore, Singtel Mobile launched MMS in September 2002 and China Beijing Mobile launched MMS in China in October 2002.

In the first quarter of 2003, more than 100 operators around the world have announced the availability of their MMS services. The service is now available worldwide and MMS is gaining thousands of new users every day.

5.3 MMS Compared with Other Messaging Services

The first usage of the term MMS dates back to 1998. At that time, operators and vendors were looking at opportunities to offer an advanced messaging service for second and third generation mobile systems. Following the success of the Short Messaging Service (SMS), standardization work on MMS was rapidly kicked off. This section describes several messaging services which are close to MMS in terms of underlying concepts and offered features.

5.3.1 SMS and EMS

The roots of mobile messaging in Europe lie in the *Short Message Service* (SMS). In its initial form, SMS is a basic service for exchanging short text messages (with a maximum of 160 simple characters). Despite its limitations, SMS is widely used today and accounts for a significant part of mobile operator revenues. In its most recent form, SMS allows short text messages to be concatenated to form larger messages and several application-level extensions have been designed on the top of SMS as a transport technology. Most notably, the *Enhanced Messaging Service* (EMS) is a standardized extension allowing SMS messages to incorporate rich media such as polyphonic melodies, simple black and white, color or grayscale images/animations, etc.

SMS and EMS are further described in Chapters 3 and 4, respectively.

5.3.2 Electronic Mail

One of the most common uses of the Internet is the *Electronic Mail* (Email). First Email systems were very basic and cumbersome but were quickly improved with the support of group sending, message attachments, automatic message forward, etc. Email has now become the universal messaging service for Internet users. In the past, Email used to be limited to the exchange of plain text messages sometimes with binary attachments. Now, the text part of Email messages can be formatted with HTML, allowing more sophisticated message presentations (inline images, tables, formatted text, etc.).

The Email service is often offered by Internet Service Providers for personal use or provided by corporate IT department for professional use. Email is commonly perceived as a “free” service by users where these users are only accountable for charges incurred for the transfer of data volumes for sending and retrieving messages. This billing model differs from the one of MMS (and SMS) where the user is billed for sending messages but the retrieval of messages is "free of charge" for the recipient.

The Email architecture relies on an interconnection of local Email clients and Email servers. The Email client is used for the composition and sending of messages to the Email server. It is also used for retrieving messages from the Email server. The Email server is responsible for storing messages in user mailboxes and is often interconnected with other Email servers to allow the exchange of messages between distinct Email systems.

The Email client is typically in charge of retrieving messages from the Email server without explicit notification of message availability from the Email server. Retrieval of messages can be triggered explicitly by the user, or the Email client can automatically poll the Email server for messages awaiting retrieval. This polling mechanism is not appropriate for mobile radio systems which still have very limited network bandwidth compared with fixed networks. Furthermore, the size of Email messages can reach several megabytes. Today, such large message sizes are still difficult to manage with most mobile devices. Several phone vendors have attempted to ship devices with embedded Email clients but these attempts have not proved to be very successful. Email extensions have been developed to cope with the limitations of mobile devices. One of the successful proprietary extensions of the Email system is commercially available in the form of the Blackberry service as described later in this chapter.

5.3.3 J-Phone's Sha-mail and NTT Docomo's i-shot

In November 2000, Vodafone K.K. (formerly known as J-Phone), the Japanese arm of Vodafone, launched a new messaging service known as *Sha-mail* (literally stands for "Picture mail" in Japanese). In August 2004, Vodafone K.K. reported 12 million Sha-mail handsets served by its mobile network. Sha-mail is a messaging service for taking pictures with a digital camera built into a mobile phone and sending them to another Sha-mail phone or to an Internet user (electronic mail message with picture as an attachment). A service extension of Sha-mail, known as *Movie Sha-mail*, also allows recording and sending short video clips (up to 5 seconds). Sha-mail messages can be stored in Sha-mail digital albums stored in the network and managed remotely by the user via a Sha-mail phone. With Sha-mail, there is no application or monthly fee and customers are only billed for communication charges (based on volume of data).

NTT Docomo is well known for its *i-mode* services launched in February 1999 in Japan. In August 2004, NTT Docomo claimed that 48 million i-mode users have been provided access to the service. The denomination i-mode refers not only to a technology for accessing the Internet from a mobile phone, it also refers to the entire i-mode value chain including technologies, business model, and marketing. i-mode offers services such as browsing (access to Internet sites with i-mode-tailored contents), downloading (ringtones, Java applications, etc.), and messaging. i-mode messaging, also known as *i-mail*, is basically based on the Internet electronic mail technology as described in the preceding section. The success of i-mode has spread to other countries outside Japan. Several operators have introduced i-mode in Europe (E-plus of Germany, KPN of Netherlands, BASE of Belgium, and Bouygues Telecom of France) and Taiwan (KG Telecom). In response to the success of Vodafone K.K.'s Sha-mail, NTT Docomo counter-attacked with the launch of a new i-mode messaging service known as *i-shot*. With i-shot, users can take pictures with an i-mode phone with a built-in camera. The picture is attached to an electronic mail message and sent to the i-shot server. The i-shot server stores the picture and sends a URL referring to it as part of an Email text message to the recipient(s). During this process, the i-mode server may modify the original picture according to the recipient's i-mode device capabilities. Upon reception of the message, the user reads the text message with the i-mode mail client and can directly launch the browser to fetch the picture identified by the URL. The i-shot service is also open to the Internet. In this context, the message is directly transferred to the recipient Internet user as an Email message with the picture as an attachment. A key advantage of i-shot is that i-shot messages can be fetched and viewed from any i-mode phone shipped with an i-mode browser. An i-shot phone is only required for originating an i-shot message. With i-shot, there is a monthly fee for accessing i-mode services and customers pay for communication charges (based on volume of data).

Vodafone K.K.'s Sha-mail and NTT Docomo's i-shot are messaging services for second generation mobile systems targeted at the mass market of mobile customers. They are proprietary services relying on existing IP-based Internet protocols and controlled by operators (NTT Docomo, Vodafone K.K., and other operator partners). At the time of writing, no third party is known to offer Sha-mail or i-shot services. Both services are open to the Internet.

The success of photo messaging services in Japan seems quite encouraging for the success of MMS in other parts of the world. However, Japan is a more data-driven market with shorter handset-replacement cycles, and therefore one cannot transfer the Japanese experience directly to other markets.

5.3.4 RIM's Blackberry

In the context of mobile communications, it was shown earlier that Internet electronic mail solutions have proven to be very impractical to use without a minimum of adaptation to the constraints of mobile devices and networks. The major barriers to the success of these solutions are the "pull" models for retrieving messages which require frequent accesses to the Email server and the fact that server access protocols are not bandwidth efficient. In order to offer an Internet electronic mail service scaled to the requirements of mobile subscribers, the Canadian company Research in Motion (RIM) designed a set of extensions for the existing Internet Email service. This extended service, offered to subscribers under the denomination *Blackberry service*, bypasses Email inadequacies to the mobile domain by enabling the following features:

- a "push" model for message retrieval
- a compression of messages
- an encryption of messages.

Two main configurations are available for the Blackberry service. The first configuration limits the impact on existing Email architectures by integrating a "desktop" Blackberry application (the Blackberry desktop redirector) in the user's personal computer used for accessing Email messages. When the user is on the move, the desktop application intercepts incoming messages, compresses them, encrypts them, and pushes them to the Blackberry device via a mobile network. The other way round, the user can compose a new message with the Blackberry device. The message is compressed and encrypted by the device and sent via the mobile network to the desktop application. The desktop application receives the message (by polling the Email server), decompresses and decrypts it, and sends it normally to the message recipients as if the message had been sent out directly by the user from his/her personal computer. A more sophisticated configuration of the Blackberry service consists of installing an extension to the email server itself (the Blackberry enterprise server). In the second configuration, the user's personal computer does not have to be left running when the user is on the move. With this configuration, messaging functions performed by the desktop application in the first configuration are performed here by the server extension. In addition, this configuration also allows the synchronization of calendaring and scheduling data between shared corporate databases and remote Blackberry devices.

The Blackberry service first started in North America and has now been deployed in other countries in Europe (e.g., United Kingdom, France, and Germany). The service fulfills particularly well the needs of itinerant professional users, who avoid using laptop computers while on the move (because of long dial-up time for accessing Email servers, etc.). Compared to other messaging services described in this section, the Blackberry service targets professional users rather than the mass market of mobile users.

5.4 Value Proposition of MMS

Why design a new messaging service in the form of MMS when there are so many existing services to choose from? In the late 1990s, SMS usage was booming and major mobile market players were looking for new service opportunities to exploit network resources for the coming years. It was understood that SMS was very limited and mobile messaging services had great margins for improvement. The Internet electronic mail available at this time was not optimized enough for low-bandwidth radio networks and input-limited mobile devices. Japanese photo messaging services were under development in a proprietary fashion and therefore could not meet the market demands in all parts of the world. What was then needed was a universal messaging service offering multimedia features to the mass market of mobile users. The design of MMS started in order to fulfill this need. MMS builds up from SMS, Email, and emerging Internet multimedia technologies. It differentiates itself from other messaging services on the following aspects:

- *Multimedia capabilities*: MMS integrates multimedia features, allowing message contents to be choreographed on the screen of the receiving device. MMS phones typically allow the composition of messages in the form of slideshow presentations composed of sounds, pictures, text, and video clips.
- *Electronic mail and phone number addressing modes*: MMS supports several addressing models, including the Internet addressing mode (e.g., gwenael@lebodic.net for an Internet user) and the phone number addressing mode (e.g., +33607080402 for a mobile user). Consequently, a message can be addressed to a recipient using an Email address or a phone number.
- *Efficient transport mechanisms*: MMS relies on an efficient message retrieval mechanism. When a message is awaiting retrieval, it is stored temporarily on the network side. The network provides a short notification to the recipient mobile device, indicating that a message awaits retrieval. The mobile device can then automatically fetch the message and

notify the user of the reception of a new message. Alternatively, the mobile device can notify the user that a message is awaiting retrieval, and it becomes the user's responsibility to retrieve the message manually at his/her own convenience. Up to now, communications between the MMS phone and the network are performed with binary protocol data units instead of text-based transactions as commonly found over the Internet. This leads to a more optimal use of scarce radio resources.

- *Charging framework*: charging is of key importance for operators since it allows the generation of users' bills according to the billing model in place. MMS offers an extensive charging framework, which can feed any operator billing system. The charging framework leaves freedom to operators for the development of billing models tailored to market specificities.
- *Future-proof open standards and worldwide acceptance*: last but not the least, MMS is the result of a collaborative work led by major market players from the mobile industry. MMS technical specifications are developed in open standardization forums with the continuous objective of designing a future-proof messaging service meeting the requirements of worldwide markets.

5.5 Billing Models

Previous sections showed that billing models for Japanese photo messaging services are based on the volume of data required for uploading and retrieving messages. As for Japanese services, the most efficient transport technology for MMS is the packet-based transmission. Nevertheless, billing models for MMS differ from the ones in place for Japanese messaging services. For MMS, the following billing models are emerging:

- *Flat rate with sending party pays*: with this billing model, the message sender pays for the cost of sending a message to one or more recipients. The message is free of charge for the receiver. The sender pays a flat rate per message and per recipient (around €0.40 per message for each recipient, regardless of message size[1]). Operators usually support postpaid charging (e.g., monthly invoice) but also allow prepaid charging (e.g., prepaid cards) for MMS. In addition, the operator may request the user to subscribe to a data service for accessing the service. The situation is more complex for roaming users where the roaming sender is typically charged a higher fee for sending a message (around € 1 per message for each recipient) and a roaming user may also be charged for receiving a message (around €1 per retrieved message while roaming).
- *Media-type-based charging with sending party pays*: this model is similar to the preceding one except that the message sender pays according to the contents of the message. For instance, the operators can offer different prices for text messages, voice messages, photo messages, and video messages. With this model, sending a 100 KB photo message would cost less than sending a 100 KB video message.
- *Subscription*: with this billing model, the user pays a monthly fee and does not pay for sending or retrieving messages. The operator may limit the number of messages that can be sent for a given period of time.

[1] Initially, the maximum message size was commonly limited to 30 KB. Many operators are now supporting message sizes up to 300 KB.

For messaging between mobile subscribers, the most prominent billing model for MMS was initially the one with a flat rate with sending party pays. Operator favors this model which has proved its efficiency for SMS. One of the positive consequence of applying this model relying on the sending party paying for the delivery of the message resides in the quasi-non-existence of spam in SMS and MMS worlds. However, message sizes are growing and message contents are becoming more sophisticated (e.g., video clips, vector graphics, etc.) and operators are now evolving towards the media-based charging with sending party pays. With such a media-type-based charging, a convergence of messaging services is appearing where the user is only aware that she is sending a message containing a specific content (text, photo, video, etc.) and should not be aware anymore of the underlying bearer service, e.g., SMS or MMS. Media-type charging is expected to greatly improve the user experience of mobile messaging.

The billing of value-added services over MMS is usually based on service subscriptions (e.g., monthly subscription fee), unless the service is subsidized by advertisement. In the latter case, the service becomes free of charge for the end-user.

5.6 Usage Scenarios

Usage scenarios for MMS are often grouped into two distinct categories: *person-to-person* messaging and *content-to-person* messaging. The person-to-person scenario is the prominent use case for initial implementations of MMS. Most recent implementations of MMS also support emerging person-to-content use cases.

5.6.1 Person-to-Person Messaging

The use of MMS in the person-to-person scenario tightly relies on the availability of multimedia capabilities in the phone (e.g., digital or video cameras). These multimedia capabilities may be built into the mobile handset as shown in Figure 5.1 or provided as external accessories that can be connected to the phone. They are used to capture still images and video clips to be inserted in multimedia messages. In this category, *photo messaging* refers to the scenario where the subscriber takes a snapshot of a scene while on the move and sends it as part of a multimedia message to one or more recipients.

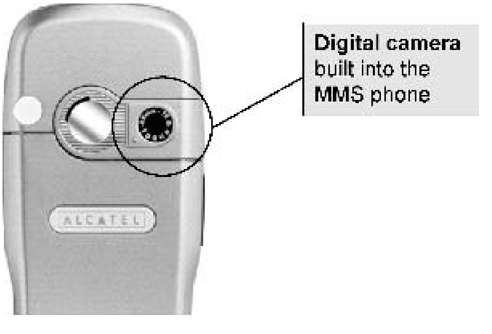


Figure 5.1 Built-in camera in MMS phone – reproduced by permission of Alcatel Business Systems

The user usually has the possibility to send the message to one or more recipients belonging to the following groups:

- *MMS users*: users who have an MMS phone and the corresponding service subscription.
- *Users of legacy handsets*: users who have a legacy phone without support for MMS. For instance, if a user sends a multimedia message (via MMS) to a legacy user, the network can generate a short message and deliver it (via SMS) to the legacy user. The short message contains the address of an Internet page showing the message contents that can be viewed by the legacy user using any Web browser, alternatively using the phone-embedded Wap browser.
- *Internet users*: Internet users can receive multimedia messages originating from MMS users. Multimedia messages, as generated by MMS phones, are not "yet" directly understandable by Email clients such as Microsoft Outlook or Lotus Notes. To cope with this issue, the multimedia message is transcoded in the MMS domain to a more suitable form understandable by Email clients. Note that a transcoded multimedia message may not represent exactly the contents of the original multimedia message. The slideshow structure of multimedia messages is often lost in the transcoding operation.

At the time of writing, there were inter-vendor activities to integrate MMS capabilities in fixedline phones,[1] aiming at allowing fixed and mobile subscribers to exchange multimedia messages. Such a service was successfully launched in Germany in 2004.

5.6.2 Content-to-Person Messaging

In the context of MMS, a *Value-Added Service (*VAS*) provider* is an organization that offers an added value service based on MMS. A VAS application may provide weather notifications, news updates, entertainment services, location-based information, goal alerts, and so on delivered to the phone as a multimedia message. For this purpose, the provider sets up a VAS application, which generates multimedia messages and sends them to one or multiple recipients via the MMS provider infrastructure. In many cases, the user needs to subscribe first to the value-added service in order to receive corresponding messages. The service can be activated by sending a message to the VAS application. Mass distribution of information can be achieved with a value-added service. In order to operate a value-added service, the VAS provider has to establish a service agreement with the MMS provider. In particular, such an agreement specifies how the revenue generated by the value-added service is shared between the MMS provider and the VAS provider. In this content-to-person scenario (also known as machine-to-person scenario), one can distinguish several successful service types, including the ones outlined below:

- *Timed MMS alerts*: the user subscribes to a service consisting of receiving a multimedia message on a regular basis (daily, weekly, etc.). This message typically contains weather forecasts, news updates, horoscopes, etc.
- *Event MMS alerts*: the user subscribes to a service consisting of receiving a multimedia message on the occurrence of specific events. For instance, the message can be sent for breaking news, football goal alerts, etc.

[1] Fixed Line MMS Forum at http://www.fixedlinemms.org.

The support of such services poses interesting technical challenges for enabling network elements (e.g., MMS center, transcoder, radio access network). Let us consider a service consisting of delivering a goal alert to 300,000 subscribers (less than 1.5% of total subscriber base of a large operator such as Vodafone D2 in Germany). One of the service requirements could be that all subscribers shall receive the multimedia message within 5 minutes following the goal. This means that over 5 minutes following the goal, the network elements enabling the service shall process 1000 messages per second.

So far, focus was given to the support of person-to-person use cases. It is now expected that with the increasing level of maturity for MMS, the design of content-to-person applications will be greatly facilitated.

5.6.3 Legacy Support and Interworking Between MMS Environments

Penetration of MMS phones is becoming higher and higher but there are still many legacy phones without MMS capabilities in the market. Any operator providing the MMS service must ensure that special functions are in place to handle multimedia messages sent to legacy phones. When a multimedia message is sent to a legacy phone then an SMS notification is usually sent to the user. This notification contains a URL (and in many cases a login and password) that allows the user to retrieve the message using any Web or Wap browser.

Can users send messages from their home network to users belonging to other network operators? The answers is often ‘‘yes’’ if the two network operators do provide the service in the same country. The answer is ‘‘probably not’’ if the two network operators provide the service in distinct countries. However, mobile network operators are currently working hard to provide message interworking across international borders. It is just a matter of time until interworking is enabled. Whenever a message cannot be sent from one network to another then the message is often treated as if it had been sent to a legacy user in the home network.

5.6.4 Further Applications

Several other applications have made appropriate use of MMS capabilities. For instance, the *postcard service* which consists of sending a multimedia message containing a photo along with a postal address and a greeting text to a specific Email address. Upon receipt of the multimedia message, the service provider prints the photo on the front of a blank postcard along with the greeting text on the back of the postcard. Once printed, the postcard is sent to the recipient(s) (postal address specified as part of the multimedia message) via the conventional post office.

MMS can be considered as a building block enabling the development of other services. For instance, it can be envisaged to develop embedded monitoring applications that regularly take photos of critical sites and send messages with these photos to a remote monitoring center (e.g., Nokia observation camera). These applications typically address the requirements of niche markets.

5.7 Architecture

Before going deeper into the description of features offered by the Multimedia Messaging Service (MMS), it is important to understand the role of each element composing the MMS architecture. This architecture encompasses network elements required for managing MMS devices and routing multimedia messages according to user or service provider instructions in a multi-vendor environment. Network elements communicate over a set of eleven identified interfaces. Interaction protocols for several of them have been standardized to ensure maximum interoperability while others are unfortunately still the subject of proprietary implementations.

One of the key interfaces in the MMS architecture enables communications between the MMS phone and the network element in charge of handling all message transactions. Available realizations of this interface are based on the WAP framework for optimal transfer of messages over bandwidth-limited radio links.

This chapter presents the MMS architecture and the role of its components. Interfaces are outlined in this chapter but an in-depth technical description of transactions that can occur over several of the MMS interfaces is provided in Chapter 6.

The MMS architecture comprises the software messaging application in the MMS phone. This application is required for the composition, sending, and retrieval of multimedia messages. In addition, other elements in the network infrastructure are required to route messages, to adapt the content of messages to the capabilities of receiving devices, and so on. Figure 5.2 shows the general architecture of elements required for the realization of the MMS service.

5.7.1 MMS Environment

The *MMS Environment* (MMSE) refers to the set of MMS elements, under the control of a single administration (MMS provider, e.g., mobile network operator), in charge of providing the service to MMS subscribers. Recipient and originator MMS clients are attached, respectively, to the recipient and originator MMSEs.

5.7.2 MMS Client

The *MMS client* (also known as MMS user agent in the 3GPP terminology) is the software application shipped with the mobile handset, which allows the composition, viewing, sending, retrieval of multimedia messages, and the management of reports. For the exchange of a multimedia message, the MMS client that generates and sends the multimedia message is known as the *originator MMS client*, whereas the MMS client that receives the multimedia message is known as the *recipient MMS client*.

The MMS client offers the following features:

- *Management of messages, notifications, and reports*: devices are commonly shipped with a "unified" message box for the management of MMS elements (messages, notifications, and reports) and other elements such as SMS/EMS messages, WAP push messages, and so on.
- *Message composition*: the message composer is used for creating new multimedia messages.
- *Message viewing*: the message viewer is used to render received messages or to preview newly created messages before sending.

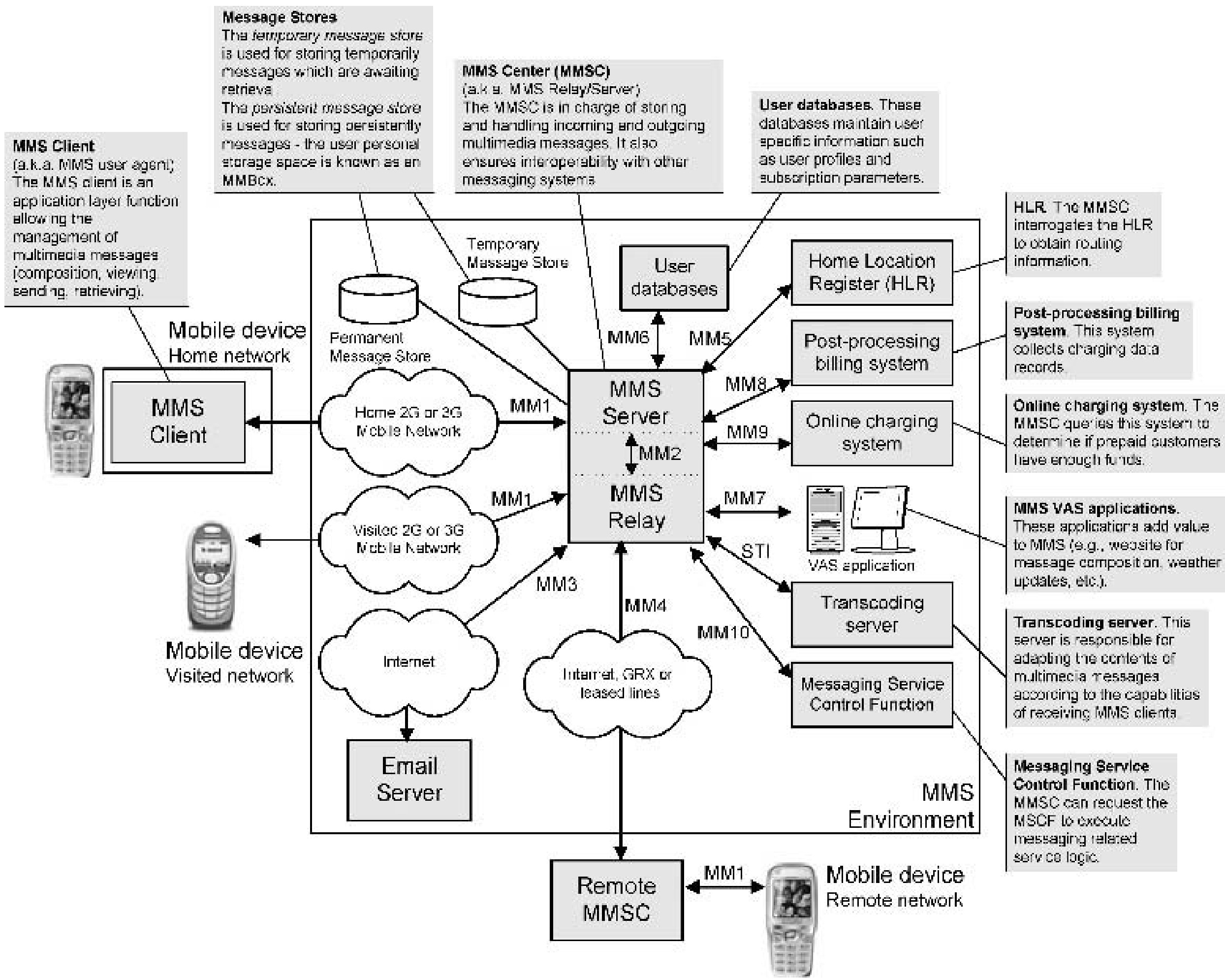


Figure 5.2 MMS architecture

- *Handling of a remote message box* stored in the user personal network-based storage space. Such storage space is known as a Multimedia Message Box (MMBox). The support of an MMBox is optional.
- *Configuration of user preferences and connectivity settings.*

5.7.3 MMS Center

The *MMS Center* (MMSC)[1] is a key element in the MMS architecture. Logically, the MMSC is composed of an MMS relay and an MMS server. The *relay* is responsible for routing messages not only within the MMSE but also outside the MMSE, whereas the *server* is in charge of storing messages. The MMSC server is in charge of temporarily storing messages that are awaiting retrieval from recipient MMS clients. The MMSC may have built-in transcoding capabilities, functions for supporting legacy users, databases for storing user profiles. However, these functions can also be realized with specialized components implemented outside the MMSC.

In order to be able to enjoy MMS, users need to be registered for the MMS service. The MMSC (or alternatively an external user repository) holds profiles for all registered users. Several processes are often available for provisioning new user profiles:

- *Pre-provisioning*: all new users are pre-provisioned for the MMS service as soon as they are giving access to the mobile network. The drawback of this solution is that user profiles are maintained in the system for users who may never use the service.
- *Provisioning by customer care support*: the user calls the customer care support and requests to be registered for the MMS service. The customer care staff uses an online tool for creating the corresponding user profile.
- *Auto-provisioning on first message*: users are automatically registered for the MMS service as soon as they send the first multimedia message. In this case, the MMSC detects that an unregistered user attempts to send a message and automatically creates a new user profile. An even more optimized solution consists of detecting in realtime the type of handset attached to the network and registering/unregistering users according to whether or not their device is MMS capable.
- *Bulk provisioning*: this method is used for registering a significant number of user profiles. This method is typically used for registering previously registered users into a newly deployed system.

In addition to the level of supported features (MMS 1.0, 1.1, 1.2, or 1.3), MMS centers are commonly characterized by their performance (e.g., 100 messages per second system capacity), their level of availability (e.g., high availability with redundant physical elements for coping with failures), support of disaster recovery (e.g., automatic fallback to a geo-redundant site), scalability, and maintainability capabilities.

A typical metric for measuring MMSC performance is the number of *messages per second* the MMSC is capable of delivering. Obviously, this performance metric is highly dependent not only on the traffic model (message size, message contents, number of recipients, etc.) but

[1] The MMSC is also known as the MMS Proxy/Relay (WAP/OMA standards) or the MMS Relay/Server (3GPP standards).

also on interactions with external systems (billing system, transcoder, etc.). Today's MMSCs are typically capable of achieving hundreds of messages per second. Such large capacities are often required for handling traffic peaks during the Christmas period and for the mass delivery of messages for services such as goal alerts.

Mobile operators initially deployed a single MMSC for serving their subscribers. It is now common for operators to deploy several MMSCs (e.g., one for person-to-person traffic and another for content-to-person traffic) or to share MMSCs with other operators (e.g., small operators sharing a single MMSC).

Optionally, the MMSC may also support a persistent message store where users can store messages persistently in their MMBoxes. This feature is particularly useful when devices have limited storage capabilities.

5.7.4 Interfaces

In an MMSE, network elements communicate via a set of interfaces. Each interface supports a number of transactions such as message submission, message retrieval, and message forwarding. Each operation is associated with a set of protocol data units with corresponding parameters (e.g., recipient address, message priority, etc.). Several interfaces have been standardized in order to ensure interoperability between systems designed by various manufacturers. Other interfaces are yet to be standardized and are therefore the subject of proprietary implementations. In this book, interfaces are referred to according to the 3GPP naming convention (MM1, MM2, etc.).

- The *MM1 interface* is a key interface in the MMS environment. It allows interactions between the MMS client, hosted in the mobile device, and the MMSC. Transactions such as message submission and message retrieval can be invoked over this interface. 3GPP has defined the functional requirements of this interface. On the basis of these requirements, the WAP Forum has derived initial WAP-based MM1 technical realizations. The Open Mobile Alliance (OMA) is now in charge of maintaining existing technical specifications for existing MM1 realizations. In addition, OMA is responsible for the development of new MM1 technical realizations for the WAP environment according to high-level requirements defined by 3GPP. This interface is also known as the MMS_M interface in the WAP/OMA standards.
- The *MM2 interface* is the interface between the two internal elements composing the MMSC: the *MMS server* and the *MMS relay*. Most commercial solutions offer a combined relay and server in the form of an MMSC. Consequently, the interface between the two components is developed in a proprietary fashion. At the time of writing, no technical realization of this interface had been standardized, and it is unlikely that one would ever be standardized. This interface is also known as the MMS_S interface in the WAP/OMA standards.
- The *MM3 interface* is the interface between an MMSC and external servers. Transactions invoked over this interface allow the exchange of messages between MMSCs and external servers such as Email servers and SMS Centers (SMSCs). This interface is typically based on existing IP-based Email protocols. MMS standards do not specify exactly how systems should be interconnected, and it is therefore common to adapt this interface to the way the external messaging system already communicates (e.g., Simple Mail Transfer Protocol

for Email). This interface is also known as the E or L interface[1] in the WAP/OMA standards.

- The *MM4 interface* is the interface bridging two MMSCs. This interface is necessary for exchanging multimedia messages between distinct MMS environments (e.g., between two distinct mobile networks). 3GPP has standardized this interface from Release 4. Transactions invoked over this interface are carried out over the Simple Mail Transfer Protocol. This interface is also known as the MMS_R interface in the WAP/OMA standards.
- The *MM5 interface* enables interactions between the MMSC and other network elements. For instance, an MMSC can request routing information from the Home Location Register (HLR) or from a Domain Name Server (DNS).
- The *MM6 interface* allows interactions between the MMSC and user databases (e.g., LDAP user repository). Unfortunately, the MM6 interface is yet to be standardized.
- The *MM7 interface* fits between the MMSC and external Value-Added Service (VAS) applications. This interface allows a VAS application to request services from the MMSC (message submission, etc.) and to obtain messages from remote MMS clients. Prior to 2004, implementations of this interface were all proprietary. 3GPP completed the work on the MM7 interface in the Release 5 timeframe, and commercial implementations of the standardized interface are now available on the market.
- The *MM8 interface* enables interactions between the MMSC and a post-processing billing system. 3GPP has standardized Charging Data Records (CDR) that are generated by the MMSC on the occurrence of certain events (e.g., message submission, message retrieval, etc.). Unfortunately, the interface used for the transfer of CDRs from the MMSC to the billing system has not been standardized yet.
- The *MM9 interface* enables interactions between the MMSC and an online charging system. With this interface, the MMSC can check whether prepaid customers have sufficient funds in their prepaid account to consume requested services. This interface has not been standardized yet.
- The *MM10 interface* allows interactions between the MMSC and a platform implementing a Messaging Service Control Function (MSCF). The MMSC requests the MSCF to execute some message-specific service logic that may influence the addressing, routing, and charging of multimedia messages. The MSCF can also access rights for users. This interface is in the process of being standardized but no standard technical realization is available yet.
- The *Standard Transcoding Interface* (STI) enables interactions between the MMSC and a media transcoder. OMA has standardized this interface.

5.8 Standardization Roadmap for MMS

Standards provide different levels of technical information to allow MMS experts to build interoperable implementations, while always improving the existing enabling technologies. Some standards describe the high level service requirements from which derive other standards dealing with service architecture and interactions between MMS devices. Some standards identify formats and codecs used in the context of MMS whereas others concentrate on billing and charging aspects. 3GPP and OMA have designed major MMS

[1] E stands for Email server and L stands for Legacy mobile messaging server.

standards required for designing MMS solutions. These standards rely on existing generic standards developed by W3C and IETF. Figure 5.3 presents a general organization of 3GPP and OMA MMS standards around the four following specification sets:

- MMS requirement specifications, service aspects, and technical realizations
- MMS codecs and support of streaming
- MMS charging aspects
- MMS-related files in the SIM/USIM.

MMS standards become more and more feature-rich as they evolve over time. At appropriate times, standards are frozen at a given level of features in order to allow vendors to produce compliant devices. In the meantime, standardization engineers carry on the work on a new set of additional features for the next generation of MMS devices. At the time of writing, four levels of features are available for the implementation of MMS solutions. These four levels of MMS features are known as MMS 3GPP Release 99, Release 4, Release 5, and Release 6 corresponding to, respectively, WAP Forum MMS 1.0, OMA MMS 1.1, OMA MMS 1.2, and OMA MMS 1.3. Features corresponding to these four levels are summarized in Table 5.1.

Each standard organization proposes a set of several maturity levels over which standards evolve over time from a draft level to a level of higher maturity (frozen for 3GPP and candidate for OMA). Figure 5.4 shows the availability of MMS standards according to the corresponding level of features (3GPP release/OMA version).

MMS standards can be downloaded from the websites of respective standardization bodies as shown in Chapter 2. Each relevant standard is introduced in Table 5.2.

An online resource page points to all MMS-related standards. This page is available from this book companion website at http://www.lebodic.net/mms_resources.htm.

5.9 WAP Realizations of MMS

In the context of MMS, the three WAP configurations introduced in Section 1.6 can be used. In these configurations, the MMSC plays the role of both application server and push initiator, whereas the MMS client is an application implemented in the WAP mobile device.

At the time of writing, the only configuration that had been deployed was the WAP 1.x legacy configuration. A smooth transition will occur in the future towards the support of configurations without the WAP 1.x gateway. During this transition period, it is expected that operators will support multiple configurations at the same time to ensure that legacy and new devices can operate seamlessly within the same network infrastructure.

The key element of the WAP 1.x legacy configuration is the protocol stack that has been optimized for the transport of data in resource-constrained networks. A drawback of this configuration is that the amount of data that can be exchanged during a single transaction between the mobile device and the network is constrained by the protocol stack limitations. For instance, 300 KB is usually seen as the maximum message size that can be handled by most WAP 1.x legacy environments. Configurations without the WAP 1.x gateway allow the exchange of larger amounts of data for each single transaction between the mobile device and the network. As the size of messages will grow with the support of large media objects

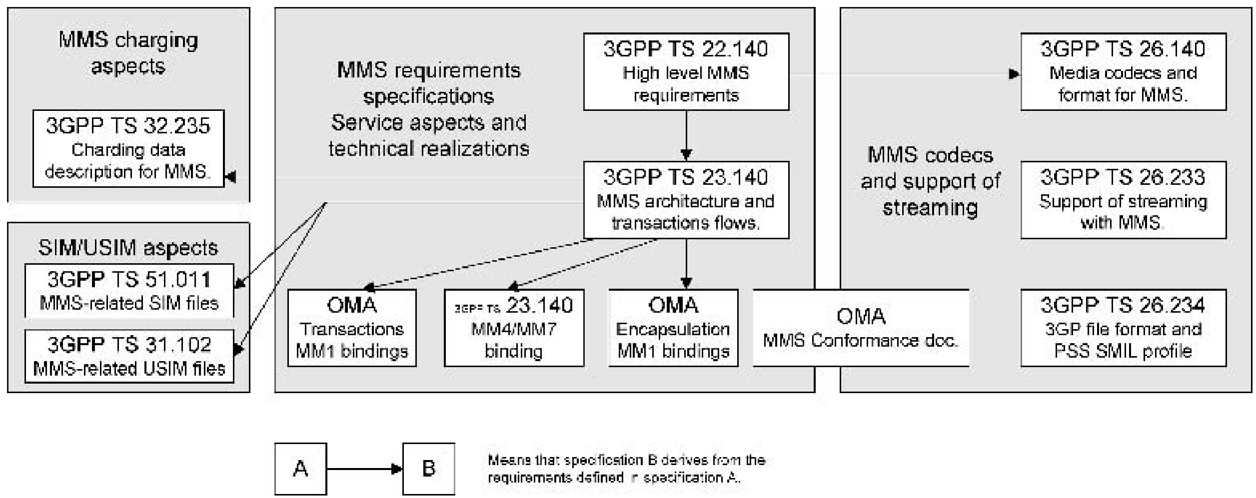


Figure 5.3 MMS standard sets

Table 5.1 MMS features

3GPP standards	WAP Forum OMA standards	Features
MMS Release 99	MMS 1.0	Basic features: - message notification - message sending/retrieval - delivery and read reports - address hiding
MMS Release 4	MMS 1.1	Additional features: - reply charging - forward from notification - enhanced read report management - message sending/retrieval over secure connections - support of MMS settings and notifications in (U)SIM - definition of the MM4 interface
MMS Release 5	MMS 1.2	Additional features: - persistent network-based storage of message (MMBox) - detailed message notification - message distribution indicator - definition of the MM7 interface - definition of the core message content domain and content classes (text, image basic, image rich, video basic, and video rich) - definition of creation modes - transcoding requirements - support of DRM forward-lock
MMS Release 6	MMS 1.3	Additional features: - definition of the content message content domain and contents classes (content basic and content rich) - addition of the megapixel content class to the core message content domain - postcard service - support of DRM combined and separate deliveries

such as video clips, the migration from WAP 1.x legacy configurations to other configurations (with WAP proxy or with direct access) will become necessary for operators.

Figure 5.5 shows a configuration of the WAP environment with a WAP 1.x gateway for the support of MMS.

Box 5.1 Recommendation for maximum group size (WAP/WSP)

In order to guarantee interoperability between MMS phones and networks, it is recommended that the maximum group size for the first group shall not exceed 5120 bytes when (Segmentation And Reassembly) SAR is used for conveying MMS transactions. No specific recommendation is made for packet length and number of packets in a group (see Section 1.6.8).

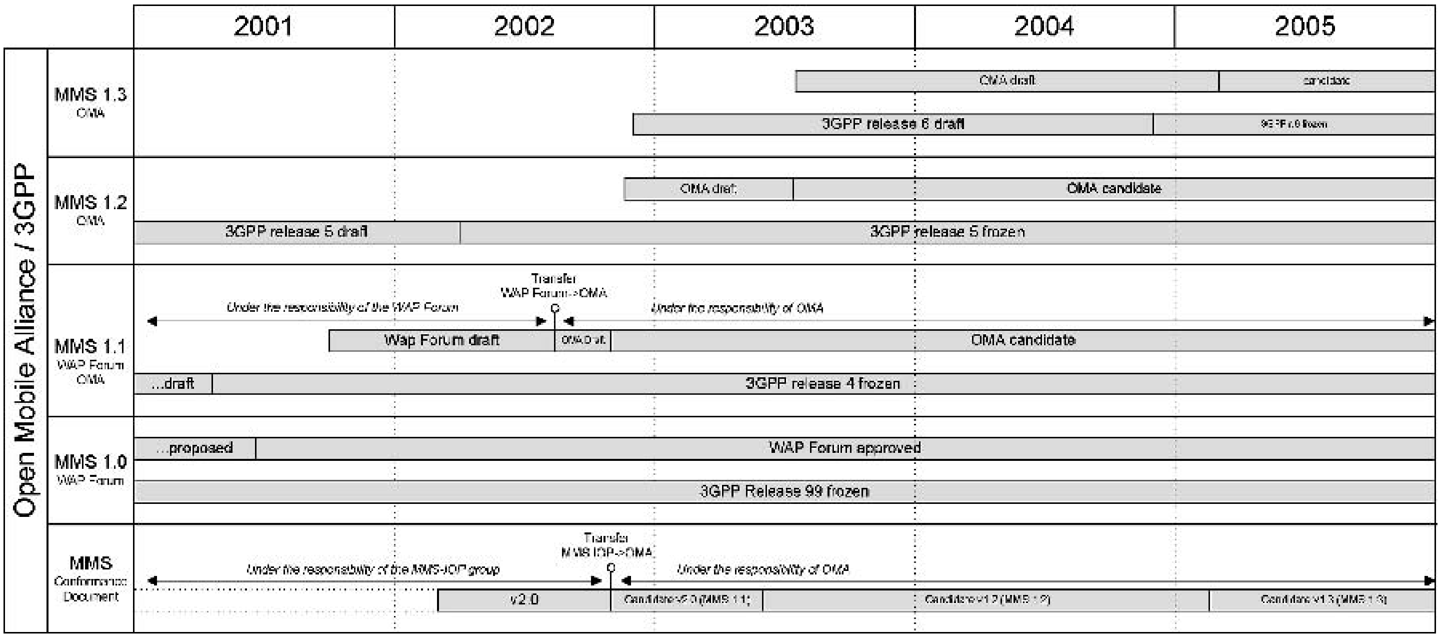


Figure 5.4 Availability of MMS standards

Table 5.2 MMS standards

Responsibility	Available from	Description
3GPP	Release 99 (MMS 1.0)	Title: **MMS service aspects** [3GPP-22.140] This standard provides a set of requirements to be supported for the provision of MMS, seen primarily from the end-user's and service providers' points of view (stage 1).
3GPP	Release 99 (MMS 1.0)	Title: **MMS functional description** [3GPP-23.140] This standard identifies the functional capabilities, the architecture, and information flows needed to support MMS (stage 2). It also provides the details for technical realizations of several interfaces (stage 3–MM4 and MM7).
WAP Forum/OMA	MMS 1.0 (Release 99)	Title: **MMS architecture overview** [WAP-205][OMA-MMS-Arch] This document is an informative document explaining the overall MMS architecture. [WAP-205] covers MMS 1.0 whereas [OMA-MMS-Arch] covers MMS 1.1, 1.2, and 1.3.
WAP Forum/OMA	MMS 1.0 (Release 99)	Title: **MMS client transactions** [WAP-206][OMA-MMS-CTR] This standard details the transaction flows between the MMS mobile device and the MMS center in the WAP environment. [WAP-206] covers MMS 1.0 whereas [OMA-MMS-CTR] covers MMS 1.1, 1.2, and 1.3.
WAP Forum/OMA	MMS 1.0 (Release 99)	Title: **MMS encapsulation protocol** [WAP-209][OMA-MMS-Enc] This standard defines the binary encapsulation of MMS protocol data units. [WAP-209] covers MMS 1.0 whereas [OMA-MMS-Enc] covers MMS 1.1, 1.2, and 1.3.
WAP Forum/OMA	MMS 1.1 (Release 4)	Title: **MMS conformance document** [OMA-MMS-Conf] This document introduces necessary restrictions in terms of transport protocol, and media codecs and formats to allow interoperability between MMS devices and servers. Version 2 of this document is applicable to MMS 1.1, and to some extent to MMS 1.0. After version 2, this document was not released as version 3 but as version 1.2 since it is provided as part of the MMS 1.2 enabler release. In addition, the document defines a profile for the SMIL language, known as the MMS SMIL.
3GPP	Release 5 (MMS 1.2)	Title: **MMS media formats and codecs** [3GPP-26.140] This standard represents the 3GPP recommendations regarding the applicability of media types, formats, and codecs for the MMS service. Prior to Release 5, these recommendations were covered in 3GPP TS 23.140 (Release 99 and Release 4).
3GPP	Release 4	Title: **Streaming service: general description** [3GPP-26.233] This standard provides a general description on how media objects composing a multimedia message can be retrieved via a streaming service in the context of MMS.
3GPP	Release 4	Title: **Streaming service: protocol and codecs** [3GPP-26.234] This standard indicates which protocols and codecs are to be used for the streaming service in the context of MMS. It also defines the file structure (.3GP) for transporting video objects in multimedia messages and a SMIL profile for advanced MMS devices.
3GPP	Release 4	Title: **Charging data description for application services** [3GPP-32.235] This standard defines all MMS related Charging Data Records (CDR). CDRs are required by billing systems for the generation of subscriber invoices.
3GPP	Release 4	Title: **Characteristics of the USIM application** [3GPP-31.102] This standard defines MMS-related USIM elementary files. A Release 99 version exists but does not cover MMS aspects.
3GPP	Release 4 only	Title: **SIM-Mobile equipment interface** [3GPP-51.011] This standard defines MMS-related SIM elementary files. This standard is only available in Release 4 (no SIM standards after Release 4).

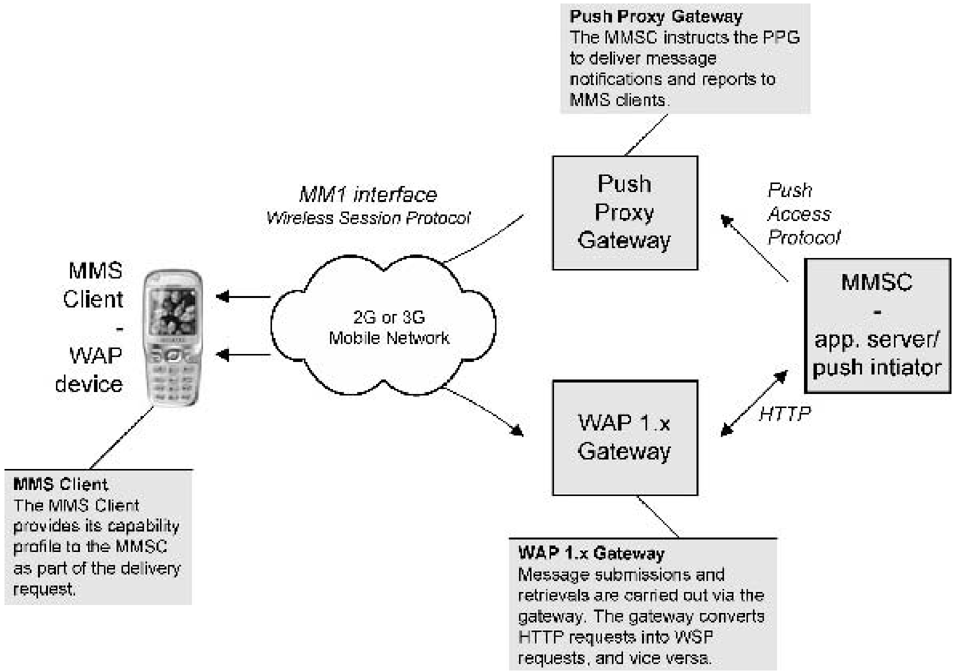


Figure 5.5 WAP 1.x configuration for MMS

5.10 Service Features

MMS offers several features for the support of person-to-person and content-to-person messaging scenarios. These features include sending and receiving multimedia messages, notifying a user that a message is awaiting retrieval, forwarding messages, and managing a network-based box where messages can be persistently stored.

As for any system enabling content sharing, Digital Rights Management (DRM) is of key importance and MMS has mechanisms for controlling the distribution of contents. In the mobile environment, devices have heterogeneous capabilities, making the provision of a homogeneous mobile messaging service difficult. To cope with this issue, MMS relies on a mechanism that adapts message contents to the capabilities of receiving devices.

The MMS framework also introduces flexible charging and billing functions, flexible enough to meet the requirements of many business models.

Next sections provide a functional-level description of all MMS features, pushing in-depth technical explanations to Chapter 6.

5.11 Message Sending

One of the most basic features offered by MMS is the sending of multimedia messages. In the person-to-person scenario, message sending involves several steps as described below:

1. The user composes the multimedia message using a message composer embedded in the MMS-capable device. A multimedia message is typically structured as a multimedia

slideshow. The user, also known as *message originator*, creates/deletes message slides and adds/removes media objects into/from message slides. Attachments can also be included in the message (e.g., electronic business cards).
2. The user instructs the mobile device to send the message to one or more recipients. According to the user's instructions, the originator MMS client (MMS management software built into the mobile device) transfers the message to the MMS Center (MMSC) of the user's MMS Environment (MMSE). This operation is also known as *message submission*. In this context, the MMSC is known as the *originator MMSC*.
3. The originator MMSC performs a number of checks (message format consistency, sufficient prepaid funds, etc.). If the submission is accepted, then the originator MMSC transfers the message to the recipient MMSC(s). Note that if the message is addressed to multiple recipients, then several recipient MMSCs may be involved in the sending process (only if recipients are subscribers from MMS environments distinct from the one of the originator).
4. Upon receipt of the message, the recipient MMSC is responsible for delivering the message to the recipient MMS client as described in the following section.

For a message submission, the message originator (or originator MMS client) can assign a validity period to the message. Once this period has expired, MMSCs involved in the message transfer automatically discard the message if it has not yet been delivered to the recipient(s). If the message originator (or originator MMS client) does not provide a validity period, then the originator MMSC provides one. Additionally, the originator MMSC may also overwrite the validity period specified by the message originator. The basic course of message sending can also be extended with the following features:

- Request for the message to be persistently stored on the network (see also the MMBox concept explained in Section 5.19).
- Request for the originator address to be hidden from recipients (i.e., anonymous message).
- Indication that a message reply will be paid for by the message originator.
- Request for the generation of delivery and read reports.
- Indication of an earliest time of delivery for the message.
- Providing a priority for the message.

In comparison to other mobile messaging services, such as SMS and EMS, large messages up to hundreds of kilobytes can be exchanged with MMS. The sending latency perceived by the user over a GPRS network can typically range from a few seconds to several tens of seconds. With most basic phone implementations, the user does not have access to other phone features while the message is being transferred to the MMSC (i.e., a modal waiting screen is displayed during the entire sending process). With more sophisticated phone implementations, message sending is performed as a background task and, during this process, the user has access to other phone features in the normal way.[1]

[1] With message sending in the background, resource conflicts may occur (e.g., most GPRS phones do not handle simultaneously a voice connection with the data connection required for sending the multimedia message). According to the phone capabilities, conflicting situations are resolved in various ways (e.g., voice call pre-empts the data call, message retransmissions, etc.).

Some early MMS devices do not support multimedia message sending. Such devices are only able to receive messages.

5.12 Message Retrieval

Message retrieval consists of transferring a message from a recipient MMSC down to the local memory of an MMS device. Two retrieval modes have been designed: *immediate* and *deferred* retrievals as defined in the following sections.

The user often has the opportunity to configure the MMS device for operating in immediate retrieval mode (also known as automatic retrieval mode) or deferred retrieval mode (see also screenshots in Figure 5.6). Immediate retrieval is usually the default device setting. According to the receiving device capabilities, the user may be able to indicate a specific retrieval mode to be applied when roaming.

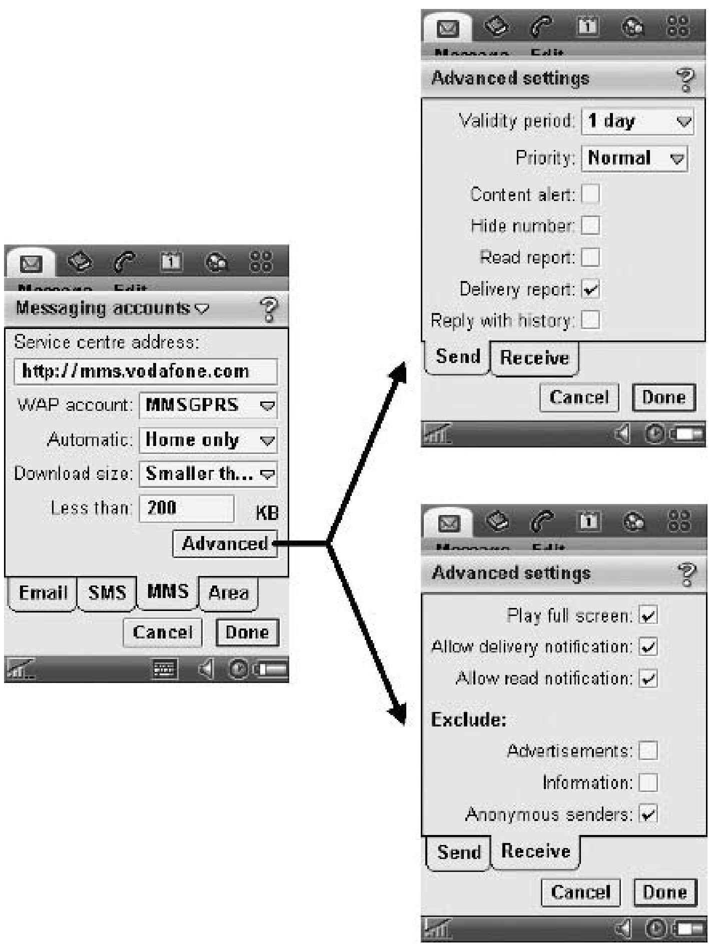


Figure 5.6 MMS settings

Note that messages received and stored in an MMS device may contain media objects, which cannot be modified or redistributed (according to rights associated to the media object by the content provider/owner). In this case, the MMS device may forbid the redistribution of protected media objects contained in a multimedia message.

5.12.1 Immediate Retrieval

From a user viewpoint, immediate retrieval follows the Short Message Service (SMS) message retrieval model. With immediate retrieval, messages are immediately transferred, if possible, to the MMS device once they have been received by the recipient MMSC. In this way, messages appear to be immediately "pushed" to the MMS device without user-specific action. However, multimedia messages can be significantly larger than SMS messages. Consequently, pushing MMS messages without filtering can overload rapidly the MMS device local memory (e.g., internal flash memory). Another drawback of immediate retrieval is that it does not provide an easy way to prevent spamming (i.e., receipt of unsolicited messages). To cope with these different issues, deferred retrieval has been introduced in the MMS standards.

5.12.2 Deferred Retrieval

Deferred retrieval has been designed for coping with some of the limitations of immediate retrieval. Deferred retrieval consists of two successive steps which are as follows:

1. Upon receipt of the multimedia message, the recipient MMSC stores the message temporarily and builds a compact notification. The notification contains information characterizing the message envelope and contents (subject, size, etc.) and is delivered to the message recipient.
2. With the notification, the recipient is informed that a message is awaiting retrieval. With deferred retrieval, it is up to the message recipient to retrieve the message at his/her own convenience. The recipient instructs the message retrieval to the MMS client, which initiates the transfer of the message from the recipient MMSC down to the MMS device local memory.

The two steps mentioned above represent the basic course of message retrieval in the deferred retrieval configuration. Alternatively, upon reception of a notification, a recipient has the opportunity to apply the following actions to the corresponding message:

- *Rejection of the message*: the message is rejected without being retrieved.
- *Forwarding of the message to a remote mailbox or to another recipient*: the message is forwarded to another recipient prior to being retrieved by the MMS device.

Recently, MMS devices without support for message retrieval have appeared on the market. These devices are used for message sending only and are typically used as observation cameras.

In the future, it may be possible for a user to configure more sophisticated retrieval settings. For instance, the user may wish to retrieve immediately high-priority messages only, always defer the retrieval of anonymous messages, and so on.

5.12.3 Retrieval when Roaming

The most common MMS billing model for the person-to-person scenario relies on the sender/originator paying for the transport of the multimedia message (in both originator and recipient environments). With this model, the message is consequently free of charge for the recipient(s). However, the picture is more complex in the roaming scenario, the recipient being attached to a visited network, which is not his/her home network. In this scenario, the user may have to pay an extra fee for resources consumed on the visited network when retrieving multimedia messages. The user usually has the opportunity to set a specific retrieval mode in the case of roaming to cope with possible billing issues. A common configuration of the MMS device consists of immediately retrieving messages when attached to the home network and deferring the retrieval of messages when roaming.

5.12.4 Automatic Rejection of Unsolicited or Anonymous Messages

An advantage of deferred retrieval over immediate retrieval is that it offers a means for the user to check the message characteristics before the retrieval of the message. However, this selective retrieval process has to be carried out manually and this can certainly cause user annoyance if the number of unwanted messages becomes high. To cope with this issue, devices often provide additional features for automatically rejecting multimedia messages fulfilling certain predefined criteria without user-specific intervention. Messages, which are typically rejected, are anonymous messages (originator address is not provided), advertisement messages (identified according to the message class), and messages that are not delivered via the MMSC used for submitting messages.

5.13 Message Reports

A message originator has the opportunity to request the generation of two types of reports for a submitted message: delivery and read reports. It is important to note that, if a message is sent to *N* recipients, then *N* reports of each kind may be returned back to the message originator. In order to preserve recipient privacy, a recipient may deny the generation of delivery and/or read reports.

5.13.1 Delivery Reports

A *delivery report* is generated by a recipient MMSC on the occurrence of the following events:

- The message has been successfully retrieved by the recipient MMS client.
- The message validity has expired and the message has therefore been discarded without being retrieved by the recipient MMS client.
- The message has been rejected by the recipient.
- The message has been forwarded by the recipient.
- The message status is indeterminate (e.g., the message has been transferred to the Internet domain where the concept of delivery report is not fully supported).

In addition to the indication of the event, a delivery report also informs about the date when the event occurred (retrieval, deletion, forward, etc.).

5.13.2 Read Reports

A *read report* is generated by the recipient MMS client on the occurrence of the following events:

- The message has been read by the recipient.
- The message has been deleted without being read.

In addition to the indication of the event, a read report also informs about the date when the message was read or deleted without being read.

5.14 Message Forward

In the deferred retrieval configuration, the recipient can use the notification to request the forwarding of the corresponding message from the recipient MMSC to a remote mailbox (e.g., Email) or to another MMS recipient. This allows a message to be transferred to another location without being retrieved first to the MMS device local memory. This is of particular interest for messages that are too large to be retrieved by the MMS device local memory owing to memory capacity limitations.

Message forward is defined in the standards from MMS 1.1.

5.15 Reply Charging

During the message composition process, the message originator has the opportunity to request for reply charging to be applied to corresponding message replies. Reply charging means that the message originator is accepting to pay for a message reply per recipient. In addition, the message originator typically sets a few conditions to be honored as a requirement for the message reply to be paid for. These conditions are as follows:

- *Reply deadline*: if the reply message is sent after the reply deadline, then the message originator is not willing to pay for it.
- *Reply size*: if the size of the reply message is larger than the reply size, then the message originator is not willing to pay for it.

If the reply charging conditions are honored by the recipient, then the reply is paid for by the message originator and the message reply is therefore ‘‘free of charge’’ for the message recipient. This applies to the first successfully delivered reply for each message recipient. If the message is forwarded by a recipient, then reply charging no longer applies to the forwarded message. Initial implementations will restrict the application of reply charging to MMS clients and Value-Added Service (VAS) applications belonging to the same MMS environment. Note that at the time of writing, available commercial solutions seldom provided support for reply charging.

Reply charging is defined in the standards from MMS 1.1.

5.16 Addressing Modes

Three different modes are used for addressing message recipients and originators which are as follows:

- *Phone number*: the phone number (e.g., +33607180000), also known as a Mobile Station ISDN (MSISDN), is the typical way of addressing another mobile user.
- *Email address*: the Email address (e.g., gwenael@lebodic.net) is a mode used for addressing Internet users.
- *Short code*: a message can be addressed to an alphanumeric short code. A short code usually identifies a service offered by the operator or a value-added service provider in the content-to-person scenario (e.g., "weather" or "888" typed by the user to access value-added services).

5.17 Settings of MMS-Capable Devices

An MMS-capable mobile device needs to be configured in order to operate properly in an MMS environment. The MMS configuration includes *connectivity settings* and *user preferences*.

5.17.1 Connectivity Settings

Connectivity settings for MMS gather all parameters required to access the network infrastructure for sending and retrieving messages, reports, and notifications. This includes the following parameters:

- *MMSC address*: this address is usually provided in the form of a Uniform Resource Identifier (URI) such as http://mms.operator.com.
- *WAP gateway profile*: this profile groups all parameters required to access the WAP gateway. This includes the type of WAP gateway address (e.g., IPv4, IPv6), the WAP gateway address, the access port, service type (e.g., connection-less, secured), the authentication type, authentication identification, and the identification secret.
- *Bearer access parameter* (e.g., GPRS or GSM circuit-switched data): these parameters are required to establish a bearer-level connection with the core network. This includes the bearer type (e.g., CSD, GPRS), parameters of the network access point (e.g., MSISDN for GSM CSD and access point name for GPRS), bearer transfer rate, call type (e.g., analog for CSD), authentication type, authentication identification, and identification secret.

Figure 5.6 shows a set of screenshots of a phone graphical user interface for the user to express preferences and configure connectivity settings.

GPRS is the transport bearer of choice for MMS for most operators who have not yet deployed 3G infrastructures. GSM Circuit Switched Data (CSD) connections are also alternative transport bearers for MMS but at the cost of higher retrieval and sending latencies. Advanced mobile devices are sometimes able to use GPRS connections with a fallback to a GSM data connection whenever the GPRS service is not available. This ensures a high level of service availability with access to high bandwidth GPRS connections whenever available.

5.17.2 User Preferences

User preferences include default parameter values, which are used for the creation of new messages and settings for retrieval modes. Users have the opportunity to update these preferences at their own convenience. Most common preference settings are listed below:

- *Request for a delivery report*: whether or not the user wishes delivery reports to be requested upon message submission.
- *Request for a read report*: whether or not the user wishes read reports to be requested upon message submission.
- *Sender visibility*: whether or not the user wishes his/her address to be hidden from message recipients (anonymous messages).
- *Priority of the message*: the default level of message priority (low, medium, or high).
- *Time of expiry*: the time at which the message expires (e.g., 10 days after submission, 1 month after submission, etc.)
- *Earliest time of delivery*: the earliest time at which the message should be delivered to message recipients (e.g., 1 day after message submission).
- *Message-retrieval mode*: specify whether the message should immediately be retrieved upon notification or retrieved later upon explicit user request. One retrieval mode is typically applied when the user is attached to the home network (message retrieval is free of charge) and another mode when the user is roaming (user may have to pay a roaming charge for retrieving the message).

Other user preferences can include the possibility to automatically reject anonymous or advertisement messages, deferring the retrieval of messages larger than a given size, and so on.

5.17.3 Storing and Provisioning MMS Settings

MMS settings can either be stored in the (U)SIM[1] or in the memory of the MMS phone (e.g., internal flash memory). The latter method is the most commonly supported by available MMS capable devices. Both methods allow the persistent storage of settings.

For the provisioning of MMS settings in the mobile device, the following three solutions are available:

1. *User configuration*: the user usually has the opportunity to configure MMS settings via the device user interface. In this scenario, settings may either be stored in the device internal memory or in the (U)SIM. Alternatively, settings may be scattered over both (U)SIM and device internal memory. If stored in the device memory, the mobile operator can instruct the device manufacturer to assign customized values to MMS settings during the device manufacturing process.

[1] The term (U)SIM identifies either the Subscriber Identity Module (SIM) or the Universal SIM (USIM).

2. *Configuration of (U)SIM-stored settings by the (U)SIM issuer*: the (U)SIM issuer (usually the network operator) can update (U)SIM-stored settings and provide the (U)SIM to the user upon service subscription. Section 5.18 describes how MMS settings are stored in the (U)SIM.
3. *Over-The-Air (OTA) provisioning of settings*: OTA provisioning refers to the possibility of sending parameters dynamically over the mobile network in order to configure a mobile device remotely. This is typically performed by sending a short message (SMS) containing MMS settings to the mobile device. Upon reception, the device updates (U)SIM-stored or device-stored settings. Nokia and Ericsson developed a common proprietary OTA provisioning mechanism for various service settings (including MMS settings) for the provisioning of handsets.[1] However, the Open Mobile Alliance (OMA) recently published the specifications of the following two standardized mechanisms for the OTA provisioning of application settings in the mobile device (including MMS settings):
   - *OMA Client Provisioning*: this mechanism was initially designed by the WAP Forum for the provisioning of WAP browser connectivity profiles and was further extended for the support of application parameters. In addition to WAP browsers, applications that are typically provisioned through this mechanism are Email applications (POP, IMAP, SMTP), MMS clients, IMPS clients, and device synchronization agents [OMA-ClientProv]. Several MMS phones available on the market already support the OMA client-provisioning mechanism for the configuration of MMS parameters.
   - *OMA Device Management*: this mechanism was initially designed by the SyncML initiative. OMA device management allows a management server to access, retrieve, and update application settings stored in the mobile device [OMA-DevMan].

5.18 Storage of MMS Settings and Notifications in the (U)SIM

As shown in the preceding section, MMS settings can be stored in the SIM or USIM. The (U)SIM is provided in the form of a smart card or an electronic chip that can be inserted into a dedicated slot of GSM/GPRS mobile phones. Like MMS settings, message notifications can also be stored in the SIM or USIM. Messages are not stored in the SIM, neither are they stored in the USIM.

3GPP has defined several elementary files that can contain MMS-related data for the SIM and the USIM. As shown earlier in this chapter, only Release 4 of [3GPP-51.011] defines the elementary files for the SIM, whereas similar files are defined for the USIM in [3GPP-31.102] from Release 4 onwards.

Available elementary files for MMS are listed in Table 5.3.

[1] Specifications for this OTA provisioning mechanism are available from the Ericsson developer website at http://www.ericsson.com/mobilityworld.

Table 5.3 MMS elementary files for (U)SIM

Elementary file	Description
EF_{MMSN}	Storage of a notification include: - notification/message status such as: a) notification read or not read. b) corresponding message retrieved, not retrieved, rejected or forwarded. - list of supported implementations (WAP-based MMS, etc.). - content of the notification. - pointer to a notification extension.
EF_{EXT8}	Storage of a notification extension.
EF_{MMSICP}	Storage of MMS issuer connectivity parameters include: - list of supported implementations. - MMSC address. - interface to core network and bearer (bearer, address, etc.). - gateway parameters (address, type of address, etc.). This elementary file may contain a prioritized list of connectivity parameter sets.
EF_{MMSUP}	Storage of MMS user preferences include: - list of supported implementations. - user preference profile name. - user preference profile parameters (sender visibility, delivery report, read report, priority, time of expiry, earliest delivery time).
EF_{MMSUCP}	Storage of MMS user connectivity parameters (same format as EF_{MMSICP}).

5.19 Multimedia Message Boxes

From MMS 1.0, the MMSC temporarily stores messages before they are retrieved by recipient MMS devices. Once messages have been retrieved, they are typically removed from the network temporary store and they are stored in the device local memory (e.g., flash memory) until deleted by the recipient. Owing to limited storage capacities of mobile devices, it is often difficult to store a large number of messages in the device internal memory. Implementations from MMS 1.2 can optionally support the concept of *Multimedia Message Boxes* (MMBoxes). An MMBox is a network-based user message box into which messages are persistently stored. The user can access and update an MMBox remotely from the MMS device or via other means (e.g., web-user interface). With an MMBox, a user has access to the following features:

- Storing persistently submitted and retrieved messages in an MMBox (if configured or requested).
- Requesting the storage of a multimedia message that is referenced in a notification.
- Storing persistently a message that is forwarded to other recipients.
- Uploading and retrieving messages to/from an MMBox.
- Forwarding a message stored in an MMBox.
- Deleting messages stored in an MMBox.
- Viewing messages stored in an MMBox and consulting attributes associated with each message.
- Updating states and flags associated with messages stored in an MMBox.

In an MMBox, a multimedia message can be associated with one of the five mutually exclusive states:

- *Draft*: a message is in the draft state when it has been uploaded and stored but has not yet been submitted.
- *Sent*: a message is in the sent state when it has been stored after submission.
- *New*: a message is in the new state when it has been received by the MMSC and is persistently stored in an MMBox without having been retrieved yet.
- *Retrieved*: a message is in the retrieved state when it has been retrieved by the MMS client.
- *Forwarded*: a message is in the forwarded state when it has been forwarded by the MMS client.

In addition to the state, a message may be associated with a set of user-defined keywords (also known as *flags*). Such keywords enable the MMS client to request the list of messages complying with given keyword search criteria (e.g., professional, personal). *MMBox totals* (number of messages stored in the MMBox) and *MMBox quotas* (in messages or bytes) can be provided to the MMBox user.

5.20 Value-Added Services

Legacy mobile messaging services such as SMS and EMS have very limited support for multimedia contents. These services fulfil the requirements of simple person-to-person scenarios well, but are often too limited for the requirements of more sophisticated content-to-person scenarios. With MMS, Value-Added Service (VAS) providers can generate appealing multimedia contents and push such contents as part of multimedia messages to multiple mobile devices. For this purpose, 3GPP has defined a standard interface, known as the MM7 interface, enabling communications between the MMSC of the MMS provider (e.g., network, operator) and the VAS provider (VASP) server as shown in Figure 5.7. The MM7 interface is based on the Simple Object Access Protocol (SOAP) over HTTP as a transport bearer ([3GPP-23.140] from Release 5). A detailed description of the MM7 interface is provided in Chapter 6.

In this context, the VAS server can submit messages to the MMSC for delivery to one or multiple recipients. After submission, the VAS server has the possibility to cancel the message delivery or replace the message to be delivered if the original message has not yet been delivered. The VAS server can also receive messages from mobile users. In this context, the VAS server is identified by a dedicated short code. Delivery and read reports are also supported over the MM7 interface.

When submitting a message, the VAS server can indicate which party is accountable for the cost of transporting the message over the network. The following configurations are possible:

- The VAS provider pays for the message.
- The recipient(s) pay(s) for the message.
- The cost of handling the message is shared between the VAS provider and the recipient(s).
- Neither the recipient nor the VAS provider pays for the message.

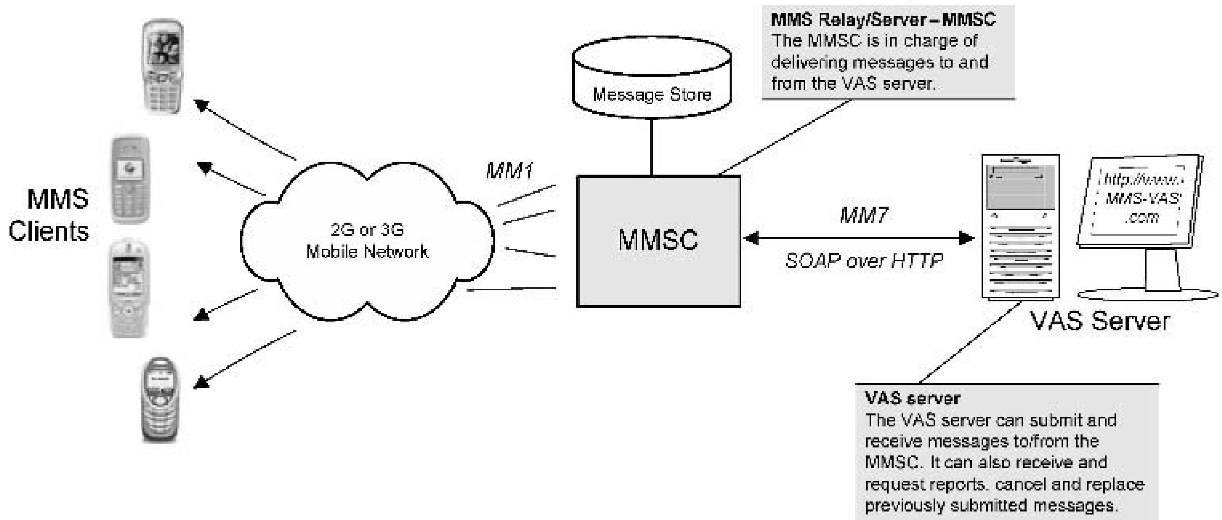


Figure 5.7 General configuration with VAS server

Of course, the VAS provider needs to set up a commercial agreement with the MMS provider (e.g., network operator) in order to apply the chosen billing model for the value-added service.

A VAS provider can achieve mass distribution of information to mobile users with MMS. Furthermore, the provider can also control how message contents are redistributed using appropriate DRM mechanisms. DRM in the context of MMS is defined in the standards from MMS 1.2 [OMA-MMS-Conf].

5.21 Content Adaptation

In an environment in which mobile devices have heterogeneous capabilities, adapting the message contents according to the capabilities of receiving devices can greatly improve the overall user experience. From Release 5, 3GPP mandates the support of content adaptation for mobile devices and highly recommends its support by MMSCs [3GPP-23.140]. In the WAP environment, the user agent profile (UAProf) technique, introduced in Section 1.6.4, is used for this purpose. Content adaptation is performed on messages during the retrieval process and is not applied to message notifications.

The recipient MMS client provides its capabilities to the MMSC during the message retrieval process. Typically, these capabilities are communicated as part of the message retrieval request and the content-adapted message is provided as part of the retrieval response. The recipient MMS client can communicate its capabilities in the following three different forms:

1. Provision of the entire set of device capabilities.
2. Indication of a URL (e.g., http://wap.sonyericsson.com/UAprof/P900R101.xml) pointing to a capability profile in a remote profile repository (usually maintained by the device manufacturer). It is common for MMS providers (e.g., network operators) to copy profiles in their own internal systems to avoid having to rely on external systems but this is at the cost of additional work to update regularly the internal systems with new profiles.
3. Provision of a differential set of information indicating changes to previously advertised capability information.

Content adaptation performed by the MMSC consists of removing media objects that are not supported by the receiving device or adjusting media objects to the capabilities of the receiving device (e.g., media format conversions, reduction of color depth, reduction of video frame rate, replacement of a video clip by an animated image, etc.). The capability profile of a device includes the following MMS characteristics:

- Supported MMS version.
- Maximum supported message size.
- Maximum supported image resolution.
- Maximum supported color depth.
- List of supported content types.
- List of supported character sets.
- Indication of whether or not the device supports streaming.

- List of supported Synchronized Multimedia Integration Language (SMIL) profiles (e.g., MMS SMIL, 3GPP SMIL Release 4, 3GPP SMIL Release 5).
- List of supported message content classes (text, image basic, image rich, megapixel, video basic, video rich, content basic, and content rich).
- Whether or not content adaptation should be disabled.

Figure 5.8 shows the basic course of retrieving a message when content adaptation is applied. In this scenario, the MMSC has built-in transcoding capabilities for adapting the

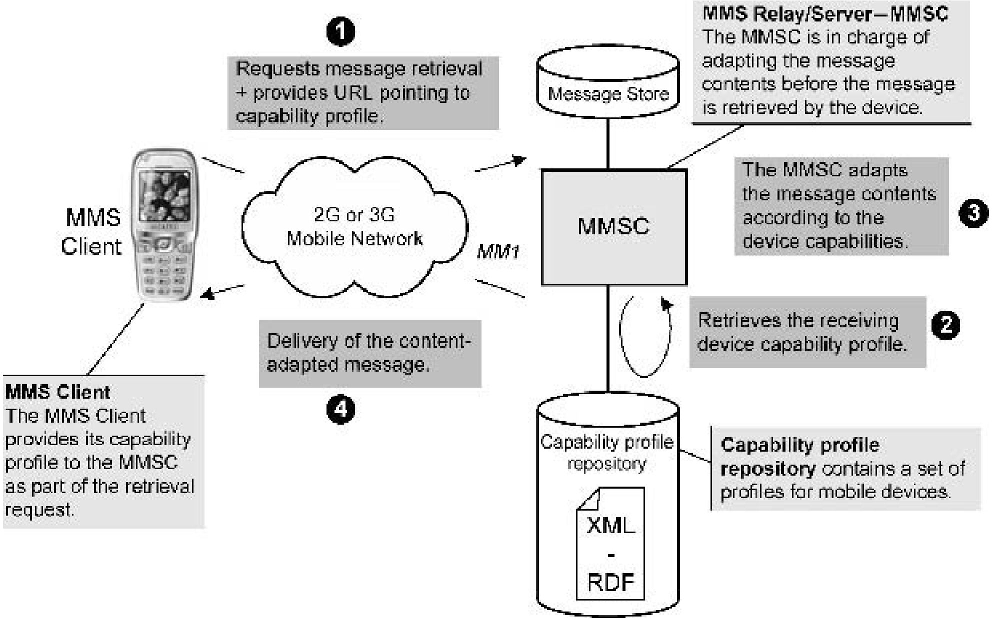


Figure 5.8 Content negotiation

message contents and does not rely on an external transcoder. Note, however, that OMA has standardized the *Standard Transcoding Interface* (STI) that specifies how an MMSC can interact with an external transcoder (possibly shared with other application platforms). The transcoding process with STI is further described in Section 6.12.

As shown in Section 1.6.4, the user agent profile[1] advertises the device hardware and software capabilities and provides the network, WAP, browser, push, and MMS device characteristics. In addition to MMS-specific parameters, generic user agent profile parameters can be used by the MMSC for the content adaptation of multimedia messages. MMS-specific user agent profile parameters are defined in [OMA-MMS-CTR] and are listed in Table 5.4. Figure 5.9 shows the partial representation of the user agent profile for an MMS-capable device.

[1]A list of pointers to known user agent profiles for MMS mobile devices is available from this book's companion website at http://www.lebodic.net/mms_resources.htm.

Table 5.4 MMS characteristics for the user agent profile as defined in [OMA-MMS-CTR]

Attribute name	Description	Examples
`MmsMaxMessageSize`	Maximum message size expressed in bytes.	`30720`
`MmsMaxImageResolution`	Maximum image dimensions expressed in pixels.	`640×480`
`MmsCCPPAccept`	List of supported content types.	"`image/jpeg`", "`audio/amr`"
`MmsCCPPAcceptCharSet`	List of supported character sets.	"`US-ASCII`" or "`ISO-8859-1`"
`MmsCCPPAcceptLanguage`	List of supported languages.	"`en`", "`fr`" for, respectively, English and French.
`MmsCCPPAcceptEncoding`	List of supported transfer encoding methods.	"`base64`", "`quoted-printable.`"
`MmsVersion`	List of supported MMS version.	"`1.0`", "`1.1`"
`MmsCCPPStreaming Capable`	Whether or not the MMS client supports streaming	"`Yes`" or "No"
`MmsSmilBaseSet`	List of supported SMIL profiles supported by the MMS client: • `SMIL-CONF`-1-2 for the SMIL profile defined in [OMA-MMS-Conf] version 1.2 • `SMIL-3GPP-R4` for the SMIL profile defined in [3GPP-26.234] Release 4 • `SMIL-3GPP-R5` for the SMIL profile defined in [3GPP-26.234] Release 4	"`SMIL-CONF-1-2`", "`SMIL-3GPP-R4.`"
`MmsContentClass`	List of message content classes supported by the MMS client: • `TX` for the text class • `IB` for the image basic class • `IR` for the image rich class • `MP` for the megapixel class • `VB` for the video basic class • `VR` for the video rich class • `CB` for the content basic class • `CR` for the content rich class	"`TX`", "`IB`", "`IR.`"
`MmsSuppressionContent Adaptation`	Request that MMSC performs no content adaptation.	"`Yes`" or "`No`"

5.22 Streaming

In the person-to-person scenario, the message delivery process consists of, first, notifying the recipient MMS client and, second, retrieving the complete multimedia message (immediate or deferred retrieval). Once the message has been retrieved, then the message content can be rendered on the recipient device upon user request. This process is suitable for most use cases. However, it is not always possible (e.g., message too large) or efficient to retrieve the

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:prf="http://
www.openmobilealliance.org/tech/profiles/UAPROF/ccppschema-20021212#" xmlns:mms="http://
www.wapforum.org/profiles/MMS/ccppschema-20010111#">
        <rdf:Description rdf:ID="Profile">
                <prf:component>
                        <rdf:Description rdf:ID="HardwarePlatform">
                        ...
                        </rdf:Description>
                </prf:component>
                <prf:component>
                        <rdf:Description rdf:ID="SoftwarePlatform">
                        ...
                        </rdf:Description>
                </prf:component>
                <prf:component>
                        <rdf:Description rdf:ID="NetworkCharacteristics">
                        ...
                        </rdf:Description>
                </prf:component>
                <prf:component>
                        <rdf:Description rdf:ID="BrowserUA">
                        ...
                        </rdf:Description>
                </prf:component>
                <prf:component>
                        <rdf:Description rdf:ID="WapCharacteristics">
                        ...
                        </rdf:Description>
                </prf:component>
                <prf:component>
                        <rdf:Description rdf:ID="MMSCharacteristics">
                                <rdf:type rdf:resource="http://www.wapforum.org/profiles/MMS/ccppschema-
                                20010111#MMSCharacteristics" />
                                <prf:MmsMaxMessageSize>30720</prf:MmsMaxMessageSize>
                                <prf:MmsMaxImageResolution>640x480</prf:MmsMaxImageResolution>
                                <prf:MmsVersion>1.0</prf:MmsVersion>
                                <prf:MmsCcppAccept>
                                        <rdf:Bag>
                                                <rdf:li>image/jpg</rdf:li>
                                                <rdf:li>image/gif</rdf:li>
                                                <rdf:li>image/png</rdf:li>
                                                <rdf:li>image/bmp</rdf:li>
                                                <rdf:li>image/vnd.wap.wbmp</rdf:li>
                                                <rdf:li>application/smil</rdf:li>
                                                <rdf:li>audio/amr</rdf:li>
                                                <rdf:li>audio/midi</rdf:li>
                                                <rdf:li>text/plain</rdf:li>
                                                <rdf:li>text/x-vCard</rdf:li>
                                                <rdf:li>text/x-vCalendar</rdf:li>
                                        </rdf:Bag>
                                </prf:MmsCcppAccept>
                                <prf:MmsCcppAccept-Charset>
                                        <rdf:Bag>
                                                <rdf:li>US-ASCII</rdf:li>
                                                <rdf:li>UTF-8</rdf:li>
                                                <rdf:li>UTF-16</rdf:li>
                                        </rdf:Bag>
                                </prf:MmsCcppAccept-Charset>
                                <prf:MmsSmilBaseSet>
                                        <rdf:Bag>
                                                <rdf:li>SMIL-CONF-1-2</rdf:li>
                                        </rdf:Bag>
                                </prf:MmsSmilBaseSet>
                                <prf:MmsContentClass>
                                        <rdf:Bag>
                                                <rdf:li>TX</rdf:li>
                                                <rdf:li>IB</rdf:li>
                                        </rdf:Bag>
                                </prf:MmsContentClass>
                        </rdf:Description>
                </prf:component>
        </rdf:Description>
</rdf:RDF>
```

Figure 5.9 Example (partial) of user agent profile complete message prior to rendering it. In these situations, it is sometimes possible to split the message content into small data chunks and to deliver the message content, chunk by chunk, to the recipient device. With this process, data chunks are directly rendered on the recipient device without waiting for the complete message to be retrieved. Once data chunks have been rendered by the recipient device, they are usually discarded. This means that the whole process of transmitting the message data chunks has to be carried out again, each time the MMS user wishes to read the message. This enhanced process for performing message delivery and rendering is known as the *streaming process* [3GPP-26.233; 3GPP-26.234].

5.22.1 Example of MMS Architecture for the Support of Streaming

In the MMS environment, it is possible to use the streaming process for the delivery of multimedia messages composed of streamable content. For this purpose, the recipient MMS client and recipient MMS environment must have streaming capabilities. Note that the support of streaming is not mandatory for the MMS client, nor is it for the MMSC. The delivery of streamable multimedia messages involves an additional element in the MMS environment. This element is known as the *media server* and is depicted in Figure 5.10. The media server may be built into the MMSC or might be a separate physical entity in the network.

The delivery of a multimedia message with the streaming process consists of the following five consecutive steps:

1. Upon receipt of the message, the recipient MMSC notifies the recipient MMS client that a multimedia message is awaiting retrieval. The recipient MMS client requests message retrieval (deferred or immediate). Note that this step does not differ from the normal message delivery process (without streaming support).
2. After receipt of the retrieval request, the recipient MMSC decides whether the message should be delivered in the normal way (also known as batch delivery) or in streaming mode. The decision is taken according to the message content, the content adaptation, and/or the user settings and preferences. If the message is to be delivered in streaming mode, then the recipient MMSC copies the message streamable content to the media server.
3. After copying the message streamable content to the media server, the recipient MMSC modifies the multimedia message prior to its delivery to the recipient MMS client. The modification consists of removing the streamable content from the multimedia message and replacing it by a presentation description or by a direct reference to the content to be streamed. The presentation description indicates the streaming transport protocols to be used, the network address of the media server, the characteristics of the content to be streamed, and so on. The presentation description is formatted according to the *Session Description Protocol* (SDP) [RFC-2327].
4. The modified message is delivered to the recipient MMS client in the normal way and stored in the device memory.
5. At a later stage, the message recipient may decide to read the message for which a modified version is locally stored. In this situation, the recipient mobile device detects that the message contains a direct reference or a presentation description referring to a remote streamable content. In order to render the streamable content, the mobile device

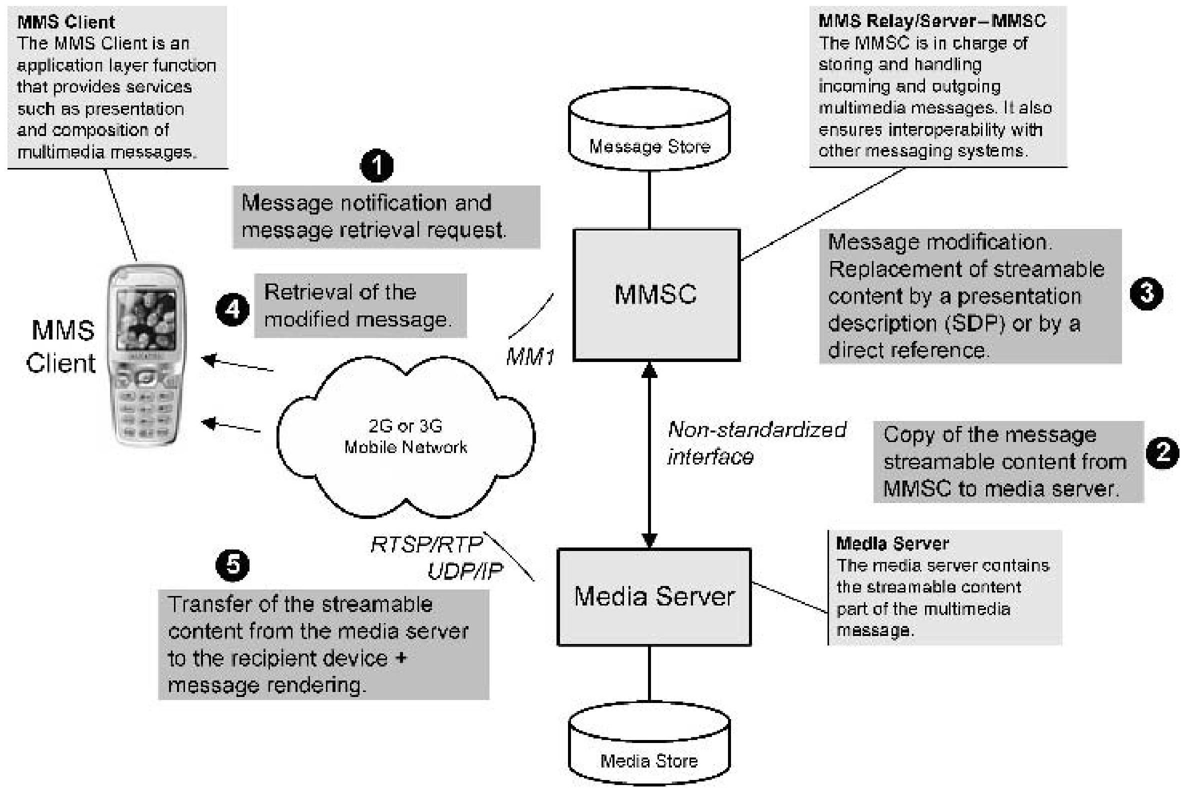


Figure 5.10 MMS environment for the support of streaming

establishes an RTP/RTSP session (e.g., over GPRS) with the media server. The recipient MMS client instructs the media server to start the delivery of the streamable content with the RTSP Play command. Once the streamable content has been rendered by the recipient device, the RTSP/RTP session is released.

In the person-to-person scenario, the way the multimedia message is delivered to the recipient is independent of the way it has been initially submitted by the message originator. In other words, the message originator does not control the way (batch or stream) a multimedia message is delivered to message recipients.

This section has presented the support of streaming delivery in the person-to-person scenario. Note that value-added service providers may also send messages, containing streamable content, to mobile users. Two alternatives are possible in the content-to-person scenario. The first alternative consists of submitting the message, in the normal way, over the MM7 interface. In this situation, the MMSC decides if the message is to be delivered in streaming or batch mode (as in the person-to-person scenario). The second alternative for the value-added service provider consists of submitting to the MMSC a message that contains a presentation description or reference pointing to a media server. In this situation, the value-added service provider decides that the message is to be delivered in streaming mode.

An example of a SMIL scene description containing a direct reference to the content to be streamed is provided in Section 5.29.10.

5.22.2 Streaming Protocols: RTP and RTSP

Two protocols can be used for retrieving some streamable content from a media server: the *Real-time Transport Protocol* (RTP) [RFC-1889] and the *Real-Time Streaming Protocol* (RTSP) [RFC-2326].

RTP is a generic transport protocol allowing the transfer of real-time data from a server. In the MMS environment, RTP enables a one-way communication from the media server to the MMS client. This communication enables the delivery of streamable content, stored in the media server, to the MMS client in streaming mode. RTP usually relies on the connectionless User Datagram Protocol (UDP) (itself relying on the IP protocol). Compared with the connection-oriented Transmission Control Protocol (TCP), UDP allows a faster and more resource-efficient transfer of data. However, UDP lacks a mechanism for reporting loss of data chunks. Consequently, with RTP, lost data chunks are not retransmitted (note that for the transfer of real-time data, transport reliability is not as important as timely delivery). RTP relies on the following three basic mechanisms for the transport of real-time data:

- *Sequence numbering*: UDP does not always deliver data chunks in the order they were sent by the media server. To cope with this, the media server tags data chunks with an incremental sequence number. This sequence number allows the MMS client to reorganize data chunks in the correct order. The sequence number is also used by the MMS client for detecting any loss of data chunks.
- *Time-stamping*: the media server time-stamps all data chunks prior to their delivery to the MMS client. After receiving data chunks, the MMS client reconstructs the original timing in order to render the streamable content at the appropriate rate.

- *Payload-type identifier*: this identifier indicates the encoding/compression schemes used by the media server. According to the payload-type identifier, the MMS client determines how to render the message content.

With RTSP, the MMS client can control the way the streamable content is delivered from the media server. For this purpose, the MMS client instructs the media server to start, pause, and play the message content during the content delivery and rendering. In other words, with RTSP, the MMS client can control the content delivery from the media server in the same way as a person controls a VCR with a remote controller. This is why RTSP is sometimes known as a ''network remote control'' for multimedia servers. RTSP is used for establishing and controlling streaming sessions but is not used for the transport of streamable content. The transport of the streamable content is handled by transport protocols such as RTP. While most streaming content delivery uses RTP as a transport protocol, RTSP is not tied to RTP.

5.23 Charging and Billing

MMS has been deployed worldwide and each operator has adapted its own MMS billing model(s) to local market requirements and MMS usages (e.g., content-to-person, person-to-person, etc.). To enable this, 3GPP has designed a flexible charging framework for the generation and treatment of relevant charging information. The design of this framework is based on generic charging principles identified in [3GPP-32.200] and charging principles specific to MMS as identified in [3GPP-22.140]. The MMSC generates charging information in the form of Charging Data Records (CDRs) on the occurrence of specific events (message submission, message retrieval, etc.). Once generated, CDRs are transferred from the MMSC to the *billing system* over the MM8 interface. Note that, at the time of writing, the MM8 interface was still to be standardized. 3GPP has identified all MMS events for which charging information should be generated [3GPP-32.235]. Corresponding CDRs are categorized into the following five classes:

- *CDRs for the originator MMSC*: this class groups CDRs created by the originator MMSC for events over MM1 and MM4 interfaces.
- *CDRs for the recipient MMSC*: this class groups CDRs created by the recipient MMSC for events over MM1 and MM4 interfaces.
- *CDRs for the forwarding MMSC*: this class groups CDRs created by a forwarding MMSC.
- *CDRs for the MMSC supporting the MMBox concept*: this class groups CDRs created by an originator MMSC supporting MMBoxes.
- *CDRs for VAS applications*: this class groups CDRs created by the MMSC for events occurring over the MM7 interface.

Table 5.5 provides the list of events and corresponding CDR names [3GPP-32.235].

CDRs include information such as the duration of message transmission, customer profile (postpaid or prepaid customer), message content type, message class and priority, message size, reply charging parameters, recipient addresses, and so on.

Two categories of customers are served in mobile networks: postpaid customers and prepaid customers. *Postpaid customers* have a contract with the network operator (e.g., for a period of 1 or 2 years) and are invoiced, typically on a monthly basis. *Prepaid customers* are

Table 5.5 MMS charging data records

Event	Interface	Category	CDR name
Message submission	MM1	Originator	O1S-CDR
Forward request	MM4	Originator	O4FRq-CDR
Forward response	MM4	Originator	OFRs-CDR
Delivery report	MM4	Originator	O4D-CDR
Delivery report	MM1	Originator	O1D-CDR
Read-reply report	MM4	Originator	O4R-CDR
Read-reply originator	MM1	Originator	O1R-CDR
Originator message deletion	n/a	Originator	OMD-CDR
Message forward	MM4	Recipient	R4F-CDR
Notification request	MM1	Recipient	R1NRq-CDR
Notification response	MM1	Recipient	R1NRs-CDR
Message retrieval	MM1	Recipient	R1Rt-CDR
Acknowledgment	MM1	Recipient	R1A-CDR
Delivery report request	MM4	Recipient	R4DRq-CDR
Delivery report response	MM4	Recipient	R4DRs-CDR
Read-reply recipient	MM1	Recipient	R1RR-CDR
Read-reply report request	MM4	Recipient	R4RRq-CDR
Read-reply report response	MM4	Recipient	R4RRs-CDR
Recipient message deletion	n/a	Recipient	RMD-CDR
Forwarding	n/a	Forwarding	F-CDR
Message store	MM1	MMBox	Bx1S-CDR
Message view	MM1	MMBox	Bx1V-CDR
Message upload	MM1	MMBox	Bx1U-CDR
Message deletion	MM1	MMBox	Bx1D-CDR
Message submission	MM7	VAS	MM7S-CDR
Delivery request	MM7	VAS	MM7DRq-CDR
Delivery response	MM7	VAS	MM7DRs-CDR
Message cancel	MM7	VAS	MM7C-CDR
Message replace	MM7	VAS	MM7R-CDR
Delivery report request	MM7	VAS	MM7DRRq-CDR
Delivery report response	MM7	VAS	MM7DRRs-CDR
Read report request	MM7	VAS	MM7RRq-CDR
Read report response	MM7	VAS	MM7RRs-CDR

not directly invoiced by the mobile operator but instead have a prepaid account with the mobile operator into which funds can be deposited, by purchasing prepaid cards for instance. The fund of the prepaid account is decreased of a certain amount each time a service is consumed by the prepaid customer. If the prepaid customer does not have sufficient fund for a requested service then the network refuses to serve the customer.

Two methods are used to deduct money from the prepaid customer's account: *realtime billing* or *hot billing*. Hot billing is similar to post-processing of CDR where processing occurs shortly after the generation of the corresponding CDR. For hot billing under normal conditions, the time period between generating the CDR and deducting the transaction price from the customer prepaid account is typically less than 1 hour.

The support of *real timebilling* for prepaid customers in the MMS environment poses some interesting technical challenges. Let us consider the scenario where a prepaid customer

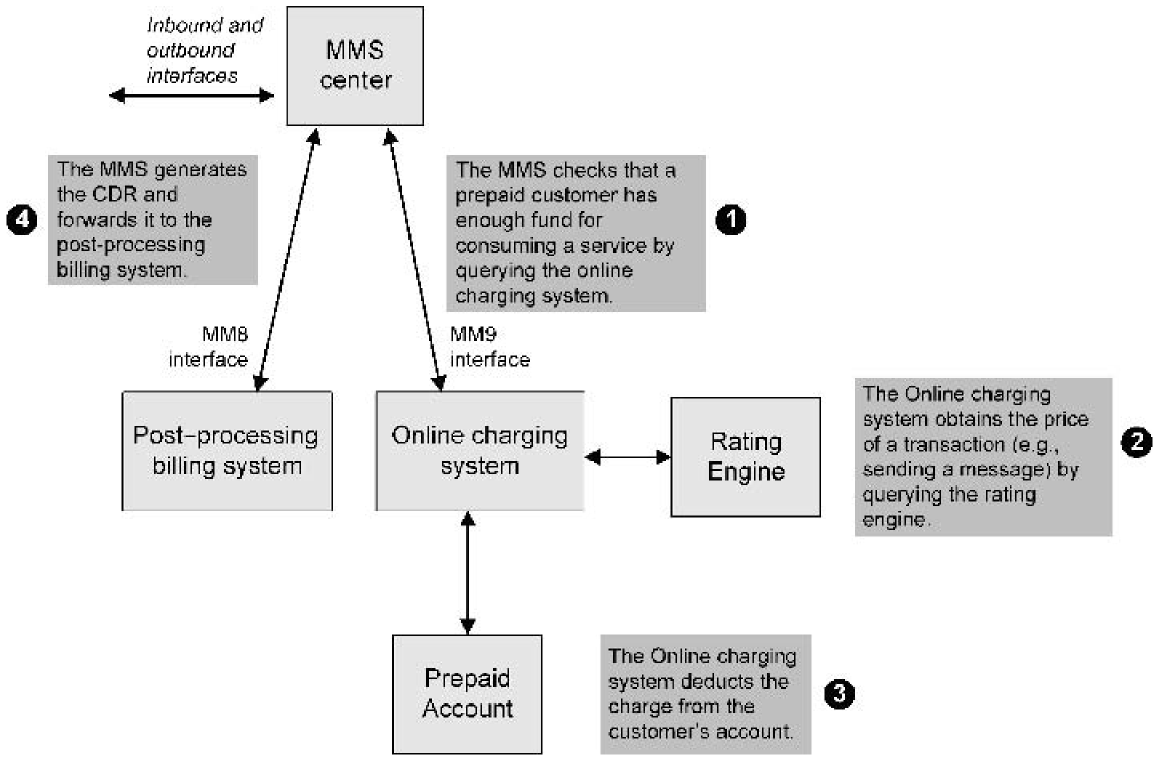


Figure 5.11 Interactions for handling prepaid customers

is sending a multimedia message. For this purpose, the first step as shown in Figure 5.11 consists for the MMS center of providing all relevant information about the transaction (e.g., size of message and destination) to the *online charging system* over the MM9 interface. This online charging system performs a query to a network element called *rating engine* to determine the price of the transaction (step 2). When queried, the rating engine provides a price to the online charging system for the corresponding transaction taking into consideration message characteristics but also according to active promotion plans (e.g., ‘‘happy hour’’ for sending messages, etc.). Once the online charging system has obtained the transaction price, it needs to check that the prepaid customer has enough fund on the prepaid account. This is performed by attempting to deduct the transaction price from the prepaid account as shown in step 3. If sufficient funds are available then the fund in the prepaid account is decreased by the price indicated by the rating engine and the online charging system confirms to the MMS center that the transaction can be accepted. Finally, the MMS center confirms the successful submission of the message to the originating device and generates a CDR to keep record of the transaction. All these steps have to be executed in a very short period of time so that the prepaid customer does not experience latencies in using the service. Later the CDR is provided over the MM8 interface to the *post-processing billing system* (step 4 in Figure 5.11).

Alternatively to the flow described in Figure 5.11, the MMSC may directly interface a rating engine and then requests the online charging system to deduct the transaction price from the prepaid customer account.

For postpaid customers, the rating of an event is performed during the processing of CDRs by the post-processing billing system, which also may do a correlation between CDRs of different events. The information in the CDRs is used for the production of invoices for postpaid customers.

5.24 Security Considerations

It is possible to establish secure connections between the MMSC and the MMS client for the retrieval or submission of multimedia messages. In the WAP environment, MMS relies on Wireless Session Protocol (WSP) or HTTP requests for the transport of messages (see Section 1.6). A message submission or retrieval request includes a Uniform Resource Identifier (URI), which starts with a protocol scheme such as `http` or `https`. The `http` scheme does not imply the use of a particular transport protocol between an MMS client and an MMSC. In this context, communications can be performed over the following protocols:

- Via a WAP 1.x gateway supporting protocol conversion between WSP and HTTP.
- Wireless-profiled HTTP or HTTP.

On the other hand, the `https` scheme implies the use of a secure connection between the MMS client and the MMSC. According to the WAP configuration in place, the secure connection can be established over the following secured protocols:

- Wireless-profiled HTTP or HTTP in accordance with the WAP TLS profile and tunneling specification [WAP-219].

- Via a WAP 1.x gateway using WSP over Wireless Transport Layer Security (WTLS) for the security layer between the MMS client and the gateway and using HTTP over TLS or SSL between the gateway and the MMSC.

It is also possible to secure communications between the MMSC and servers of value-added service providers. HTTP sessions between the VAS server and the MMSC can be established with HTTP over TLS or SSL.

Mobile users are authenticated at the SGSN/GGSN and/or at the WAP gateway levels. This is a pre-requisite for their transaction requests (message sending, message retrieval, etc.) to reach the MMS center.

Authentication techniques defined in [RFC-2617] for “basic” and “digest” authentication can be used to authenticate the VAS provider during the establishment of sessions for message submission. A VAS provider may also be authenticated with a VAS identifier and a password prior to interacting with the MMSC. Note that an authentication mechanism based on public/private key cryptography can also be used in this context.

5.25 Multimedia Message

When compared to other messaging service, one of the major differentiating characteristics of MMS is the organization of message media objects into multimedia presentations. This organization enables the creation of simple point-shoot-and-send messages with feature-limited device-embedded composers (person-to-person) to the design of sophisticated multimedia slideshows with feature-rich professional tools (content-to-person). The internal message structure, which has been selected for MMS, has close similarities with the structure of Internet Email messages. Like most Email messages, multimedia messages are structured as multipart messages containing various media objects such as texts, images, sounds, video clips, and so on.

A multimedia multipart message usually contains a scene description. A scene description tells the receiving device how media objects contained in the message are choreographed on the screen and over time in order to produce a meaningful multimedia presentation.

Following sections present the structure of the multimedia message and describes how to design scene descriptions. It also explains the MMS categorization of media types, which allow devices to interoperate in a coherent fashion.

5.26 Multipart Structure

In the MMS environment, a multimedia message can take multiple forms in order to be efficiently conveyed over the various transport channels composing the full message transfer path. The channel linking the MMS client and the MMS Center (MMSC) is often bandwidth-limited (particularly over the radio part); therefore, multimedia messages are binary-encoded for efficient transfer over this link. Alternatively, the multimedia message is text-encoded for transfers over Internet protocols between MMSCs, from an MMSC towards the Internet domain or from/to Value-Added Service (VAS) servers.

Contents of a multimedia message range from simple text to sophisticated media objects with optional intermedia synchronization. Multimedia message objects are wrapped into an *envelope*, which allows various network elements to route the message towards the recipients

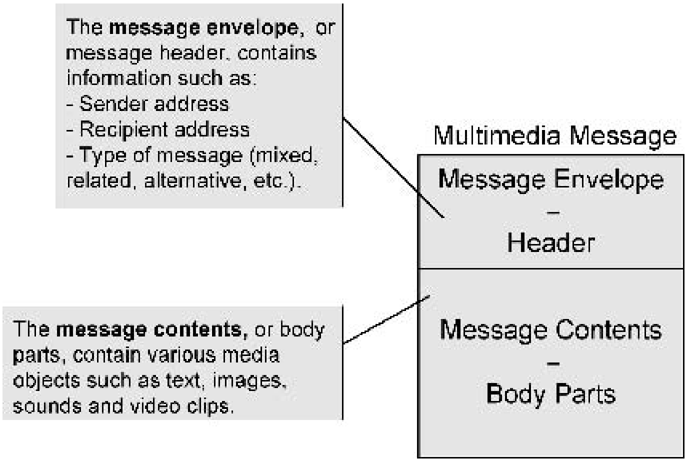


Figure 5.12 Structure of a multimedia message

(addresses of primary and secondary recipients) and which characterize the message contents (class, priority, subject, etc.).

The basic structure of the message envelope is derived from the one defined by the Internet Engineering Task Force (IETF) for the Internet Email [RFC-2822] ([RFC-2822] obsoletes the well-known [RFC-822]). Furthermore, the encapsulation method for inserting media objects into a multipart message is derived from the Multipurpose Internet Mail Extensions (MIME) standards [RFC-2045; RFC-2046; RFC-2047; RFC-2048; RFC-2049]. Media objects are encapsulated in individual object containers known as *body parts*. The WAP Forum and 3GPP have published recommendations indicating how to use these IETF standards (RFC 2822 and MIME) in the context of MMS. Furthermore, the WAP Forum has complemented IETF standards by defining and publishing a generic binary representation of the multipart structure for a more efficient transport over bandwidth-limited bearers. The Open Mobile Alliance (OMA) is now the organization responsible for the maintenance and evolution of this binary representation [OMA-WSP] and of its MMS extensions [OMA-MMS-Enc]. A detailed description of this binary representation for MMS is provided in Section 6.2.11.

Figure 5.12 shows the multimedia message structure relying on a message envelope (message header) and message contents (body parts).

5.26.1 Message Envelope

As discussed above, a multimedia message consists of an envelope (also known as message header) and message contents (encapsulated in message body parts). The envelope informs about the following message characteristics:

- Address of the message originator (From);
- Address of the message recipient(s), organized into primary recipients and secondary recipients (To, Cc, and Bcc);
- Priority of the message (low, medium, or high);

- Class of the message (auto, personal, informational, or advertisement);
- Date and time when the message was sent;
- Validity period;
- Reply charging parameters;
- Request for delivery and/or read reports;
- Message subject;
- Sender visibility;
- Earliest delivery time;
- Message distribution indicator;
- MMBox status.

Each of these characteristics has a dedicated set of parameters. Parameters are either defined in [RFC-2822] or specifically defined for MMS. In the latter case, the name of the parameter is prefixed with `X-MMS`. Table 5.6 shows common parameters that can be included in a message envelope/message header.

The value assigned to the `Content-Type` parameter of the message header indicates how the message body parts are organized in the message. The two values presented in Table 5.7 can be used in the context of MMS.

Boundary delimiters separate multiple parts of a message. The name of a boundary delimiter is assigned to the subtype parameter called *boundary*. The name of a boundary delimiter has a size ranging from 1 to 70 characters. With a multipart textual representation, each part begins with two hyphens (--) followed by the boundary delimiter name. The multipart message is terminated by a carriage return/line feed followed by two hyphens, the boundary delimiter name and two additional hyphens.

In theory, multipart structures can be nested in upper multipart structures in order to build a complete hierarchy of body parts. In the context of MMS, the most common structure consists of having a main multipart structure (mixed or related) containing all body parts composing the message (images, sounds, video clips, etc.).

5.26.2 Encapsulation of Media Objects

In addition to the envelope, multimedia messages contain media objects such as texts, images, sounds, and video clips. Each media object is encapsulated in a container known as a body part. The overall multipart structure of a multimedia message is depicted in Figure 5.13.

Each body part is associated with a number of header parameters as shown in Table 5.8. The only mandatory parameter for a body part is the `Content-type` parameter.

5.27 Message Content Domains and Classes

As shown in previous sections, a message is composed of a header (envelope) and body parts. In theory, any media object with a proper body part encapsulation may be inserted in a multimedia message. Practically, a minimum set of common formats and codecs has been identified in order to ensure interoperability between MMS phones produced by different vendors.

3GPP and 3GPP2 made recommendations regarding the support of formats and codecs in the context of MMS ([3GPP-23.140] prior to Release 5 and [3GPP-26.140] from Release 5

Table 5.6 Message envelope/header parameters[1]

Parameter name	Description
`To`	Address of primary message recipient(s).
`Cc`	Address of secondary message recipient(s). Carbon copy.
`Bcc`	Address of secondary message recipient(s). Blind carbon copy.
`From`	Address of message originator.
`Date`	Date and time when the message was sent.
`Message-ID`	A unique identifier for the message. This helps correlating delivery and read reports with the original message.
`Subject`	Subject of the message.
`X-Mms-Expiry`	Validity period of the message. After this date, MMSCs can discard the message if it has not yet been delivered to the recipient.
`X-Mms-Delivery-Time`	The message originator can specify an earliest delivery time for the message.
`X-Mms-Distribution-Indicator`	A value-added service provider may use this flag to indicate that the message cannot be redistributed freely.
`X-Mms-Reply-Charging`	Whether or not the message originator has requested reply charging.
`X-Mms-Reply-Charging-Deadline`	The reply must be sent before the specified deadline for the reply to be paid for by the message originator.
`X-Mms-Reply-Charging-Size`	The size of the message must be smaller than the specified size for the reply to be paid for by the message originator.
`X-Mms-Reply-Charging-ID`	A unique identifier for the reply charging transaction.
`X-Mms-Delivery-Report`	Whether or not the message originator requested a delivery report to be generated.
`X-Mms-Read-Reply`	Whether or not the message originator requested a read report to be generated.
`X-Mms-Message-Class`	The class of the message (auto, personal, informational, advertisement).
`X-Mms-Priority`	The priority of the message (low, medium, or high).
`X-Mms-Sender-Visibility`	Whether or not the message originator requested his/her address to be hidden from recipients.
`Content-type`	The content type of the message. A description of values that can be assigned to this parameter is provided below in this section.

[1] For the sake of clarity, parameters related to the network storage of messages (MMBox) are not presented in Table 5.6. An exhaustive list of parameters is provided in Chapter 6.

for 3GPP devices and [3GPP2-C.P0045] for 3GPP2 devices). However, 3GPP/3GPP2 recommendations are not restrictive enough to ensure sufficient interoperability between MMS phones. In 2001, an informal group of vendors known as the MMS interoperability group (MMS-IOP) met to design a specification restricting the number of formats and codecs supported by early MMS phones with the objective to ensure interoperability. This specification, referred to as the *MMS conformance document*, constituted the basis for the design of all early commercial MMS devices. Considering the importance of guaranteeing interoperability between MMS devices, the responsibility for the MMS conformance

Table 5.7 Multipart messages/content types

Content-type value[1]	Description
`Multipart/Mixed` or `application/vnd.wap.multipart.mixed`	A mixed multipart structure contains one or more body parts. The order in which body parts appear has no significance. This structure is commonly used when the message does not contain a scene description.
`Multipart/Related` or `application/vnd.wap.multipart.related`	A related multipart structure [RFC-2557] is used for aggregating multiple body parts into a single structure. The optional `Start` parameter refers to a starting body part. For instance, in a multimedia message, the `Start` parameter typically refers to a scene description in the SMIL format.

[1] WAP registered content-types are listed at http://www.wapforum.org/wina/wsp-content-type.htm.

document was transferred in 2002 to the Open Mobile Alliance. OMA published its first MMS conformance document, part of an enabler release, as MMS conformance document version 2.0.0 [OMA-MMS-Conf] (MMS 1.1). The MMS conformance version 2.0.0 is applicable to devices compliant to MMS 1.1 and to some extent MMS 1.0. The successor version of this document was not released as version 3.0 but as version 1.2 since it is published as part of the MMS 1.2 enabler release. Version 1.2 builds up from the MMS conformance document version 2.0.0 by adding the support of video and introducing the concept of message content domains and classes. The MMS conformance document version 1.2 is not only applicable to devices compliant with MMS 1.2 but also to devices compliant with previous versions of the MMS standards (MMS 1.0 and MMS 1.1). It primarily targeted devices whose commercial availability was in the 2004 timeframe. At the time of writing, OMA was in the process of freezing version 1.3 of this document. Version 1.3 primarily targets devices whose commercial availability is in the 2005/2006 timeframe.

The OMA MMS conformance document makes a clear distinction between formats and codecs that have to be supported by 3GPP devices and the ones that have to be supported by 3GPP2 devices. *3GPP devices* are devices compliant to 3GPP standards and designed for European, Asian, and African markets.

Box 5.2 Support of MMS formats/codecs: 3GPP TS 26.140, 3GPP2 C.P0045, and/or OMA MMS conformance document?

Three MMS standards provide recommendations regarding the support of formats and codecs in the context of MMS. Which one(s) is/are applicable, 3GPP TS 26.140, 3GPP2 C.P0045, and/or the OMA MMS conformance document? Currently, the OMA MMS conformance recommends the use of a group of formats/codecs that constitute two subsets of the ones identified in 3GPP TS 26.140 and 3GPP2 C.P0045. In other words, the MMS conformance document is more restrictive than the two others. Consequently, it is advised to design solutions following the requirements of the OMA conformance document for both 3GPP and 3GPP2 devices.

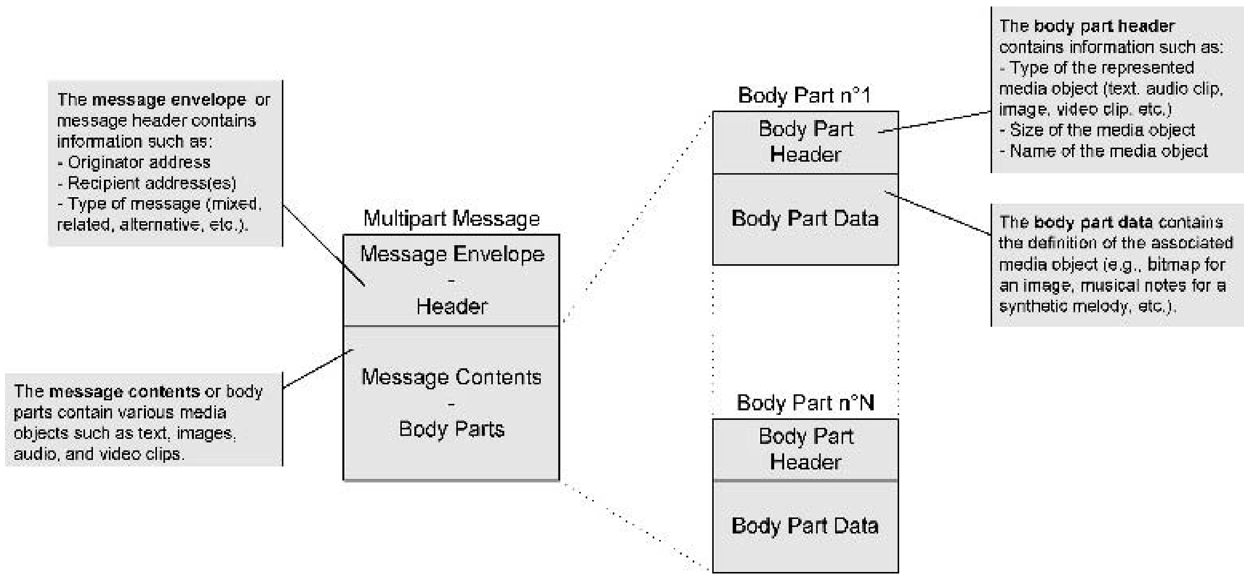


Figure 5.13 Structure of a multipart message

Table 5.8 Message body part header parameters

Parameter name	Description
`Content-ID`	A unique identifier for the body part in the multipart message [RFC-2392]. The identifier is typically inserted between square brackets. Example: `Content-ID: <ui_bp_0006>`
`Content-Location`	A user-readable name that is commonly used for naming the media object contained in the body part [RFC-2557]. This is particularly useful when the user wishes to extract the media object from the message (image to be used as wallpaper, audio clip to be used as ring tone, etc.). Example: `Content-Location: house.jpg`

`Content-Disposition`	An indication on how the media object should be displayed [RFC-1806]. Values can be `INLINE` for an inline display of the corresponding object or `ATTACHMENT` for a display in the form of an attachment. However, this parameter is seldom used in the context of MMS. Another technique is used in MMS for differentiating inline message objects from message attachments (see Section 5.29.8).
	
	Optionally, this parameter can be associated with a filename sub-parameter for naming the corresponding media object. Example:
	
	`Content-Disposition: INLINE;`
	`Filename = house.jpg`

`Content-Type`	The content type of the corresponding message object. A list of well known content-types for MMS is provided in Appendix E. Examples:
	
	`Content-type: image/jpeg`
	
	`Content-type: audio/midi`
	
	Optionally, certain content types can be associated with one or more parameters such as the character set or a name/filename. Examples:
	
	`Content-type: text/plain;`
	`charset = US-ASCII`
	
	`Content-type: image/GIF;`
	`name = house.jpg`

5.27.1 Message Content Domains

Considering the full standardization picture, messages can be categorized into following four nested message content domains [OMA-MMS-Conf] (version 1.2):

- *Core message content domain*: this domain refers to messages containing media objects with formats/codecs identified in the OMA conformance document [OMA-MMS-Conf]. These messages are typically generated in the person-to-person messaging scenario. Devices allowing the composition of messages belonging to the core message domain are subject to a low interoperability risk. Messages belonging to the core message domain can further conform to one of the six hierarchical message content classes known as: *text*, *image basic*, *image rich*, *video basic*, *video rich*, and *megapixel*.
- *Content message content domain*: this domain extends the core message content domain by introducing two new message content classes mainly targetted at the support of the content-to-person messaging scenario. The two new message content classes are known as: *content basic* and *content rich*.
- *Standard message content domain*: this domain groups messages containing media objects with formats/codecs identified by 3GPP in [3GPP-26.140] and 3GPP2 in [3GPP2-C.P0045]. Regarding the number of formats/codecs in [3GPP-26.140 ; 3GPP2-C.P0045] and the optional nature of most of them, devices allowing the composition of messages belonging to the standard domain, without restrictions other than using formats/codecs identified in [3GPP-26.140 ; 3GPP2-C.P0045], are subject to a medium interoperability risk.
- *Unclassified message content domain*: this domain basically includes all messages composed of media objects in the unlimited set of available formats/codecs. Devices allowing the composition of messages belonging to the unclassified domain, without restrictions regarding the set of usable formats/codecs, are subject to interoperability issues at a high-risk level.

5.27.2 Message Content Classes

In the MMS conformance document from version 1.2, the concept of message content classes[1] in the core and content message domains is introduced with the objective to guarantee interoperability between conformant MMS clients. The definition for each class identifies the maximum size for a message compliant to that class, a set of allowed formats and codecs and which OMA DRM mechanisms are applicable, if any. MMS clients that comply with the conformance document version 1.2 do support the requirements of two or more content classes in terms of message creation, submission, retrieval, and/or presentation. Conformant MMS clients are all expected to support the text class, which is the simplest of the eight defined content classes. In addition, a conformant MMS client also supports at least another content class defined in the core message domain. MMS client conformant to the version 2.0.0 (prior to version 1.2) of the MMS conformance document do support the image basic content class. Three additional classes have been introduced in version 1.2 of the MMS conformance document: image rich, video basic, and video rich (core content domain). In

[1] MMS introduces two concepts of classes: message classes (advertisement, personal, informational, auto) and message content classes (as described in this section). They are two distinct concepts.

addition, three more classes have been introduced in version 1.3: megapixel (core message domain), and content basic and content rich (content message domain). For the definition of these content classes, codecs and formats have been categorized into 10 media types as shown in Table 5.9 for 3GPP devices and Table 5.10 for 3GPP2 devices. This categorization was initially introduced in [3GPP-26.140] and is used in the subsequent sections of this book.

A multimedia message is conformant to a given content class if the following conditions are fulfilled:

- All media objects contained in the message are of a media format/codec allowed for the given class. In addition, media objects must fulfil the format requirements in terms of image resolutions, size, and so on.
- The message fulfills the requirements of the message class definition in terms of maximum message size allowed (30, 100, or 300 KB) and applicability of the Digital Rights Management (DRM) mechanism (e.g., OMA DRM forward-lock cannot be applied to messages expected to conform to the requirements of the image basic class).

The MMSC can perform content adaptation to a multimedia message conformant to one of the classes, so it becomes conformant to another class better handled by the receiving device.

5.27.3 MMS Client Functional Conformance

MMS clients can be characterized according to their support in terms of message creation, submission, retrieval, and presentation (i.e., rendering) of messages compliant to the eight message content classes from the core and content message domains.

- *Creation conformance*: an MMS client is said to be creation conformant towards a given class (text, image basic, image rich, video basic, video rich, megapixel, content basic, or content rich) if the client allows the insertion of any media formats/codecs allowed for the given message content class and if, of course, messages created by the MMS client are conformant to the given message class. A creation conformant MMS client typically warns the user if the created message diverges from the requirements of the given content class.
- *Submission conformance*: an MMS client is said to be submission conformant towards a given class if the client is able to submit any multimedia message compliant to this given class. A submission conformant MMS client typically warns the user if the submitted message diverges from the requirements of the given content class.
- *Retrieval conformance*: an MMS client is said to be retrieval conformant towards a given class if the client is able to retrieve any multimedia message belonging to this given class.
- *Presentation conformance*: an MMS client is said to be presentation conformant towards a given class if the client is able to render all media objects contained in any multimedia message belonging to this given class.

An MMS client is said to be *fully conformant* to a given content class if it is creation, submission, retrieval, and presentation conformant to the given content class. An MMS client is said to be *partially conformant* to a given content class if it is compliant in creation,

Table 5.9 Message content classes for 3GPP devices

Content domains	Core message content domain						Content core content domain	
Content classes	Class text	Class image basic	Class image rich	Class video basic	Class video rich	Class megapixel	Class content basic	Class content rich
Text	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16
Still image	None	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG
Bitmap image	None	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP	GIF87a, GIF89a, WBMP
Vector graphics	None	None	None	None	None	None	None	SVG Tiny
Speech	None	AMR narrowband	AMR narrowband	AMR narrowband	AMR narrowband	AMR narrowband	AMR narrowband	AMR narrowband
(Music) Audio	None	None	None	None	None	None	None	MPEG4 AAC
Synthetic audio	None	None	SP-MIDI	SP-MIDI	SP-MIDI	SP-MIDI	SP-MIDI	SP-MIDI
Video	None	None	None	H.263 with AMR-NB (.3GP)	H.263 with AMR-NB (.3GP)	None	H.263 with AMR-NB (.3GP)	H.263 with AMR-NB (.3GP)
Personal Information Manager	None	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar
Scene description	MMS SMIL	MMS SMIL	MMS SMIL	MMS SMIL	MMS SMIL	MMS SMIL	PSS SMIL	PSS SMIL
DRM support	No	No	Forward-lock	Forward-lock	Forward-lock	Forward-lock	Forward-lock Separate delivery Combined delivery	Forward-lock Separate delivery Combined delivery
Max image resolution	Not applicable	160 × 120	640 × 480	640 × 480	640 × 480	1600 × 1200	640 × 480	1600 × 1200
Message size	≤30 KB	≤30 KB	≤100 KB	≤100 KB	≤300 KB	≤300 KB	≤100 KB	≤300 KB

Table 5.10 Message content classes for 3GPP2 devices

Content domains	Core message content domain						Content core content domain	
Content classes	Class text	Class image basic	Class image rich	Class video basic	Class video rich	Class megapixel	Class content basic	Class content rich
Text	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16	US-ASCII, UTF-8, UTF-16
Still image	None	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG	Baseline JPEG
Bitmap image	None	GIF87a, GIF89a, WBMP, PNG	GIF87a, GIF89a, WBMP, PNG	GIF87a, GIF89a, WBMP, PNG	GIF87a, GIF89a, WBMP, PNG	GIF87a, GIF89a, WBMP, PNG	GIF87a, GIF89a, WBMP, PNG	GIF87a, GIF89a, WBMP, PNG
Vector graphics	None	None	None	None	None	None	None	SVG Tiny
Speech	None	AMR narrowband or 13K	AMR narrowband or 13K	AMR narrowband or 13K	AMR narrowband or 13K	AMR narrowband or 13K	AMR narrowband or 13K	AMR narrowband or 13K
(Music) Audio	None	None	None	None	None	None	None	MPEG4 AAC
Synthetic audio	None	None	SP-MIDI, General MIDI level 1	SP-MIDI, General MIDI level 1	SP-MIDI, General MIDI level 1	SP-MIDI, General MIDI level 1	SP-MIDI, General MIDI level 1	SP-MIDI, General MIDI level 1
Video	None	None	None	- H.263 and MPEG4 for video - 13K or AMR-NB for audio (.3GP2)	- H.263 and MPEG4 for video - 13K or AMR-NB for audio (.3GP2)	- H.263 and MPEG4 for video - 13K or AMR-NB for audio (.3GP2)	- H.263 and MPEG4 for video - 13K or AMR-NB for audio (.3GP2)	- H.263 and MPEG4 for video - 13K or AMR-NB for audio (.3GP2)
Personal Information Manager	None	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar	vCard and vCalendar
Scene description	MMS SMIL	MMS SMIL	MMS SMIL	MMS SMIL	MMS SMIL	MMS SMIL	PSS SMIL	PSS SMIL
DRM support	No	No	Forward-lock	Forward-lock	Forward-lock	Forward-lock	Forward-lock Separate delivery Combined delivery	Forward-lock Separate delivery Combined delivery
Max image resolution	Not applicable	160 × 120	640 × 480	640 × 480	640 × 480	1600 × 1200	640 × 480	1600 × 1200
Message size	≤30 KB	≤30 KB	≤100 KB	≤100 KB	≤300 KB	≤300 KB	≤100 KB	≤300 KB

submission, retrieval, or presentation but not conformant to all four aspects. For instance, an observation camera, which is conformant to the image basic class in creation and submission only is said to be partially conformant to the image basic class.

Content classes in the content domain are applicable to the content-to-person scenario. MMS clients are not expected to be creation or submission conformant to these classes (content basic and content rich classes).

5.27.4 Creation Modes

MMS clients may support three distinct message creation modes: restricted, warning, and free. With the *restricted mode*, the MMS client does not allow the creation of messages that do not belong to the message content class it is compliant to. With the *warning mode*, the MMS client allows the creation of messages that do not belong to the message content class it is compliant to. However, in this mode, a warning is given to the user upon creation of such messages. With the *free mode*, the creation of messages not belonging to the message class the MMS client is compliant to is allowed without warning given to the user. These modes may be pre-configured, configured by the user, or configured by the operator via a device management mechanism (see Section 5.17.3).

5.28 Media Types, Formats, and Codecs

This section presents the different media types supported in the context of MMS. MMS devices may support one or more media types (still image, video, speech, etc.). And for each supported media type, the MMS device supports at least one media format/codec (e.g., AMR for speech, H.263 for video, GIF and WBMP for bitmap images, etc.).

Appendix E provides a list of corresponding format/codec content types.

5.28.1 Text

Like other media objects, text in multimedia messages is contained in message body parts. Each text body part is characterized with a content type identifying one of the available character sets. The Internet Assigned Numbers Authority (IANA) publishes, in [IANA-MIBEnum], the list of character sets along with unique character set identifiers known as *MIBEnum*. Regardless of the message content class, [OMA-MMSConf] (version 1.2) mandates the support of the three character sets listed in Table 5.11.

Table 5.11 Character sets

Character set	MIBEnum
US-ASCII	3
UTF-8	106
UTF-16[1]	1015

[1] [OMA-MMA-Conf] (version 2.0) indicates that MIBEnum for UTF-16 is 1000. This is an error since MIBEnum 1000 identifies UCS-2. This has been the source of interoperability problems for initial MMS solutions. This error is corrected from version 1.2 of the same document.

MMS clients support the UTF-16 character set for backward compatibility with first commercial implementations. However, for interoperability reasons, it is recommended not to use UTF-16 for encoding text in multimedia messages.

5.28.2 Bitmap and Still Images

In the context of MMS, several formats can be used for representing images in multimedia messages. *Bitmap images* are expected to become widely used in MMS because of the large availability of such image formats over the Internet. Bitmap images are exact pixel-by-pixel mappings of the represented image (‘‘exact’’ only if no lossy compression is used to reduce the bitmap size). Common formats for bitmap images are GIF, the Portable Network Graphic (PNG) format and the Wireless BitMaP (WBMP) format [WAP-237]. PNG is a file format whose definition is published in the form of a World Wide Web Consortium (W3C) recommendation [W3C-PNG]. PNG is used for representing bitmap images in a lossless fashion. PNG ensures a good compression ratio and supports image transparency. OMA mandates the support of GIF (GIF87a and GIF89a) and WBMP for 3GPP/3GPP2 devices [OMA-MMS-Conf] for all message content classes, except for the text class. In addition, it mandates the support of PNG for 3GPP2 devices for all message content classes, except for the text class.

Box 5.3 Web resources for images

PNG at W3C: http://www.w3c.org/Graphics/PNG/

On the other hand, OMA mandates the support of JPEG with the JFIF[1] file format for *still images* [OMA-MMS-Conf] for all message content classes, except for the text class. In addition, MMS clients can include additional information (camera make, model, firmware revision, photographic settings such as exposure time, aperture, F-stop, etc.) as part of the JPEG file by using the EXIF[2] file format. With EXIF, such information from the phone digital camera can be inserted as part of the multimedia message for later use by third parties such as photo finishers to improve the printing of photo hardcopies.

For the composition of messages, the user has access to images stored locally in the handset (e.g., photo album). In addition, the MMS client has often access to photos taken with a digital camera (built into the phone or as an external accessory). Digital cameras allow the capture of pictures according to various resolutions modes. It is very common to refer to VGA[3] display modes when specifying the capture resolution of a camera. Image resolution modes shown in Table 5.12 are commonly supported by MMS-enabled phones (resolution is expressed in number of horizontal pixels × number of vertical pixels) in Table 5.12.

In addition from supported capture and display resolutions, an MMS phone is also characterized by the color depth of its display screen and of its digital camera capture

[1] JPEG File Interchange Format.
[2] Exchangeable Image File Format for Digital Still Camera.
[3] Video Graphics Array (VGA) was introduced in 1987 by IBM and has become the accepted minimum standard for PC display systems.

Table 5.12 Image resolutions

Display mode	Resolution
VGA	640 × 480
QVGA (stands for Quarter VGA)	320 × 240
QQVGA	160 × 120
QQQVGA	80 × 60

capabilities. Standardization organizations do not make any recommendation regarding the minimum color depth to be supported by MMS devices. However, one can observe that the lowest supported color depth for the display screen of available devices is 8 bits (256 colors). The majority of early MMS phones did support a minimum color depth of 12 bits (4096 colors) and now many MMS devices already support a color depth of 16 bits (65,536 colors) and even more for more advanced devices.

5.28.3 Vector Graphics

Vector graphics are based on descriptions of graphical elements composing the represented synthetic image/animation. These descriptions are usually made using instructions similar to those of a programming language. Vector graphics instructions are processed by a graphics processor to reconstruct graphical elements contained in the represented image/animation. Metafile and the emerging SVG are two well-known vector graphic formats used for representing images. SVG is an open standard based on XML and published as a W3C recommendation [W3C-SVG]. Advantages of SVG include the possibility of dynamically scaling the represented image according to the capabilities of the receiving device (e.g., screen size, frame rate). Furthermore, the size of SVG representing synthetic images/animations can be very low compared to the size of equivalent bitmap/still image and video representations.

However, scalable graphic formats are not appropriate for representing all types of images/animations. For instance, photographs and recorded video clips are usually not well represented with vector graphics. Representing photographs with a scalable vector graphic format may lead to very large representations, larger than equivalent bitmap/still image representations. Additional processing capabilities are usually required to render vector graphics (integer and floating point calculations).

In the context of MMS, SVG is the most suitable vector graphics format and OMA mandates the support for *SVG tiny* profile for devices compliant to the content rich content class [OMA-MMS-Conf] (version 1.3).

Box 5.4 Web resources for vector graphics

SVG at W3C: http://www.w3c.org/Graphics/SVG/

5.28.4 Audio

Several audio codecs are used in the context of MMS. Audio codecs are usually classified using three media types as defined below:

- *Speech audio* codecs used to represent speech samples such as voice memos;
- *Audio codecs* used to represent audio clips including recorded music teasers;
- *Synthetic audio* formats consist of specifying commands instructing a synthesizer on how to render sounds. Such formats are used for the representation of melodies.

5.28.4.1 Speech

For the *speech audio* media type, Adaptive Multi-Rate (AMR) has long been the codec of choice. The AMR codec is typically used to represent speech in mobile networks for voice communications. This codec can adapt its rate according to available resources. AMR compresses linear-PCM speech input at a sample rate of 8 kHz adaptively to one of eight data rate modes: 4.75, 5.5, 5.9, 6.7, 7.4, 7.95, 10.2, and 12.2 Kbps. Note that AMR, when configured at given rates, becomes directly compatible with technical characteristics of other codecs specified by several standardization organizations. For instance, AMR, at the 12.2 Kbps rate, is compatible with the Enhanced Full Rate (EFR) speech codec defined by ETSI for GSM. Furthermore, AMR, at the 7.4 Kbps rate, becomes compatible with the IS-136 codec. Finally, AMR, configured at the 6.7 Kbps rate, becomes compatible with the speech codec used in the Japanese PDC.

This initial specification of the AMR codec is also referred to as the AMR narrowband (AMR-NB). AMR-NB is suitable for representing recorded speech and does not have the capabilities of representing adequately recorded music teasers. In the context of MMS, AMR data is stored and transported in multimedia messages according to the file format specified by IETF in [RFC-3267].

An extension of AMR-NB, known as AMR-wideband (AMR-WB), can be used for the representation of music clips, in addition to representing voice with a better quality than AMR-NB. AMR-WB compresses linear-PCM speech/music input at a sample rate of 16 kHz adaptively to a multitude of data rate modes from 6.6 to 23.85 Kbps.

Another speech codec called 13K has been adopted for 3GPP2 markets and can be used by MMS clients embedded in 3GPP2 devices.

OMA mandates the support of AMR-NB for speech for 3GPP devices for all message content classes [OMA-MMS-Conf], except for the text class. In addition, OMA specifies that for 3GPP2 devices all message content classes, except the text class, do support either the AMR-NB codec or the 13K codec (manufacturer implementation choice).

5.28.4.2 Audio and Synthetic Audio

MPEG-4 AAC low complexity codec appears to be the most appropriate *audio codec* for the representation of music teasers in the context of MMS. Two other candidates were AMR-WB and MP3. MP3 stands for MPEG Layer-3 and is an audio-compressed format. MP3 is based on perceptual coding techniques that address the perception of sound waves by the human ear. OMA has selected MPEG-4 as the codec for the content rich content class for both 3GPP and 3GPP2 devices [OMA-MMS-Conf] (version 1.3).

For the *synthetic audio* media type, MIDI and iMelody formats have been used in the context of MMS. Since the release of MIDI 1.0 in 1982 [MMA-MIDI], MIDI has become the most widely used synthetic music standard among musicians and composers. The MIDI standard encompasses not only the specifications of the connector and cable for interconnecting MIDI-capable devices but also the format of messages exchanged between these devices. Only the format of MIDI messages is of interest in the context of MMS.

Melodies in the MIDI format are represented by a sequence of instructions that a sound synthesizer can interpret and execute. In an MMS device, this MIDI sound synthesizer may be implemented either as a software-based synthesizer or as a hardware MIDI chipset. Instructions rendered by the sound synthesizer are in the form of MIDI messages. For instance, a MIDI message can instruct the synthesizer to use a specific instrument or to play a given note. In theory, MIDI messages can be sent in real time to the synthesizer.

In the scope of MMS, the MIDI melody is first conveyed as part of a message and later rendered by the sound synthesizer when requested by the user. For this purpose, additional timing information needs to be associated with MIDI messages in order to tell the synthesizer when to play the melody notes. To achieve this, timing information along with MIDI messages are formatted as Standard MIDI Files (SMF). MIDI and SMF are also used in the scope of extended EMS and are further described in Section 4.4.16 of this book.

Note that the iMelody format is not a format suggested for MMS by standardization organizations. Nevertheless, several MMS-capable devices available on the market do support iMelody tunes in multimedia messages.

OMA mandates the support of SP-MIDI for all content classes, except for the text and image, basic content classes for 3GPP/3GPP2 devices. In addition, OMA mandates the support of General MIDI level 1 for all content classes, except for the text and image, for 3GPP2 devices [OMA-MMS-Conf] (version 1.3).

5.28.5 Video

First MMS devices did not support video clips in multimedia messages. At present, several MMS devices do have support for video and this trend is growing.

The International Telecommunication Union (ITU) has published several video standards including H.261 (late 1980/early 1990s), H.263 (mid-1990s), and more recently H.26L. In the context of MMS, messages conformant to the video/content basic or video/content rich class may contain video clips encoded according to H.263 profile 0 level 10. The maximum size of video clips is limited by the maximum message size inherent to each message content class: 100 KB for video basic and 300 KB for video/content rich.

MPEG stands for Moving Pictures Expert Group and is an organization that develops technical specifications for representing and transporting video. The first specification from this organization was MPEG-1, published in 1992. MPEG-1 allows video players to render video in streaming mode. MPEG-2, introduced in 1995, supersedes MPEG-1 features and is mainly used for compression and transmission of digital television signals. In December 1999, the group released the specification for MPEG-4 (ISO/IEC 14496), based on an object-oriented paradigm, where objects are organized in order to compose a synthetic scene. This is a major evolution from previous MPEG formats. The compact MPEG-4 language used for describing and dynamically changing scenes is known as the Binary Format for Scenes (BIFS). BIFS commands instruct the MPEG-4 player to add, remove, and dynamically

Table 5.13 Video frame resolutions

Display mode	Resolution
4CIF	704×546
CIF	352×288
QCIF (stands for Quarter CIF)	176×144
Sub-QCIF	128×96

change scene objects in order to form a complete video presentation. Such a technique allows the representation of video scenes in a very compact manner. MPEG-4 also supports streaming in low bit rate environments. MPEG-4 has proved to provide acceptable streaming services over 10 Kbps channels. Mobile networks often provide very variable and unpredictable levels of resources for services. To cope with these network characteristics, MPEG-4 can prioritize objects to transmit the most important objects only when the system is running short of resources. Owing to the limitations of available mobile devices, 3GPP recommends the use of the visual simple profile level 0 of MPEG-4: the simplest of available profiles and levels for MPEG-4 (see Box 5.5).

Box 5.5 MPEG video codecs
Web resources for MPEG

MPEG-4 Industry Forum: http://www.m4if.org

An MMS device supporting video is typically capable of displaying a digital video in one of the video frame resolutions listed in Table 5.13. A mobile device supporting H.263 profile 0 level 10 is able to support at least the Sub-QCIF and the QCIF video resolutions.

For 3GPP devices, OMA mandates the support of [ITU-H.263] for video basic, video rich, megapixel, and content rich content classes [OMA-MMS-Conf] (version 1.3). If the video clip also includes an audio track then it shall be represented as an AMR-NB clip.

For 3GPP2 devices, OMA mandates the support of [ITU-H.263] and MPEG-4 for video basic, video rich, megapixel, content basic, and content rich content classes [OMA-MMS-Conf] (version 1.3). If the video clip also includes an audio track then it shall be represented as an AMR-NB clip or alternatively as a 13K clip.

Specific file formats are used for timed multimedia (e.g., video with audio) in the context of MMS and are known as the *3GPP file format* (3GP) for 3GPP devices [3GPP-26.234] and the *3GPP2 file format* for 3GPP2 devices [3GPP2-C.P0045].

5.28.6 Personal Information Manager Objects

If a Personal Information Manager (PIM) is present in the MMS mobile device, then electronic business cards and calendaring/scheduling information may be exchanged with MMS. PIM objects are represented with vCard and vCalendar formats.

The *vCard format* is used for representing electronic business cards [IMC-vCard]. This format is already widely used with Personal Digital Assistants (PDAs) and is becoming the de facto format for the exchange of electronic business cards over infrared links. It is also becoming common to attach a vCard object, as a signature, to an Email message. A vCard object contains basic contact details such as last name, first name, postal and electronic addresses, phone, mobile and fax numbers, and so on. It may also contain more sophisticated elements such as photographs or company logos. The vCard format is further described in Section 4.4.14.

On the other hand, the *vCalendar format* is used to represent items generated by calendaring and scheduling applications [IMC-vCalendar]. As for the vCard format, it is widely used with PDAs and is becoming the de facto format for the exchange of calendaring/ scheduling information. A vCalendar object is composed of one or more elements of types event and todo. An *event* is a calendaring/scheduling element representing an item in a calendar. A *todo* is a calendaring/scheduling element representing an action item or assignment. The vCalendar format is further described in Section 4.4.15.

5.29 Scene Description

Previous sections have described how several media objects can be included in a multimedia message. In order to enrich the user experience, it is common to organize the way media objects are rendered on the receiving device screen and when they should be rendered over a common timeline. This allows the creation of truly multimedia presentations in which media objects are choreographed in a meaningful manner on the receiving device. This media object organization, also called scene description, is defined using a format/language such as the Synchronized Multimedia Integration Language (SMIL) or XHTML. The minimal subset of SMIL defined in [OMA-MMS-Conf], also known as MMS SMIL, became the de facto scene description representation for available MMS devices and is now published as part of OMA MMS standards. In the meantime, 3GPP has elaborated a more sophisticated SMIL profile for MMS.

5.29.1 Introduction to SMIL

The *Synchronized Multimedia Integration Language* (SMIL), pronounced "smile," is an XML-based language published by W3C. A major version of this language, SMIL 2.0 [W3C-SMIL], is organized around a set of modules defining the semantics and syntax of multimedia presentations (for instance, modules are available for the timing and synchronization, layout and animation, etc.). SMIL is not a codec nor a media format but rather a technology allowing media integration. With SMIL, the rendering of a set of media objects can be synchronized over time and organized dynamically over a predefined graphical layout to form a complete multimedia presentation. SMIL is already supported by a number of commercial tools available for personal computers including RealPlayer, Quicktime, and Internet Explorer.

Owing to small device limitations, a subset of SMIL 2.0 features has been identified by W3C to be supported by devices such as PDAs. This subset, called *SMIL basic profile*, allows small appliances to implement some of the most useful SMIL features without having to support the whole set of SMIL 2.0 instructions. Unfortunately, SMIL basic profile appeared to be still difficult to implement on MMS mobile devices. To cope with this difficulty, a group of manufacturers designed an even more limited SMIL profile, known as the *MMS SMIL*, supported by most MMS phones that support scene descriptions. The Open Mobile Alliance is now the organization in charge of maintaining and publishing the MMS SMIL specification as part of [OMA-MMS-Conf] (available from MMS 1.1).

In the meantime, 3GPP has produced specifications for a *3GPP SMIL profile*, also known as the packet-switched streaming SMIL profile (*PSS SMIL profile*). The 3GPP SMIL profile is gaining support for the content-to-person scenario whereas MMS SMIL is still sufficient for most person-to-person scenarios. The 3GPP SMIL profile is still a subset of SMIL 2.0 features, but a superset of the SMIL basic profile, and is published in [3GPP-26.234].

Designers of SMIL multimedia presentations can:

- describe the temporal behavior of the presentation
- describe the layout of the presentation on a screen
- associate hyperlinks with media objects
- define conditional content inclusion/exclusion on the basis of system/network properties.

Box 5.6 Resources for SMIL

A comprehensive two-part tutorial on SMIL is identified in the further reading section of this chapter (Bulterman, 2001, 2002).

SMIL at W3C: http://www.w3c.org/AudioVideo

5.29.2 Organization of SMIL 2.0

A major version of the language, SMIL version 2.0, has been publicly released by W3C in August 2001. The 500-page SMIL 2.0 specifications define a collection of XML tags and attributes that are used to describe temporal and spatial coordination of one or more media objects that form a multimedia presentation. This collection is organized into 10 major functional groups as shown in Figure 5.14.

Each functional group is composed of several modules (from 2 to 20). The aim of this SMIL organization is to ease the integration of SMIL features into other XML-derived languages. A number of profiles have been defined on the basis of this organization. A SMIL profile is a collection of modules. So far, several profiles have been introduced such as the SMIL 2.0 language profile, XHTML SMIL profile, and the SMIL 2.0 basic profile (as introduced earlier).

5.29.3 Spatial Description with SMIL

SMIL 2.0 content designers are able to define sophisticated spatial layouts. The presentation rendering space is organized by *regions*. Each region in the layout can accommodate a graphical object such as an image, a video clip, or some text. Regions can be nested in each other in order to define advanced multimedia presentations. The tag `root-layout` defines the main region of the presentation. Sub-regions to be nested within the main region are defined with the `region` tag. The SMIL example in Figure 5.15 shows how two sub-regions, one accommodating an image and the other some text, can be defined within the main region.

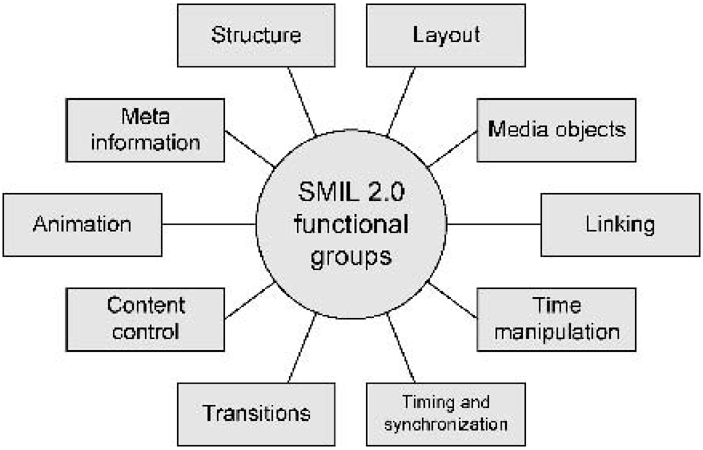


Figure 5.14 SMIL 2.0 functional groups

```
<layout>
        <root-layout width="128" height="128" />
        <region id="Image" width="128" height="72" left="0" top="0" />
        <region id="Text" width="128" height="56" left="0" top="72" />
</layout>
```

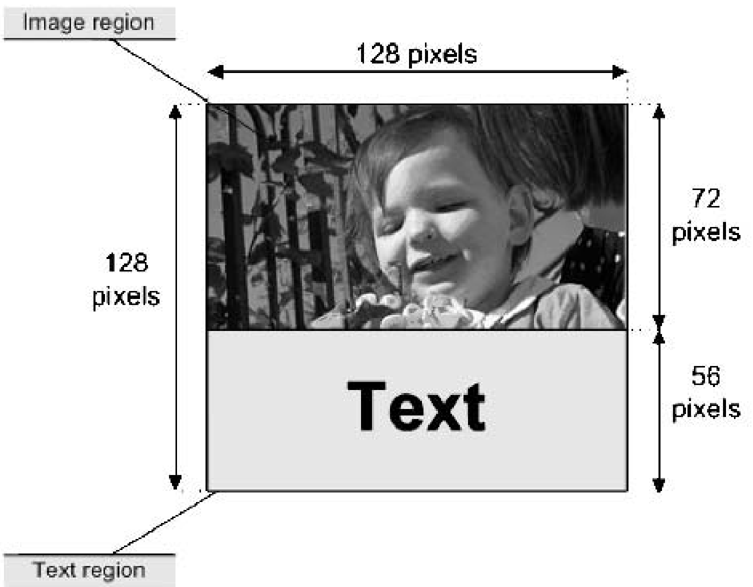


Figure 5.15 SMIL/layout container/smiling Louise after bed-time

5.29.4 Temporal Description with SMIL

Objects in a SMIL presentation can be synchronized over a common timeline. For this purpose, a set of time containers can be used such as the sequential and the parallel time containers:

- *Sequential time container*: this container, identified by the `seq` tag, enables the sequencing of an ordered list of objects. Each object is rendered in turn and the rendering of each object starts when the rendering of the previous object has terminated. An absolute time duration for the rendering of each object may also be specified as shown in Figure 5.16.

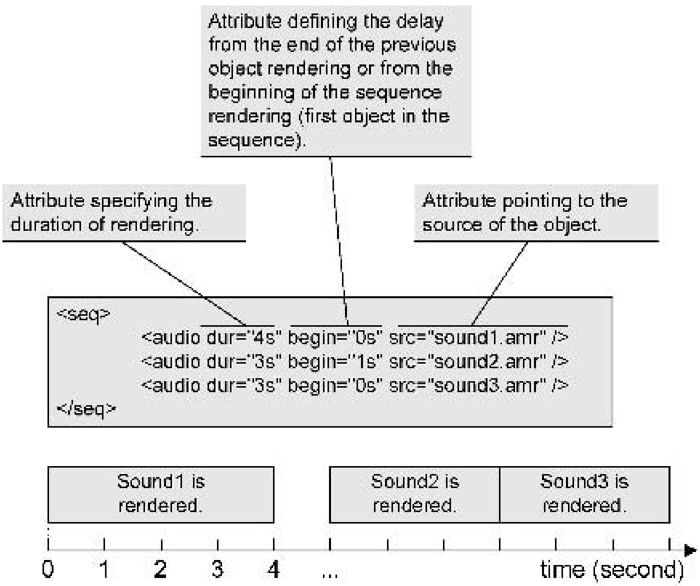


Figure 5.16 SMIL/sequential time container

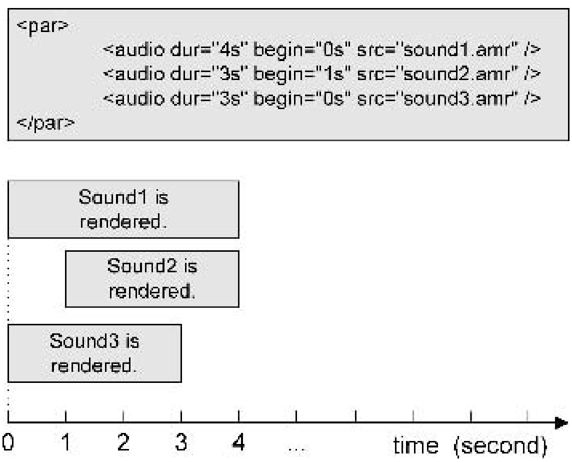


Figure 5.17 SMIL/parallel time container

- *Parallel time container*: this container, identified by the `par` tag, enables the rendering of several objects in parallel as shown in Figure 5.17.

In a scene description, containers can be nested in order to create a whole hierarchy of time containers for the definition of sophisticated multimedia presentations.

5.29.5 SMIL Basic Profile

As indicated in previous sections, W3C has defined a SMIL basic profile for SMIL 2.0. The SMIL basic profile is a subset of the full set of SMIL 2.0 features, appropriate for small appliances such as PDAs. In the short term, this profile does not appear to be suitable for mobile devices, which still have very limited capabilities. In order to cope with such limitations, MMS SMIL defined in the following section is used instead.

5.29.6 MMS SMIL and the OMA Conformance Document

With early MMS standards, manufacturers lacked an appropriate message scene description language: a language rich enough to allow the design of basic multimedia presentations but simple enough to be supported by devices with very limited capabilities. To fulfill this requirement, MMS-IOP, an informal working group of industrial partners, designed a profile for SMIL that fulfills the need of first MMS capable devices. The profile is known as *MMS SMIL* and was initially defined outside standardization processes in a document known as the *MMS conformance document*. In addition to the definition of MMS SMIL, the conformance document also provides a set of recommendations to ensure interoperability between MMS-capable devices produced by different manufacturers.

With MMS SMIL, a multimedia presentation is composed of a sequence of slides (a slideshow) as shown in Figure 5.18. All slides from the slideshow have the same region configuration. A region configuration can contain at most one text region (named `Text`), at most one image/video region (named `Image`), and each slide can be associated with at most one audio clip (speech or synthetic audio). The audio clip may be encapsulated into the video clip (e.g., 3GP file) or inserted in the message as an independent media object (e.g., AMR or SP-MIDI file). However, [OMA-MMSConf] (from version 1.2) forbids the inclusion of both

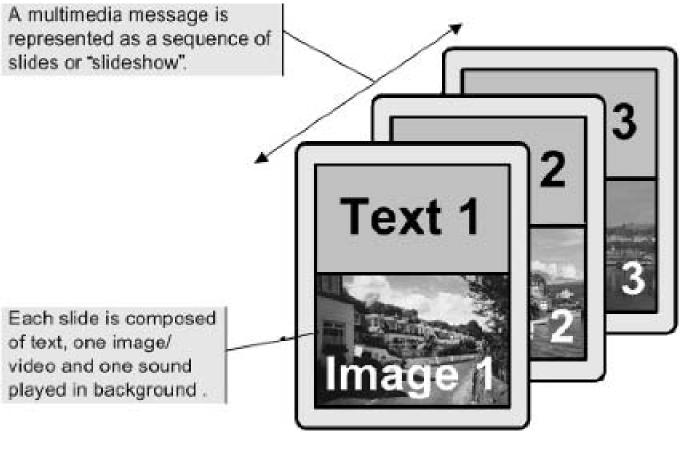


Figure 5.18 Message presentation with MMS SMIL

an independent audio clip and a video clip in the same slide, even if the video clip does not contain any audio track.

For slideshows containing both, a text region and an image region, the term *layout* refers to the way regions are spatially organized: vertically (portrait) or horizontally (landscape). Mobile devices have various screen sizes and it may therefore happen that a particular layout for a message scene description does not fit in the device display. In this situation, the layout may be overridden by the receiving device (from portrait to landscape or vice versa).

A message presentation may contain timing settings that allow an automatic slideshow rendering, during which the switching from one slide to the following is performed automatically after a time period defined as part of the message scene description. However, MMS phones often allow an interactive control by ignoring the timing settings and by allowing the user to manually switch from one slide to the following by simply pressing a key.

The MMS conformance document identifies SMIL 2.0 features that can be used for constructing the slideshow. A slideshow should have a sequence of one or more parallel time containers (instructions within the two tags `<par>`...`</par>`). Each parallel time container represents the definition of one slide. Each media object identified inside the parallel time container represents one component of a slide. Supported language elements and attributes in MMS SMIL are listed in Table 5.14. In addition to the definition of MMS SMIL, the MMS conformance document also contains the following rules:

- Dimensions of regions inside the root region should be expressed in absolute terms (i.e., pixels) or in percentage relative to the dimensions of the root region. It is not recommended to mix absolute and relative dimensions in the same scene description.

Table 5.14 Language elements/attributes in MMS SMIL

Functional groups	Elements	Attributes	Content model
Layout	`Layout`		`region, layout`
	`Region`	`left, top, height, width, fit, id`	
	`Root-layout`	`width, height`	
Media	`Text`	`src, region, alt, begin, end`	
	`Img`	`src, region, alt, begin, end`	
	`Audio`	`src, alt, begin, end`	
	`Video`	`src, region, alt, begin, end`	
	`Ref`	`src, region, alt, begin, end`	
Structure	`Smil`		`head, body`
	`Head`		`layout`
	`Body`		`par`
Timing and synchronization	`Par`	`dur`	`text, img, audio, ref`
Meta information	`Meta`	`name, content`	

- The `src` attribute must refer to a media type compliant with the associated element. For instance, the value associated with the `src` attribute of an `img` element must refer to an image, and nothing else.
- Timing information should be expressed in integer milliseconds.
- Maximum image dimensions are 160 × 120 pixels for messages belonging to the image basic class, 640 × 480 for messages belonging to image rich, video basic, video rich, and content basic content classes and 1600 × 1200 pixels for messages belonging to the megapixel and content rich content classes.

The MMS conformance document specifies that a slide may be composed of a text region only, of an image region only, or of two regions, one for the text named `Text` and another one named `Image`. The region named `Image` can accommodate either an image or a video clip. This means that a slide cannot contain both an image and a video clip.

Considering the two layout types (portrait and landscape) for messages containing both an image/video region and a text region, a slideshow can be formatted according to the six configurations shown in Figure 5.19. In its simplest form, a slideshow is configured for

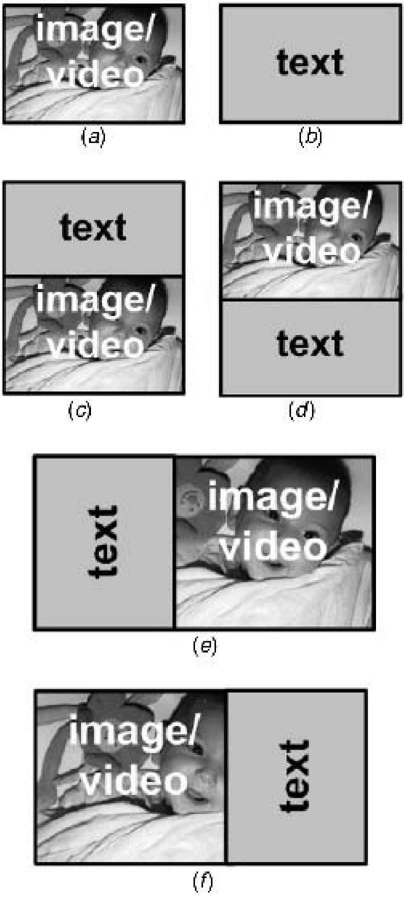


Figure 5.19 Scene description layouts

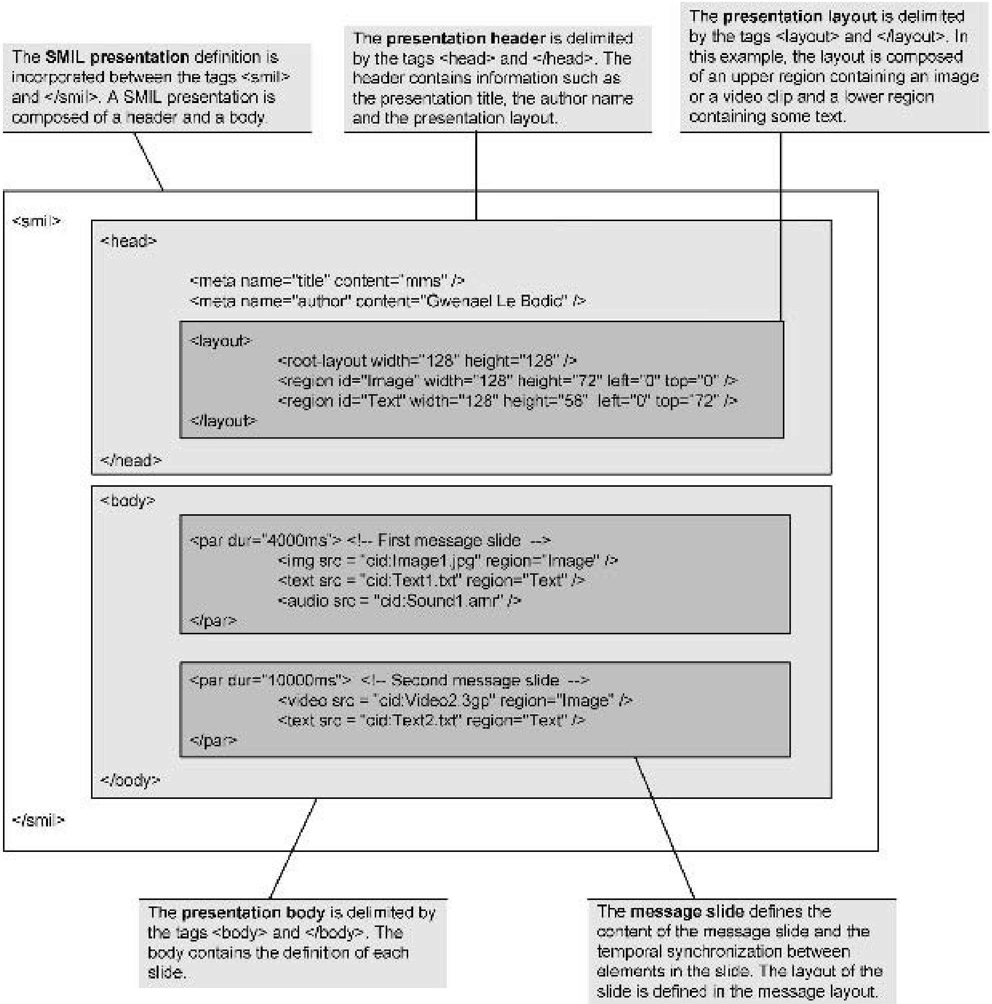


Figure 5.20 Example of SMIL scene description

representing only an image/video or only text on each slide. Such configurations are shown in Figure 5.19 (a) and (b). With the portrait layout, the image/video region may be positioned at the bottom or at the top of the screen. Examples of portrait layouts are depicted in Figure 5.19 (c) and (d). With the landscape layout, the image/video may be positioned at the right or at the left of the screen. Examples of landscape layouts are depicted in Figure 5.19 (e) and (f).

Figure 5.20 shows an example of MMS SMIL scene description corresponding to a two-slide configuration. The two slides are structured according to the portrait layout with the image/video region at the top and the text region at the bottom of the screen. In this example, the first slide contains an image (`Image1.jpg`), some text (`Text1.txt`), and an audio clip (sound1.amr), whereas the second slide contains a video clip (Video2.3gp) and some text (Text2.txt). The first slide is displayed during 4 seconds followed by the second slide for 10 seconds. The entire slide presentation time is therefore 14 seconds.

According to the MMS conformance document, if the multimedia message contains a scene description, then the Content-type header field of the multipart message header is set to application/vnd.wap.multipart.related; otherwise it is set to application/vnd.wap.multipart.mixed. Furthermore, if a presentation is available, the start parameter of the Content-type header field refers to the content identifier (Content-ID parameter) of the scene description present in the message and the type parameter indicates the format of the scene description (e.g., SMIL presentation) as shown below:

```
Content-type:application/vnd.wap.multipart.related;
                    start=<0000>;
              type='application/smil'
```

5.29.7 SMIL Namespace

An XML namespace identifies element types and attribute names that can be used in an XML document. In the context of MMS, a namespace can be specified for the SMIL scene description as the xmlns attribute of the smil element as shown below:

```
<smil xmlns = "http://www.w3.org/2000/SMIL20/CR/Language">
```

or

```
<smil xmlns = "http://www.w3.org/2001/SMIL20/Language">
```

The first example identifies the namespace for the SMIL 2.0 candidate release (commonly identified namespace with existing MMS clients) and the second example identifies the final W3C recommendation for SMIL 2.0.

MMS clients support a limited subset of available element types and attribute names. Some early MMS clients even ignore totally the SMIL description. For MMS clients that do support SMIL, the namespace is usually not specified as part of the SMIL scene description and the namespace identifier is commonly omitted as shown below:

```
<smil>
```

If the namespace attribute is omitted from a scene description, then a SMIL parser assumes that the scene description element types and attribute names are the ones defined in SMIL 1.0.

5.29.8 Linking the Scene Description with Body Parts

Like basic media objects, a scene description is encapsulated in a message body part. The scene description refers to message media objects with a unique object identifier in the message and instructs how the referred media object should be rendered by the MMS client. The two most common object identifiers used as references are the `Content-ID` and the `Content-Location`.

With MMS-SMIL, a scene description is structured as a sequence of slides. Each slide definition, in the scene description, refers to message media objects displayed inline on the slide (for images, text, and video clips) or rendered when the slide is displayed on the screen (for audio clips). Message media objects that are not referred to by the scene description are considered as *message attachments*. Message attachments are usually not displayed inline in message slides but the MMS client often provides features for storing attachments in the device local memory (e.g., internal flash memory or external flash card). This feature is particularly useful for exchanging PIM elements such as vCard or vCalendar objects.

5.29.8.1 Linking with Content Identifier

The most straight-forward method for referring to body parts in a scene description consists of using the content identifier (`Content-ID` parameter) as a unique identifier for a message body part. As shown in Section 5.26.2, each body part is associated with a content identifier in the following form:

```
Content-ID: <body_part_006>
```

In the example above, the value used as content identifier is `body_part_006` (square brackets are removed). In a SMIL scene description, a reference to a given body part using the content identifier is made according to the following instruction:

```
<img src = "cid:body_part_006" region = "Image">
```

The prefix "`cid`:" indicates that the method used for referring to the body part is the method based on content identifiers. In this example, the SMIL instruction `<img ... region = "Image">` indicates that the media object is an image to be inserted in the region named `Image`. Similarly, the SMIL instruction `<video ... region = "Image">` indicates that a video clip is inserted in the region named `Image`, whereas the instruction `<text ... region = "Text">` indicates that a text file is to be inserted in the region named `Text`. The instruction `<audio ... >` for including an audio clip in a slide makes no reference to a specific region.

Figure 5.20 shows a scene description whose references are made of content identifiers.

5.29.8.2 Linking with Content Location

Alternatively, the scene description can also refer to body parts using content locations. Each body part can be associated to a content location [RFC-2557] as shown below:

```
Content-location: Image1.jpg
```

In the example above, the value used as content location is `Image1.jpg`. In a SMIL scene description, a reference to a given body part using the `content-location` parameter is made according to the following instruction:

<img src = "Image1.jpg" region = "Image">

The fact that the prefix "`cid`:" is missing from the reference means that the scene description uses content locations for referring to body parts. As for the content identifier method, the SMIL instruction '<img ... region = "Image">' indicates in which region of the slide the referred media object should appear.

Note that the value assigned to the content location parameter can also be used by the message designer (or by the MMS client) to provide a default file name to the message media object in case the recipient wishes to store the object in the receiving device local memory (see Box 5.7).

Figure 5.21 below shows a complete scene description whose references are made of content locations.

Box 5.7 Recommendations for linking body parts and the scene description

With first MMS development, referencing methods based on `Content-ID` and `Content-Location` were often misused. In many cases, the `cid` prefix was missing from the references made of content identifiers. To cope with this issue, assigning the same value to the content identifier and content location reduces the probability to face interoperability problems with receiving MMS clients. However, this is not always possible since the content location is also used for providing a default file name in case the recipient wishes to store the associated media object in the receiving device local memory (e.g., photo album).

Mixing the two methods by using content location and content identifiers for referring to body parts, in the same scene description, is possible but not recommended.

5.29.9 Naming Body Parts

Upon receipt of a message, the user may wish to extract a media object from the message in order to reuse it later for the composition of new messages. In this case, the media object is stored in the local memory of the device (e.g., flash memory). The user may provide a name for this object to be able to retrieve it easily later in the local storage of the device. However, the device may also propose a name if the content author (e.g., the VAS provider) had provided one during message creation. In this case, the name of the object is specified as part of the message.

Several methods are available for naming media objects (i.e., naming message body parts). Methods are outlined below. The list below is sorted with recommended methods appearing at the top of the list (i.e., method based on `Content-Type` is preferred):

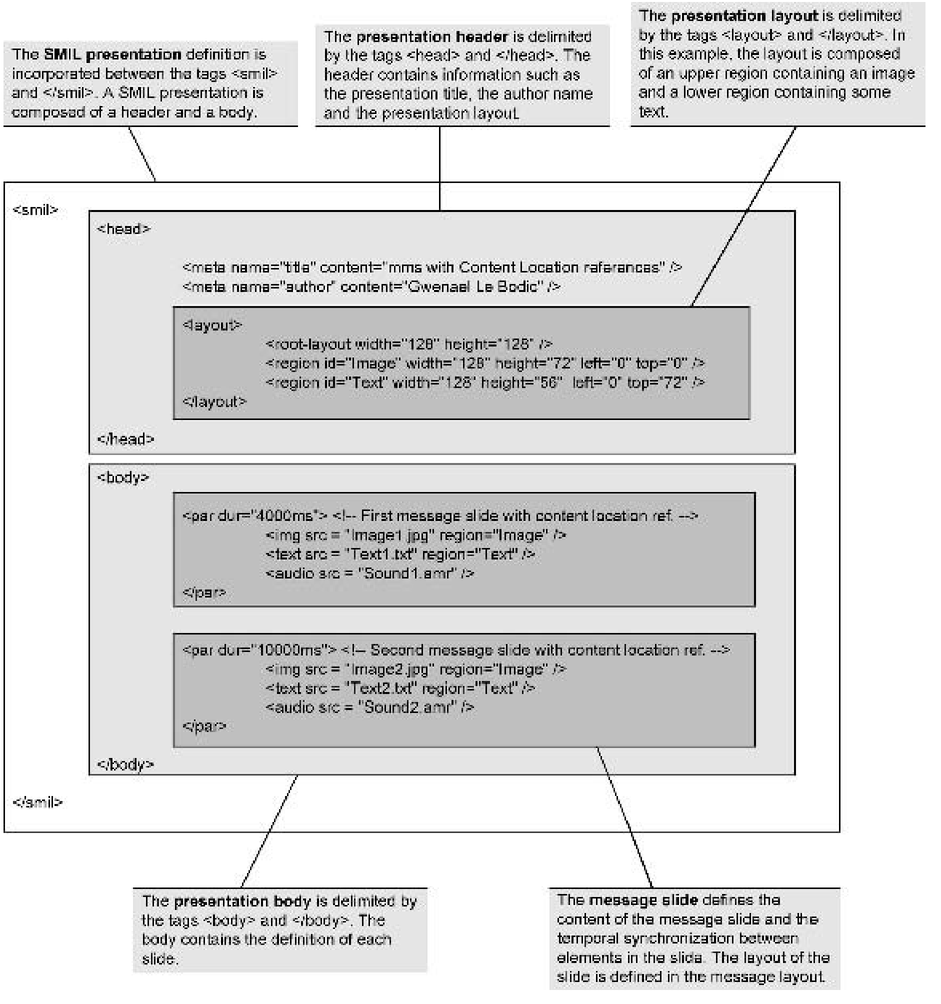


Figure 5.21 Scene description with content-location referencing

- Name of the object is assigned to the `name` sub-parameter of the `Content-type` parameter of the corresponding body part.
- Name of the object is assigned to the `filename` sub-parameter of the `Content-disposition` parameter of the corresponding body part.
- Name of the object is assigned to the `Content-location` parameter of the corresponding body part.

5.29.10 Support of Video Streaming

It has been shown in this chapter how a media object (e.g., video clip) can be inserted in a multimedia message and linked to a particular region of a SMIL scene description. An alternative method consists of indicating in the multimedia message a reference to a media object located in a remote media server. This method is typically used for referring to a video clip to be streamed from a media server down to the mobile device during message viewing and so reducing significantly the message size. Section 5.22 explained that the reference to the media object is provided as part of a presentation description (in the SDP format) or as a direct reference to the media object. In the latter case, the direct reference can be indicated as part of the SMIL scene description. The scene description below shows part of a scene description instructing the MMS client to retrieve a video clip from a media server in streaming mode using RTSP and to render this video clip in the region named `Image`:

```
<video src = ‘‘rtsp://www.lebodic.net/video/sample.3gp’’
region = “Image”>
```

5.29.11 Support of Color with SMIL

MMS SMIL does not provide support for slide colors (background color and text foreground color). However, SMIL 2.0 includes a set of element attributes for the support of slide colors. These attributes are optional in the scene description. A single slide background color can be specified for the entire message by assigning a color value to the `background-color` attribute of the `root-layout` tag as shown below:

```
<root-layout background-color = ‘‘yellow’’ width = ‘‘352’’
height = “144”/>
```

In addition, a text foreground color may be specified for each slide text object. This is performed by assigning a color attribute when linking the text object with the text region as shown below:

```
<text src = “Text1.txt” region = “Text”>
   <param name = “foreground-color” value = “blue”/>
</text>
```

It is common to use the following 16 color keywords defined in [W3C-HTML4]: `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white`, and `yellow`. In addition, RGB hexadecimal color codes [W3C-sRGB] may also be used (values are prefixed with the hash mark “#”) as shown below:

```
black   = “#000000”        green  = “#008000”
silver  = “#C0C0C0”        lime   = “#00FF00”
gray    = “#808080”        olive  = “#808000”
white   = “#FFFFFF”        yellow = “#FFFF00”
maroon  = “#800000”        navy   = “#000080”
red     = “#FF0000”        blue   = “#0000FF”
purple  = “#800080”        teal   = “#008080”
fuchsia = “#FF00FF”        aqua   = “#00FFFF”
```

The support of color is not specified in the MMS conformance document. Consequently, the support of text colors is not widely supported by MMS clients.

5.29.12 3GPP SMIL Profile or PSS SMIL

MMS SMIL is sufficient for most person-to-person scenarios. However, it is fairly limited for the support of content-to-person scenarios. OMA has therefore introduced the support of a 3GPP-defined SMIL profile also known as PSS SMIL from MMS version 1.3. It extends MMS SMIL by supporting hyperlinks, region nesting, etc.

5.29.13 XHTML as an Alternative to SMIL

As an alternative to SMIL, eXtensible HTML (XHTML) is a language that can also be used for representing message scene descriptions. In particular, XHTML Mobile Profile (XHTML MP) [WAP-277] extends HTML Basic Profile published by W3C [W3C-XHTML-Basic]. HTML MP is a subset of HTML 1.1 but a superset of HTML Basic Profile. XHTML MP has been specifically tailored for resource constrained devices. However, HTML MP remains a suitable language for the definition of rich MMS scene descriptions.

The OMA conformance document [OMA-MMS-Conf] (version 1.3) does not identify XHTML as an alternative to SMIL for MMS clients. However, OMA identifies that XHTML MP may be used for formatting the text contained in multimedia messages. For this purpose, all MMS clients compliant to the content core content domain do support the rendering of XHTML (when referened from a SMIL scene description).

5.30 Example of a Multimedia Message

As shown previously, the multipart structure of a multimedia message is represented in a binary form in order to be efficiently transported between the MMS client and the MMSC. This binary representation is directly derived from the MIME concepts introduced in Section 5.26. Figure 5.22 shows the textual representation of a multimedia message composed of two slides.[1] Each slide contains one image, some text, and one sound clip. The size of the entire MMS PDU is 25 KB.

Figure 5.23 shows how the message may look like when displayed on the receiving device.

5.31 DRM Protection of Media Objects

In the content-to-person scenario, certain business models require the capability to control the usage and the redistribution of protected contents. For this purpose, OMA has introduced three digital rights management mechanisms as introduced in Section 1.6.9: forward-lock, combined delivery, and separate delivery. These mechanisms can be used in the context of MMS as described in the next sections.

An MMS device indicates as part of its user agent profile whether or not it supports DRM mechanisms.

5.31.1 Forward-Lock

Media objects that needs to be protected against forwarding can simply be encapsulated into OMA DRM forward-lock envelopes. A forward-lock envelope is organized as a

[1] The binary representation of the message is available from this book's companion website at http://www.lebodic.net/mms_resources.htm.

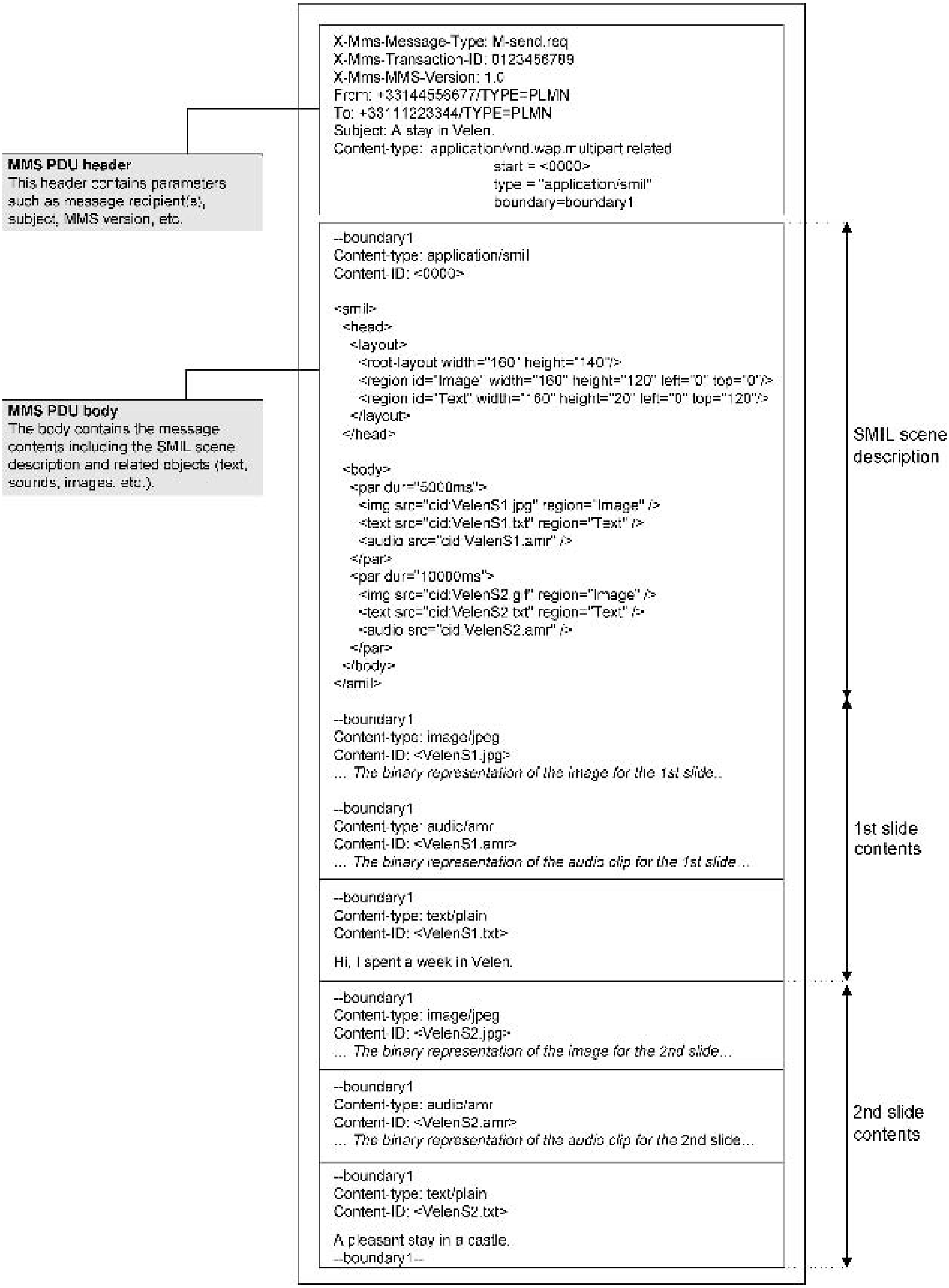


Figure 5.22 Full multimedia message example

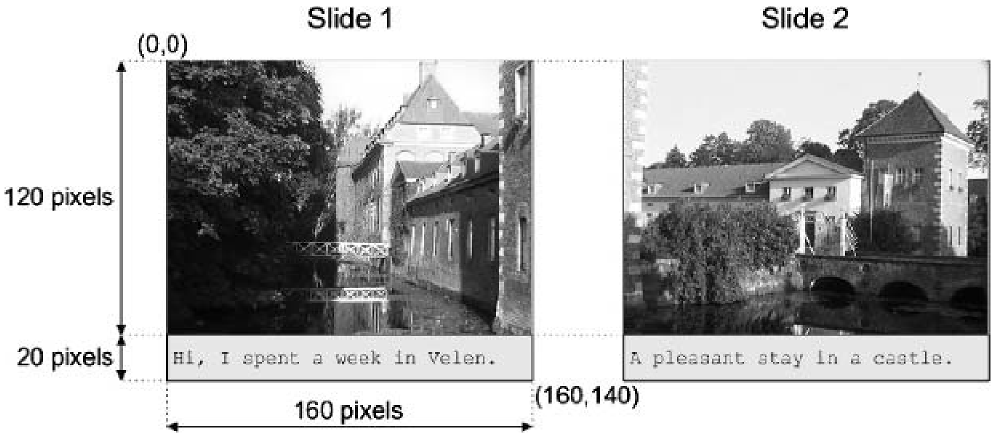


Figure 5.23 Message rendering on receiving device

one-part structure for which the encapsulated element is the protected media object as shown below:

```
Content-type: application/vnd.oma.drm.message;
              boundary = boundary_mms
```

As for multipart structures, the boundary delimiter of the forward-lock content type is used for delimiting the encapsulated element. The whole multimedia message cannot be encapsulated in a forward-lock envelope. Figure 5.24 shows a multimedia message in which the image of the first slide is encapsulated in a forward-lock envelope.

Note that the OMA DRM *forward-lock* technique offers a simple mechanism for protecting media objects against forwarding and modifications in a fairly secure environment. In the Internet world, such a technique would be easily bypassed by any knowledgeable software developer. However, MMS device resources are not yet completely open to applications developed by parties other than the device vendors and their trusted partners. In this context, it is consequently more difficult to bypass the OMA DRM forward-protection.

Box 5.8 Relationships between Message Distribution Indicator and OMA DRM forward-lock

The application of OMA DRM forward-lock, in the context of MMS, is introduced in standards from MMS 1.2. The concept of message distribution indicator was also introduced in standards from MMS 1.2. Note that these two concepts are not competing technologies for DRM. The message distribution indicator is simply an "informational" indication provided to the user stating whether or not the VAS Provider (VASP), originating the message, requested the message not to be forwarded by the recipient(s). The standard does not mandate any functional requirements on the recipient device for preventing the redistribution (or modification) of the associated media object(s)/message. On the other hand, a device compliant to OMA DRM standards is expected to prevent the user from forwarding (or modifying) any media object encapsulated in an OMA-DRM forward-lock envelope.

5.31.2 Combined Delivery

With *combined delivery*, the media object is included as part of the multimedia message along with the right object. The right object indicates the restrictions regarding the usage and redistribution of the media object.

5.31.3 Separate Delivery

With *separate delivery*, the media object is encrypted and included as part of the multimedia message. The right object is not included as part of the multimedia message and is provided to the device using a separate distribution channel (e.g., WAP push).

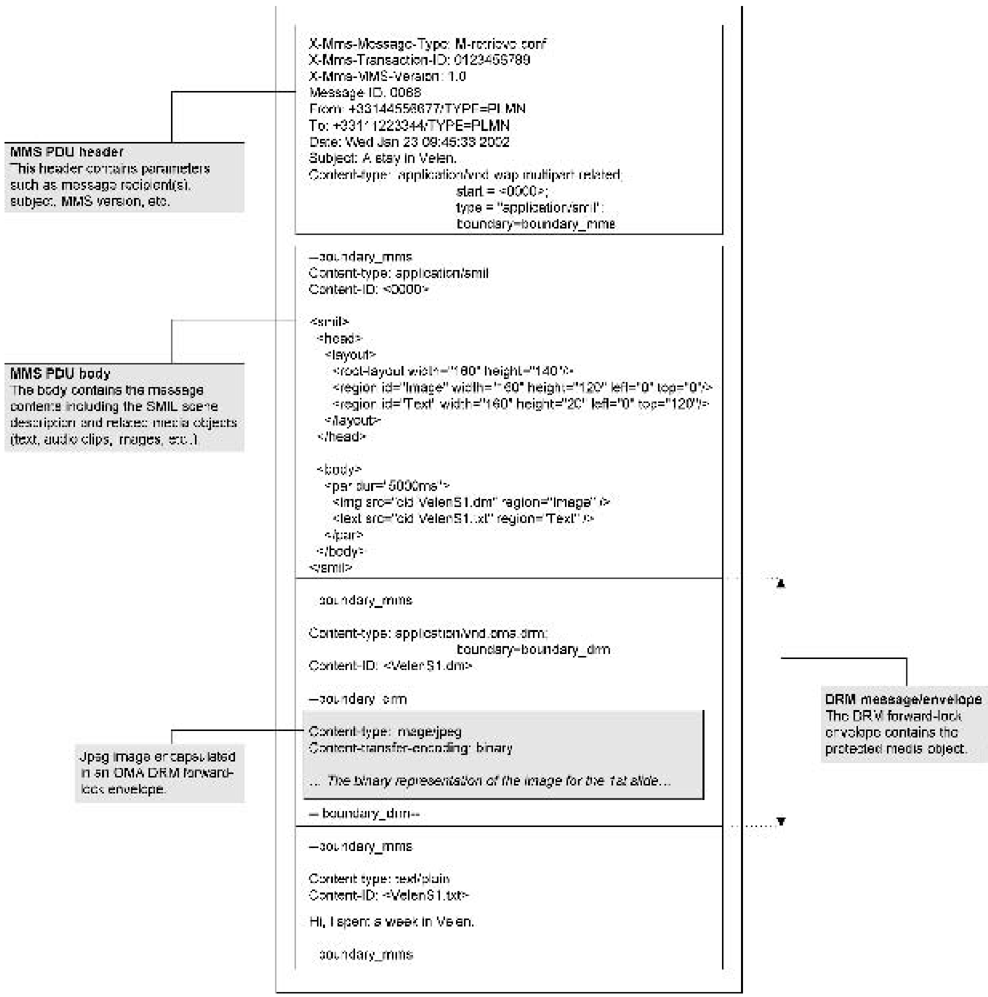


Figure 5.24 Multimedia message with forward protected media object

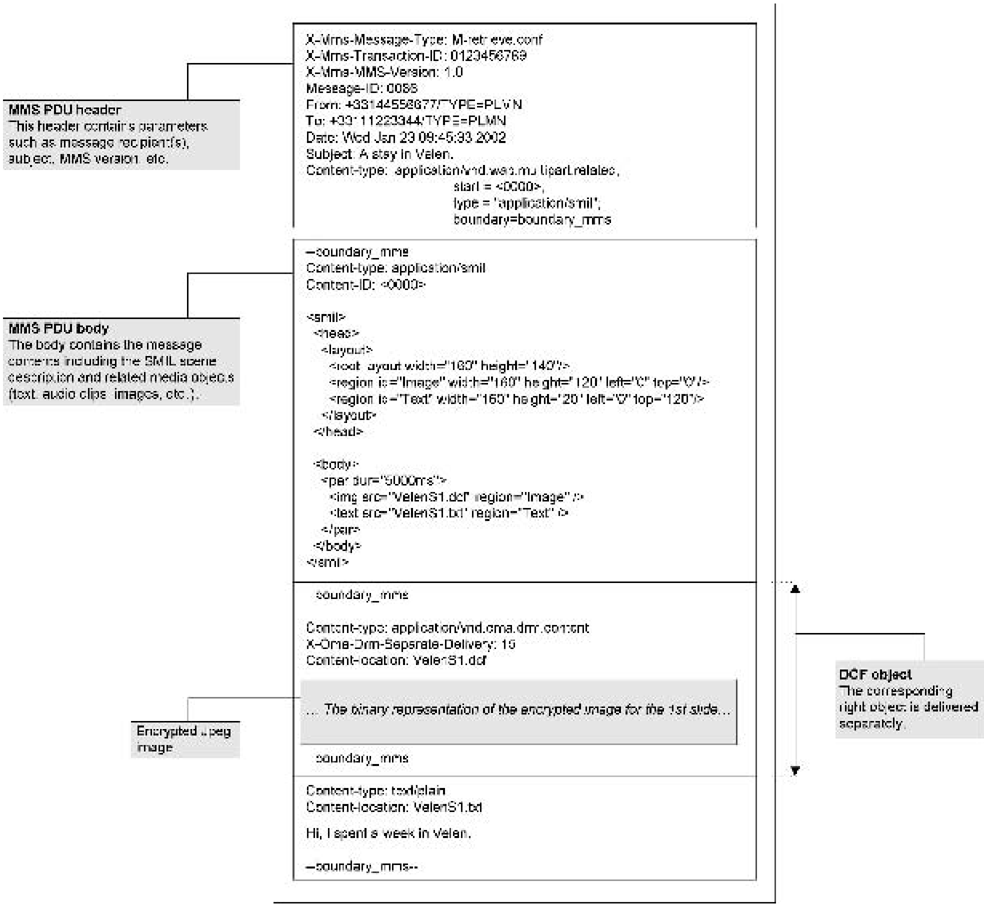


Figure 5.25 Multimedia message with DRM protected object (separate delivery)

Figure 5.25 shows a multimedia message containing a media object protected according to the separate delivery mechanism. The presence of the `X-Oma-Drm-Separate-Delivery` parameter warns the receiving device that the required rights object should arrive via a separate distribution channel. Optionally, a delay time is assigned to this parameter to indicate the latest time when the right object is planned to be delivered. In this example, the right object should arrive in the device no later than 15 seconds after the successful retrieval of the multimedia message.

According to the device capabilities, the user may be able to forward a previously received message containing a protected object, without forwarding the corresponding right object of course. In this case, the recipient of the forwarded message is proposed to acquire the rights object from the content owner (Web or WAP access to a content portal). This scenario represents the basis of the *superdistribution* of protected contents in the MMS environment.

5.32 Postcard Service

As introduced earlier in this chapter, the *postcard service* consists of sending a multimedia message containing a photo along with a postal address and a greeting text to a value-added service provider. Upon receipt of the multimedia message, the service provider prints the photo on the verso of a blank postcard (image side) along with the greeting text on the back of the postcard as shown in Figure 5.26. Once printed, the postcard is sent to the recipient(s) (postal address specified as part of the multimedia message) via the conventional post office.

Prior to MMS 1.3, several proprietary implementations of the postcard service were implemented and the service gained success rapidly in the market. In order to ensure interoperability between different implementations, OMA MMS 1.3 introduces a standard realization of the postcard service. This realization specifies that the addressing information

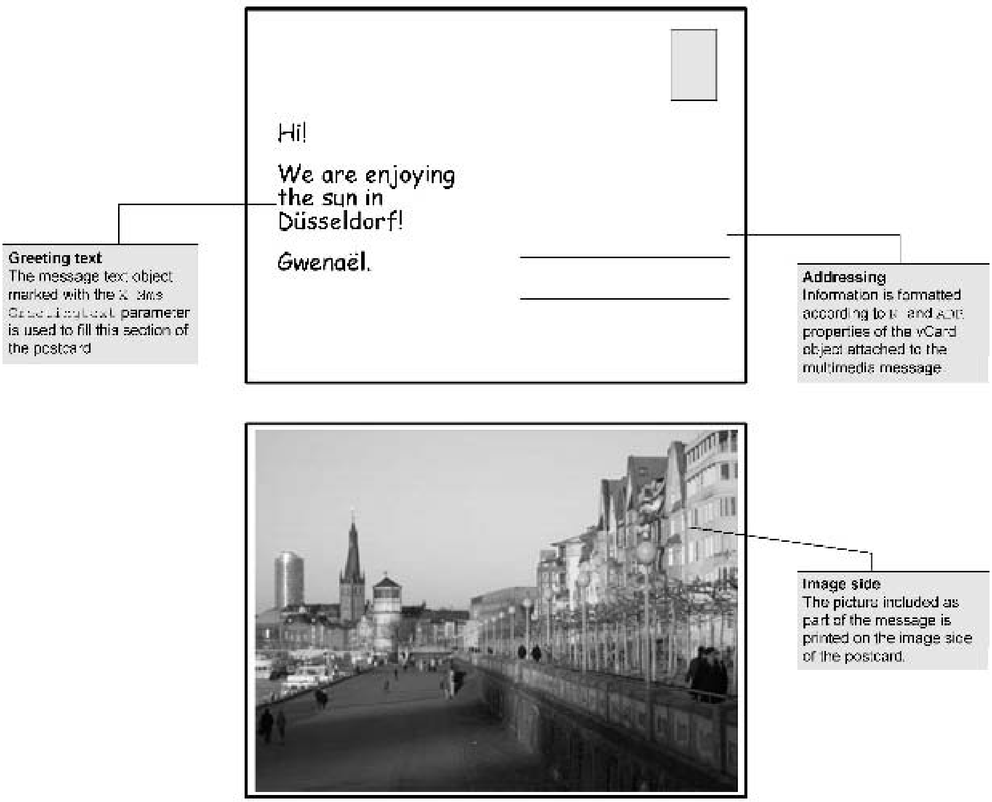


Figure 5.26 Postcard message - Mannesmannufer in Düsseldorf

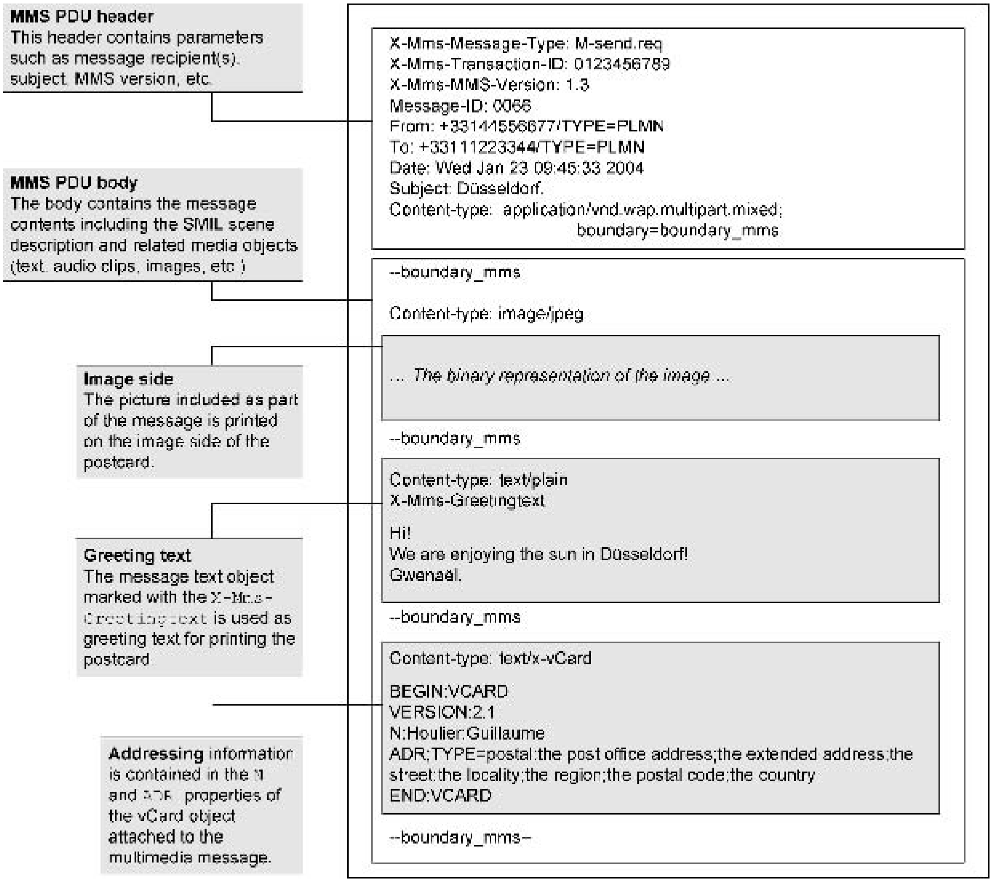


Figure 5.27 Postcard message–message structure

is provided as part of the `N` and `ADR` properties of a vCard object attached to the multimedia message (one vCard object per recipient). The greeting text is provided as part of a text media object marked with the `X-Mms-Greetingtext` parameter. An example of postcard message is shown in Figure 5.27. The vCard format is further described in Section 4.4.14.

5.33 Message Size Measurement

A definition for message size is difficult to derive for multimedia messages. Indeed, a multimedia message can take various forms according to the interface it is conveyed over. Content adaptation can also change the size of a retrieved message compared to the size of the submitted message.

However, it is fundamental for the success of MMS to rely on a commonly understood message size definition since it is used for charging purpose, for characterizing a message in a notification, and so on. In the past, standardization organizations published various methods for calculating the size of a message. Only recently, 3GPP published a method that seems to have won wide acceptance ([3GPP-23.140] Release 5). The definition is as follows:

Message size: the size of a multimedia message is calculated as the sum of the size of the message subject and the size of all elements (including scene description) encapsulated in the message body parts.

The size of the message subject is the size of the value assigned to the subject parameter without the subject parameter name (textual representation) or corresponding assigned number (binary representation). For instance, if the subject is defined as "subject: A small house" (textual representation), then only "A small house" is taken into consideration for the size calculation.

The size of an element encapsulated in a message body part is either of the following:

- The size of the media object (e.g., image, video clip, etc.) if the content type of the message body part is not a multipart structure (`multipart/related` or `multipart/mixed`). Body part parameters (e.g., `content-type`, `content-location`, etc.) are not taken into account for the calculation of the size of a message element.
- The size of an element that is a multipart/one-part structure (e.g., protected media object, see Section 5.31) is the total number of bytes contained in the multipart element contained in the main body part excluding parameters of the main body part but including parameters of the inner multipart element(s).

5.34 Commercial Solutions and Developer Tools

Work on the design of standards for the Multimedia Messaging Service (MMS) had been kicked off in 1998. Four years later, first commercial implementations of MMS Centers (MMSCs) and MMS phones appeared on the market. Today, MMS is still in its infancy, but major industry players endorsed the service and have great expectations of its commercial future.

At the time of writing, more than 120 different MMS phone models were available on the market. First MMS phones appeared on the market in the second quarter of 2002, and each new quarter saw the availability of a few more models. It would not be practical to list the capabilities of these phones in this book. However, this book comes with a companion website which references MMS devices and describes their capabilities (the ones advertised in corresponding user agent profiles by device manufacturers). Figure 5.28 shows a screenshot of the device section of the website. The website is available at the following address:

`http://www.lebodic.net/mms_resources.htm`

The following vendors are known to commercialize MMSC solutions: Alcatel, Comverse, Ericsson, Huawei, LogicaCMG, Materna, Nokia, Oksijen Technoloji, Openwave, Symsoft, Tecnomen, TeleDNA, Unisys, and Wiral. However, the MMSC market is mainly shared between the following four vendors:

- Nokia
- Ericsson
- LogicaCMG
- Comverse.

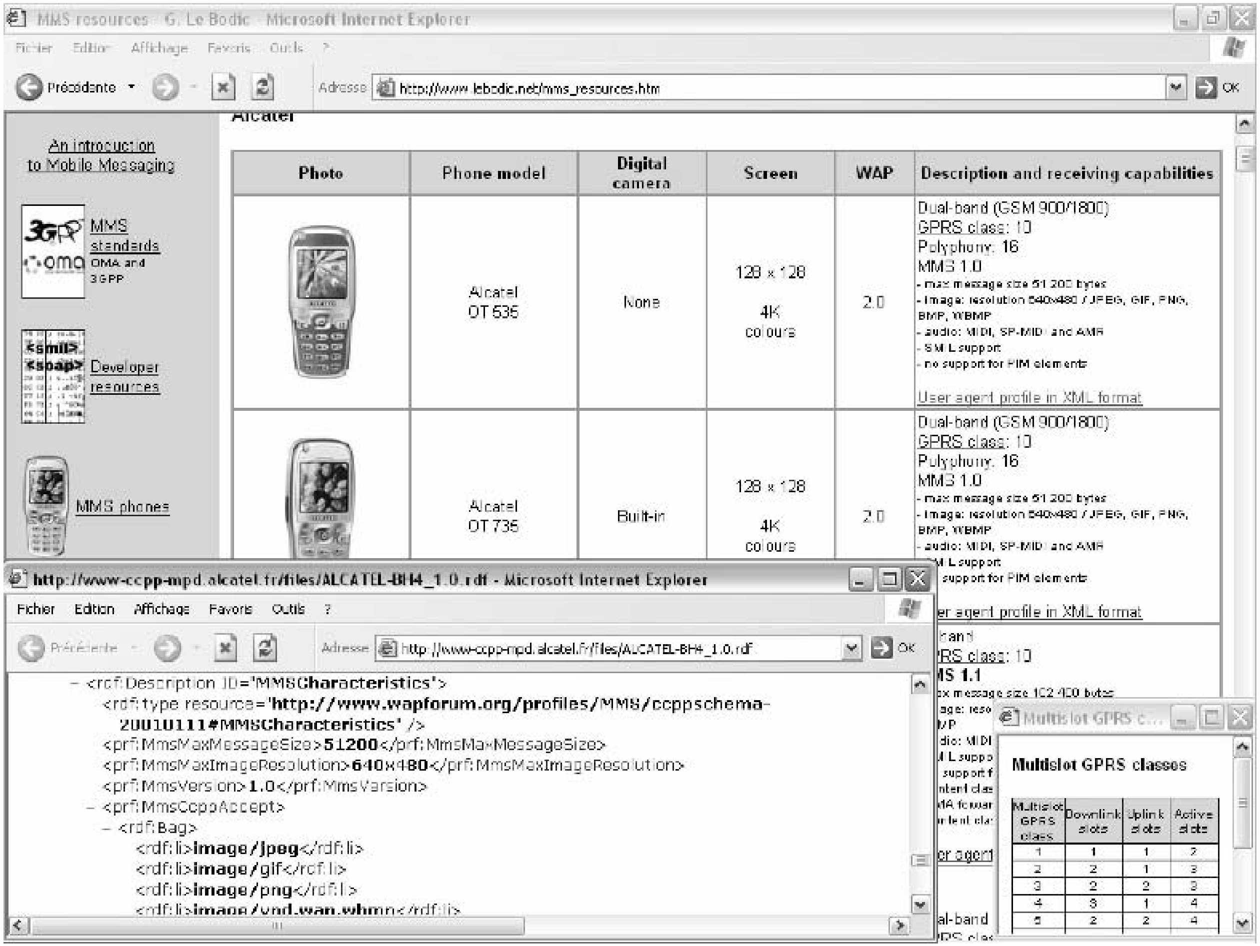


Figure 5.28 Screenshop of this book companion website

Several companies provide tools to allow the development of MMS software applications. This includes the following tools:

- Nokia emulator for MMSC interface: this tool emulates the interface of Nokia's MMSC for Value-Added Service (VAS) applications. As described in Chapter 6, this interface is known as the MM7 interface.
- Nokia MMS Java library: the Java library can be used for developing MMS-based applications in Java. This library includes the following features:
  - message creation and encoding
  - message decoding
  - message sending to Nokia MMSC or to Nokia emulator MMSC.
- Nokia developer's suite for MMS: this is a software add-on for Adobe GoLive 6.0. Adobe GoLive, with this add-on, becomes an authoring tool facilitating the generation of multimedia messages with rich multimedia presentations.
- Nokia Series 60 SDK Symbian OS and MMS extension: this software development kit allows the development of applications for devices based on the Series 60 platform. This SDK includes a Series 60 emulator enabling the development and testing of applications with a PC.
- Nokia mobile Internet tool kit: this tool can be used for testing multimedia messages. It features a ready-to-use application for creating a message from various media objects. Created messages can be provided to one of Nokia's handset simulator (e.g., Nokia 6590 and Nokia 7210 emulators).
- Comverse development environment: this development environment contains a set of tools including the following:
  - MMS message player (preview multimedia messages on a variety of MMS phones)
  - MMS software development kit (Java libraries for message composition, submission to Comverse MMS center, message reception from Comverse MMS center, and message parsing) and WAV to AMR converter.
- Sony Ericsson MMS home studio: the studio includes the following:
  - MMS message editor (created message can be sent to an Ericsson phone connected to a PC via infrared link, Bluetooth, or serial cable)
  - MMS message viewer (preview the message with Ericsson handset simulator). These tools are available from Ericsson mobility world website (see Box 5.9).
- Ericsson AMR converter: this basic tool converts WAV files to AMR files.

Box 5.9 Resources for developers of MMS applications

Nokia developer Forum at http://www.forum.nokia.com
Ericsson Mobility World at http://www.ericsson.com/mobilityworld
Comverse developer zone at https://developer.comverse.com/dzone/
Access to resources for developers requires prior online registration.

5.35 The Future of MMS

When guessing what the future of MMS could be, a glance at ongoing standardization activities can provide some outlook. 3GPP has recently been focusing on improving the interface to enable interactions between value-added service providers and the MMS system. This interface is key importance for content-to-person messaging which is likely to be one of the major revenue stream for MMS in the future. GSMA is looking at enabling interworking between different MMS platforms. Today, national interworking has happened, however, the exchange of multimedia messages over international borders is still quite a challenge. This should change in the near future. 3GPP2 is looking at realizations of MMS which do not rely on WAP. One of such realizations relying on the Session Initiation Protocol (SIP) for notification and report management, and HTTP for transport of messages has already been defined. Alike solutions will become of interest when deploying IMS, with the objective of relying on IP-based protocols only. Many application platforms have been deployed in order to support a growing number of services (SMS, MMS, Email, browsing, instant messaging, etc.). These platforms have been designed as ‘‘silo’’ platforms with their own storage capabilities, transcoding functions, user profile repositories, etc. OMA is looking at optimized design alternatives by allowing platforms to be horizontally integrated in a layer where modular platforms can advertise their services to other platforms and common functions (storage, repositories, etc.) can be shared between platforms. These activities tell that there is still much room for improving MMS as it stands today.

Back in 2002, MMS solutions did support limited message size (30 KB) and low number of formats and codecs (no video, no vector graphics). MMS standards have evolved and solutions now support large message size (300 KB) and many sophisticated formats and codecs. From an end-user perspective, the form-factor of devices has evolved to integrate more mutlimedia capabilities. Initial MMS handsets were, without any possible confusion, always referred to as phones, or multimedia phones. New handsets available on the market already support 4 or 5 million pixel built-in cameras, with high-resolution screens and ability to record short video clips. It has now become difficult to really differentiate a multimedia phone from a digital camera, and to some extent to a camcorder. Multimedia phones are becoming digital cameras with, in addition from their phone capabilities, the ability to exchange instantly photos with friends, ordering prints from an online shop or sending a postcard. It looks like mobile phone and digital camera industries are now sharing common markets. Hopefully, the synergy between the two industries will soon bring some very exciting devices in the hands of mobile users.

5.36 Further Reading

[1] T. Natsuno, *i-mode Strategy*, John Wiley & Sons, Chichester, 2002.
[2] I.E.G. Richardson, *Video Codec Design*, John Wiley & Sons, Chichester, 2002.
[3] D.C.A. Bulterman, (Ed. Peiya Liu), *SMIL 2.0, part I: overview, concepts, and structure*, IEEE Multimedia Magazine, 8(4), 2001, 82–88.
[4] D.C.A. Bulterman, (Ed. Peiya Liu), *SMIL 2.0, part II: examples and comparisons*, IEEE Multimedia Magazine, 9(1), 2002, 74–84.

6

Multimedia Messaging Service, Transactions Flows

Each end-to-end feature offered by the Multimedia Messaging Service (MMS) relies on a series of consecutive transactions occurring over one or more of the 11 identified MMS interfaces. These transactions allow the transfer of messages and associated reports between MMS communicating entities including network servers and mobile devices.

This chapter presents transactions required for the realization of the service in the context of two representative scenarios: the person-to-person and content-to-person scenarios. Furthermore, this chapter provides a comprehensive description of transactions than can occur over MM1, MM4, MM7, and STI interfaces. The transport protocols for these interfaces have been fully defined by standardization organizations. Other relevant interfaces are also outlined.

6.1 Introduction to the MMS Transaction Model

MMS entities (MMS clients or MMS centers) communicate by invoking *transactions* over a set of 11 interfaces. A transaction is typically composed of a service *request* and a corresponding service *response/confirmation* containing the transaction results (e.g., message sending request and message sending confirmation). However, several transactions are limited to a service request only and are also known as *indications*. A Protocol Data Unit (PDU) is associated with each service request, response or indication that can occur over one of the MMS interfaces. A PDU is composed of a set of mandatory, optional, or conditional parameters.

In this book, the Third Generation Partnership Project (3GPP) convention for naming PDUs is employed for the description of transactions that can occur over MM4 and MM7 interfaces. The convention consists of suffixing the request name by `.REQ` and suffixing the response name by `.RES`. In addition, request and response names are prefixed by the associated interface name. For instance, the request for routing forward a message over the MM4 interface is named `MM4_forward.REQ` and the corresponding response is named

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2

MM4_forward.RES. On the other hand, the Open Mobile Alliance (OMA) convention for naming PDUs is employed for the description of transactions that can occur over the MM1 interface. This convention is further presented in Section 6.2.

6.1.1 Person-to-Person Scenarios

The simplest transaction flow for a message exchange is the one that involves two MMS clients attached to the same MMS Environment (MMSE). In this context, the message exchange usually involves one MMSC only. The process of exchanging a message is composed of four steps as shown in Figure 6.1 and described below:

1. *Message submission over the originator MM1 interface*: this transaction is composed of a submission request (M-send.req PDU) and a corresponding submission response (M-send.conf PDU).
2. *Message notification over the recipient MM1 interface*: this transaction is composed of a notification indication (M-notification.ind PDU) and a notification response indication (M-notifyresp.ind PDU).
3. *Message retrieval over the recipient MM1 interface*: this transaction is composed of a retrieval request (WSP/HTTP GET.req PDU), retrieval response (M-retrieve.conf PDU), and optionally a retrieval acknowledgment indication (M-acknowledge.ind PDU).
4. *Delivery reporting over the originator MM1 interface*: a delivery report is conveyed over the MM1 interface only if the generation of a delivery report was requested during message submission and if the recipient did not deny the generation of such a report. The transaction is composed of a delivery-report indication (M-delivery.ind PDU).

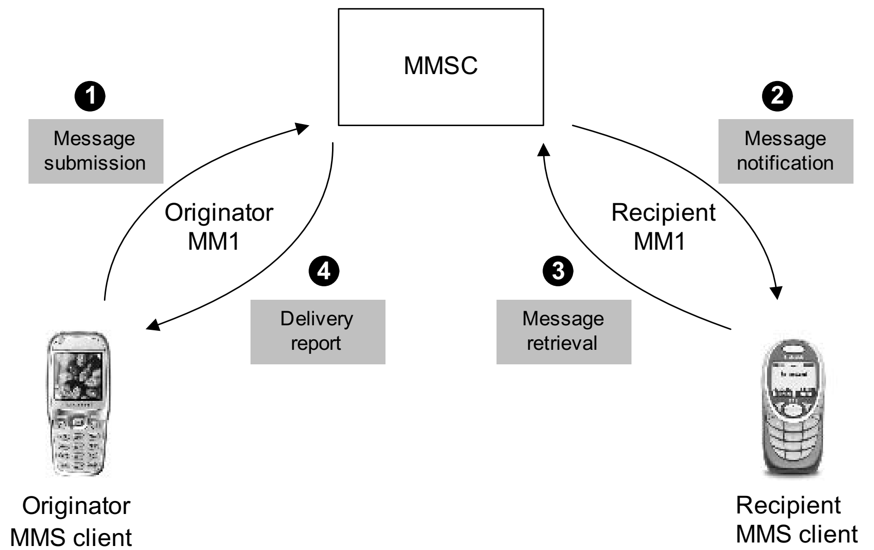


Figure 6.1 General transaction flow/message exchange – single MMS environment

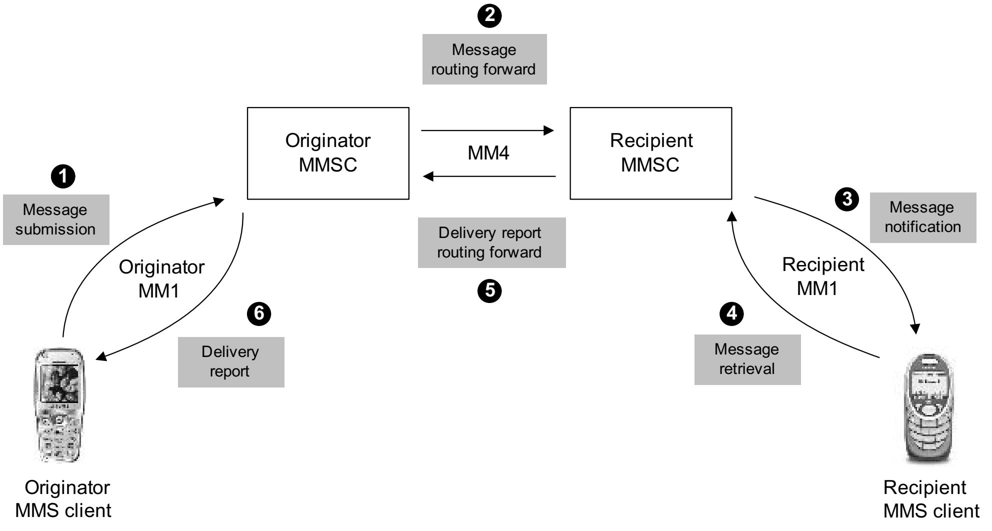


Figure 6.2 General transaction flow/message exchange – multiple MMS environments

The exchange of a message between MMS clients attached to distinct MMSEs involves two MMSCs: the originator MMSC and the recipient MMSC. The two MMSCs are interconnected via the MM4 interface. The process of exchanging a message is composed of four to six steps as shown in Figure 6.2 and described below:

1. Message submission over the originator MM1 interface.
2. *Routing forward the message over the MM4 interface*: this transaction is composed of a forward request (`MM4_forward.REQ` PDU) and a corresponding forward response (`MM4_forward.RES` PDU).
3. Message notification over the recipient MM1 interface.
4. Message retrieval over the recipient MM1 interface.
5. *Routing forward the delivery report over the MM4 interface*: this transaction is composed of a submission request (`MM4_delivery_report.REQ` PDU) and a corresponding submission response (`MM4_delivery_report.RES` PDU). The delivery report is generated by the recipient MMSC upon confirmation of message retrieval by the recipient MMS client.
6. Delivery reporting over the originator MM1 interface.

Note that a read report can also be generated when the message has been read by the recipient. For the sake of clarity, transactions related to the management of a read report are not shown in Figures 6.1 and 6.2. A read report is generated by the recipient MMS client when the recipient has read or deleted the message only if the generation of a read report was requested by the originator and if the recipient did not deny the generation of such a report. In the recipient environment, the read report is conveyed from the recipient MMS client to the recipient MMSC in the form of a read-report indication (`M-read-rec.ind` PDU) over the MM1 interface. The read report is forwarded over the MM4 interface with a transaction

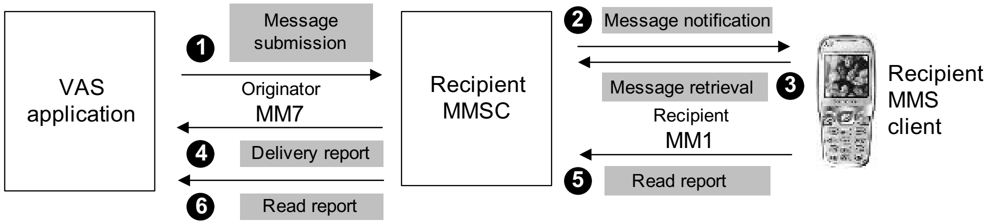


Figure 6.3 General transaction flow/message exchange – from VAS application to MMS client(s)

composed of a forward request (`MM4_read_reply_report.REQ` PDU) and a forward response (`MM4_read_reply_report.RES` PDU). Eventually, the read report is pushed to the originator MMS client as part of a read-report indication (`M-read-orig.ind` PDU) in the originator MMSE over the MM1 interface.

6.1.2 Content-to-Person Scenarios

Content-to-person refers to the scenario where the message originates from a VAS application and is delivered to one or more MMS recipients. For this purpose, the VAS Provider (VASP) operates a VAS application, usually Internet hosted and connected to an MMSC via the MM7 interface. In this configuration, the MMSC interacts with recipient MMS clients over the MM1 interface as already described in the preceding section.

Figure 6.3 shows interactions between a VAS application, the MMSC, and one recipient MMS client for the exchange of a message from the VAS application down to the recipient MMS client. Optionally, the VAS application can request the generation of delivery and read reports for the message. Note that the recipient often has the possibility of denying the generation of such reports.

The process of exchanging a message from the VAS application to the recipient MMS client is composed of four to six steps as described below:

1. *Message submission over the MM7 interface*: this transaction is composed of a submission request (`MM7_submit.REQ` PDU) and a corresponding submission response (`MM7_submit.RES` PDU).
2. Message notification over the MM1 interface (see preceding section).
3. Message retrieval over the MM1 interface (see preceding section).
4. *Delivery reporting over the originator MM7 interface*: a delivery report is conveyed over the MM7 interface only if the generation of a delivery report was requested during message submission and if the recipient did not deny the generation of such a report. The transaction is composed of a delivery-report request (`MM7_delivery_report.REQ` PDU) and a corresponding delivery-report response (`MM7_delivery_report.RES` PDU).
5. Read reporting over the MM1 interface (see preceding section).
6. *Read reporting over the MM7 interface*: as for delivery reports, a read report is conveyed over the MM7 interface only if the generation of a read report was requested during message submission and if the recipient did not deny the generation of such a report. The

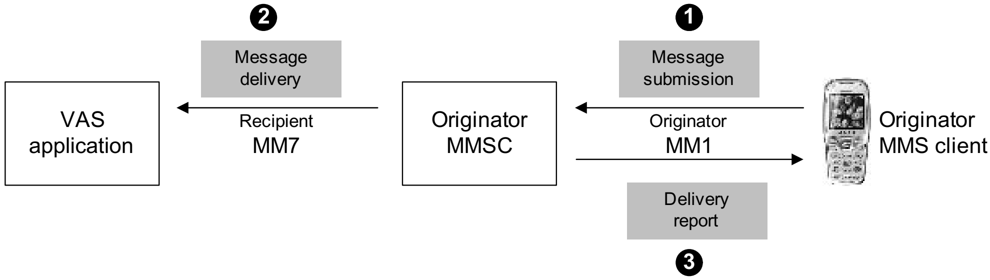


Figure 6.4 General transaction flow/message exchange – from MMS client to VAS application

transaction is composed of a read-report request (`MM7_read_reply_report.REQ` PDU) and a corresponding read-report response (`MM7_read_reply_report.RES` PDU).

There is always a period during which the submitted message is stored temporarily in the MMSC, awaiting retrieval from the recipient MMS client. During this period, it is possible for the originating VAS application to cancel the message delivery or to replace the message due to be delivered. The cancel transaction over the MM7 interface is composed of a cancel request (`MM7_cancel.REQ` PDU) and a corresponding cancel response (`MM7_cancel.RES` PDU). Similarly, the replace transaction over the MM7 interface is composed of a replace request (`MM7_replace.REQ` PDU) and a corresponding replace response (`MM7_replace.RES` PDU).

On the other way round, the MMS client also has the possibility of submitting a message addressed to a VAS application. This may be a reply related to a previous message originated by the VAS application or it can be a new unrelated message. Figure 6.4 shows interactions between the MMS client, the MMSC, and the VAS application for the exchange of a message from the MMS client up to the VAS application.

The process of exchanging a message from the MMS client to the VAS application is composed of two to three steps as described below:

1. Message submission over the originator MM1 interface (see Section 6.1.1).
2. *Message delivery over the MM7 interface*: this transaction is composed of a delivery request (`MM7_deliver.REQ` PDU) and a corresponding delivery response (`MM7_deliver.RES` PDU).
3. Delivery reporting over the MM1 interface (see Section 6.1.1).

6.1.3 How to Read the PDU Description Tables

This chapter details the composition of each PDU that can occur over several MMS interfaces. The composition of a PDU is represented in a concise table as shown in Figure 6.5.

Values of the "From" column indicate from which version of the MMS standards the corresponding parameter becomes applicable. For instance, the value "1.1" means that the corresponding parameter was introduced in MMS 1.1 and, therefore, such a parameter may

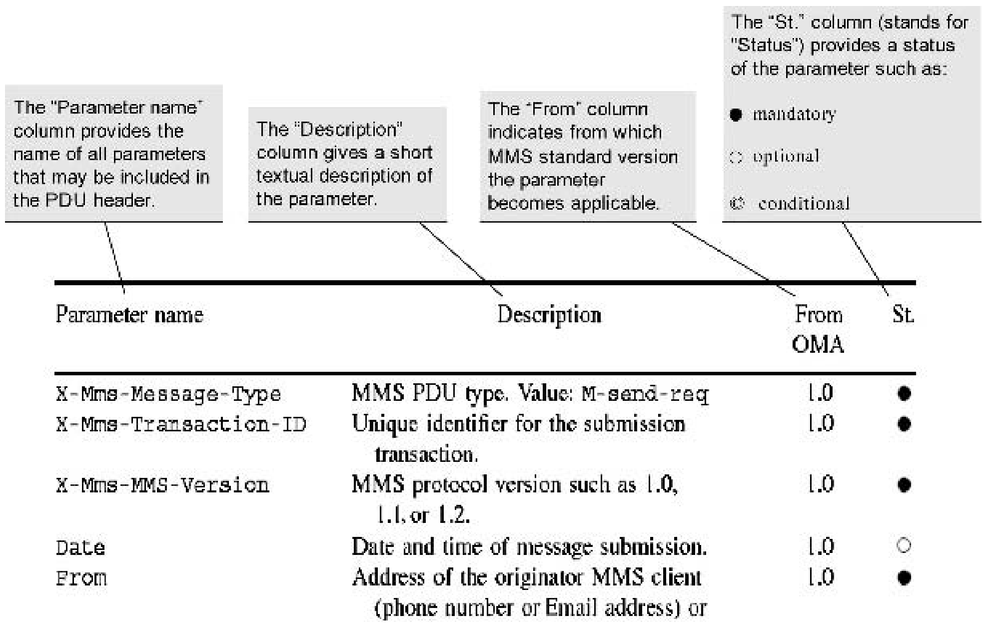


Figure 6.5 How to read the PDU description tables

be found in MMS PDUs versioned 1.1, 1.2, and so on but will never be present in a PDU versioned 1.0.

The column "St." which stands for "Status," indicates whether the presence of the corresponding parameter is mandatory, optional, or conditional as shown below:

● mandatory
○ optional
© conditional.

6.2 MM1 Interface, MMS Client–MMSC

The *MM1 interface* allows interactions between the MMS client and the MMSC. Several primitives, also known as PDUs, can be invoked over this interface for transactions such as message notification, message submission, message retrieval, and so on. According to the OMA terminology for MMS [OMA-MMS-Arch], the MM1 interface is also known as the MMS_M interface. In this book, the convention for naming MM1 PDUs is the one used in OMA standards. This convention consists of prefixing the primitive name with the interface name and suffixing the primitive name with the primitive type as shown in Figure 6.6.

Three types of PDUs can be invoked over the MM1 interface which are as follows:

- *Request*: a request PDU is invoked from an MMS entity (MMS client or MMSC) to request a service to be provided by another MMS entity (MMS client or MMSC). In this context, the serving MMS entity, which accepts or rejects the request, notifies the requesting MMS entity of the request status with a confirmation PDU. A request PDU

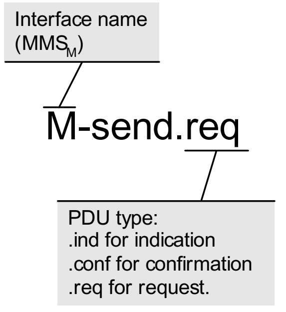


Figure 6.6 Convention for naming MM1 PDUs

is often marked with a transaction identifier. The name of a request PDU is suffixed with `.req` (e.g., `M-send.req`).

- *Confirmation/response*: a confirmation PDU is invoked by a serving MMS entity to confirm the status of a previously requested service. It is marked with the transaction identifier of the corresponding request PDU. The name of a confirmation PDU is suffixed with `.conf` (e.g., `M-send.conf`).
- *Indication*: an indication PDU is invoked by an MMS entity to notify another MMS entity of the occurrence of an event (message notification, reports, etc.). The name of an indication PDU is suffixed with `.ind` (e.g., `M-notification.ind`). An indication is not confirmed.

Table 6.1 lists the ten transactions that can occur over the MM1 interface (20 PDUs).

For a transaction composed of a request and a corresponding confirmation, the confirmation usually contains a status code indicating if the serving MMS entity accepts or rejects the service request. In the case of rejection, the confirmation status code identifies the reason for rejection. It is common to differentiate rejections due to permanent errors from rejections due to transient errors. A rejection due to a *permanent error* means that the request in its current form is not acceptable, whereas a rejection due to a *transient error* means that the request cannot be processed because of some transient error conditions, but the request may be accepted, in its current form, later. For instance, an MMS client failing to submit a message to an MMSC because of a transient error (e.g., MMSC busy) may reattempt to request the service in the same way at a later stage.

The next sections of this chapter present interactions between MMS entities for each transaction flow (e.g., message submission, message retrieval, etc.). MMS PDUs involved in these transaction flows are composed of a PDU *header* (a sequence of parameters) and an optional PDU *body* (containing multimedia message contents or informational text) as shown in Figure 6.7. Each PDU is conveyed over the MM1 interface in a binary form for transport efficiency. Section 6.2.11 of this chapter explains how each parameter is binary encoded according to the type of its assigned values (integer, date, string, etc.) and provides guidelines regarding the maximum lengths of several parameter values. Once binary encoded, each MMS PDU is encapsulated in a transport-level Wireless Session Protocol (WSP) or HTTP PDU (request/response) and conveyed over a data connection established between the mobile device and the network or conveyed as part of an SMS message (WAP push).

Table 6.1 List of MM1 transactions/PDUs

Transaction	PDU name	Description	From OMA
Message submission	M-send.req	Message submission request	1.0
	M-send.conf	Message submission confirmation	
Notification	M-notification.ind	Message notification indication	1.0
	M-notifyresp.ind	Message notification response indication	
Message retrieval	WSP/HTTP GET.req	Message retrieval request	1.0
	M-retrieve.conf	Message retrieval confirmation	
	M-acknowledge.ind	Message retrieval acknowledgment indication	1.0
Delivery report	M-delivery.ind	Delivery report indication	1.0
Read report	M-read-rec.ind	Read-report indication (recipient MMSE)	1.1
	M-read-orig.ind	Read-report indication (originator MMSE)	
Message forward	M-forward.req	Message forward request	1.1
	M-forward.conf	Message forward confirmation	
Message store or update into MMbox	M-Mbox-Store.req	MMbox message store/update request	1.2
	M-Mbox-Store.conf	MMbox message store/update confirmation	
View contents of MMbox	M-Mbox-View.req	MMbox contents view request	1.2
	M-Mbox-View.conf	MMbox contents view confirmation	
Message upload to MMbox	M-Mbox-Upload.req	MMbox message upload request	1.2
	M-Mbox-Upload.conf	MMbox message upload confirmation	
Message deletion from MMbox	M-Mbox-Delete.req	MMbox message deletion request	1.2
	M-Mbox-Delete.conf	MMbox message deletion confirmation	

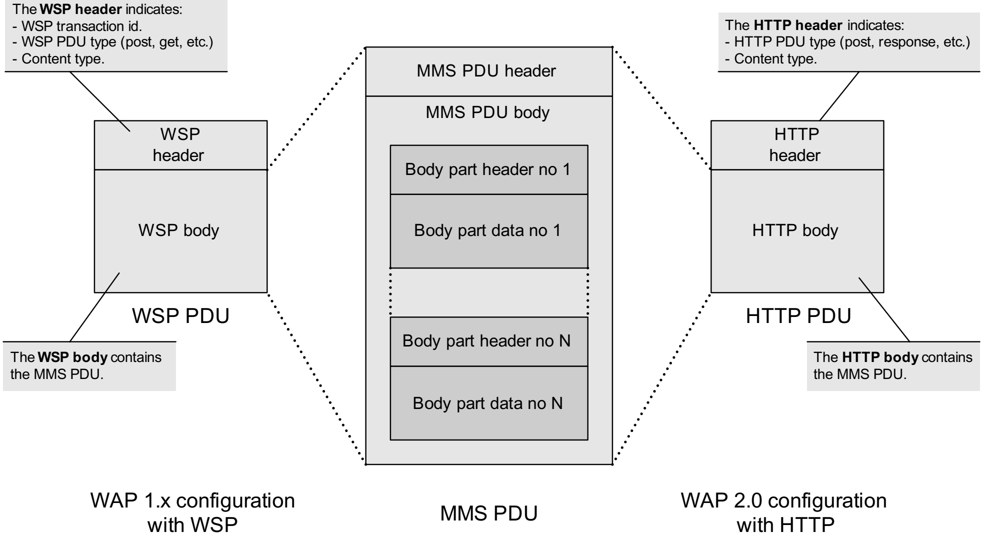


Figure 6.7 Encapsulation of an MMS PDU in a WSP or HTTP PDU

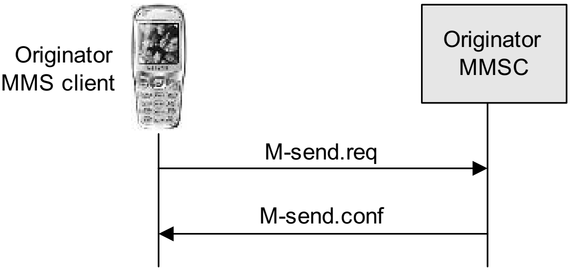


Figure 6.8 MMI message submission transaction flow

Figure 6.7 shows how an MMS PDU is encapsulated in a WSP or an HTTP PDU. At the WSP/HTTP layer, an MMS PDU is marked with the following content type:

`application/vnd.wap.mms-message`

Each MMS PDU header is composed of a number of optional, conditional, and mandatory parameters. If an optional parameter is omitted from the PDU header, then a default value is implicitly considered for this parameter. In this book, the default value is indicated in the corresponding parameter description when appropriate.

6.2.1 Message Submission

Message submission refers to the submission of a multimedia message from the originator MMS client to the originator MMSC. The transaction flow for message submission is shown in Figure 6.8. The PDU corresponding to the submission request is named `M-send.req` and is confirmed by the PDU named `M-send.conf`.

In WAP technical realizations of MMS, the mobile device establishes a data connection with the network. This allows the MMS client to communicate with the MMSC. This data connection can rely on a circuit-switched data connection (e.g., GSM) or a packet-switched connection (e.g., GPRS). The connection is typically established through a WAP gateway in a WAP 1.x configuration. It is common for operators to set up a dedicated WAP gateway for handling MMS traffic in addition to the one that handles browsing and download traffics.

Parameters composing the message submission request PDU (`M-send.req`) are presented in Table 6.2, whereas parameters composing the message submission confirmation PDU (`M-send.conf`) are presented in Table 6.3 .The body of the submission request PDU contains the submitted multimedia message and the PDU header represents the message envelope. The submission confirmation PDU does not contain any body (header only).

The originator MMS client can specify a date and time for the submission request (parameter `Date` of the request in the GMT[1] format). If such date and time are not provided by the MMS client, then it becomes the originator MMSC's responsibility to provide the date and time for the submitted message. In any case, the MMSC always has the possibility of overwriting a date and time provided by the MMS client. Note that, unfortunately, the MMS protocol does not provide any means for the MMS client to be informed of the date/time assigned by the originator MMSC.

[1]Greenwich Mean Time.

Table 6.2 MM1 message submission request (`M-send.req`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-send-req`	1.0	●
`X-Mms-Transaction-ID`	Unique identifier for the submission transaction.	1.0	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.0, 1.1, 1.2, or 1.3.	1.0	●
`Date`	Date and time of message submission.	1.0	○
`From`	Address of the originator MMS client (phone number or Email address) or "insert token" if the originator address is to be provided by the MMSC.	1.0	●
`To`	One or multiple addresses (phone number or Email address) for message recipient(s). Primary recipients.	1.0	○
`Cc`	One or multiple addresses (phone number or Email address) for message recipient(s). Secondary recipients.	1.0	○
`Bcc`	One or multiple addresses (phone number or email address) for message recipient(s). Secondary recipients/blind copy.	1.0	○
`Subject`	A short textual description for the message.	1.0	○
`X-Mms-Message-Class`	Message class such as "auto" (automatically generated by the MMS client), "personal" (default), "advertisement," and "informational." Other classes can also be defined in the form of text strings.	1.0	○
`X-Mms-Expiry`	Expiry date. Default value for this parameter is "maximum."	1.0	○
`X-Mms-Delivery-Time`	Earliest delivery time. Default value for this parameter is "immediate delivery."	1.0	○
`X-Mms-Priority`	Priority such as "low," "normal" (default), or "high."	1.0	○
`X-Mms-Sender-Visibility`	Visibility of the sender address. This parameter is either set to "show" (default) for showing the sender address to recipient(s) or "hide" for hiding the sender address to recipient(s). From MMS 1.2, "show" is not anymore the default value for this parameter. If this parameter is not present in an MMS 1.2 PDU, then network preferences for the sender anonymity feature are used.	1.0	○
`X-Mms-Delivery-Report`	Request for a delivery report. This parameter indicates whether or not delivery report(s) are to be generated for the submitted message. Two values can be assigned to this parameter: "yes" (delivery report is to be generated) or "no" (no delivery report requested). If the message class is "auto," then this parameter is present in the submission PDU and is set to "no."	1.0	○

(*Continued*)

Table 6.2 (*Continued*)

Parameter name	Description	From OMA	St.
`X-Mms-Read-Report`	Request for a read report. This parameter indicates whether or not read reports are to be generated for the message. Two values can be assigned to this parameter: "yes" (read report is to be generated) or "no" (no read report requested). If the message class is auto, then this parameter is present in the submission PDU and is set to "no."	1.0	○
`X-Mms-Reply-Charging`	Request for reply charging. The presence of this parameter indicates that reply charging is requested by the message originator. Two values can be assigned to this parameter: "requested" when the originator is willing to pay for the message reply(s) or "requested text only" when the originator is willing to pay for message reply(s) containing text only. In any case, two parameters (reply message size and reply deadline) specify conditions for the message reply to be paid for by the originator.	1.1	○
`X-Mms-Reply-Charging-Deadline`	Reply charging–deadline. This parameter specifies the latest time for the recipient(s) to submit a message reply. This parameter is only present in the PDU if reply charging is requested.	1.1	○
`X-Mms-Reply-Charging-Size`	Reply charging–maximum message size. This parameter specifies the maximum size for message replies. This parameter is only present in the PDU if reply charging is requested.	1.1	○
`X-Mms-Reply-Charging-ID`	Reply charging–identification. This parameter is inserted in a reply message only and refers to the original message identifier (`Message-ID` parameter).	1.1	○
`X-Mms-Store`	MMBox storage request. This parameter indicates whether the originator MMS client requests to save the message in the originator's MMBox apart from sending it.	1.2	○
`X-Mms-MM-State`	MMBox message state. When `X-Mms-Store` is set to "yes," this parameter indicates the message state in the originator's MMBox (e.g., sent, draft, etc.). If `X-Mms-Store` is set to "yes" and if this parameter is not present then the message default state is "sent."	1.2	○
`X-Mms-MM-Flags`	MMBox message flag. This parameter indicates the list of flags associated to a message stored in the MMBox (considered only if `X-Mms-Store` is set to "yes").	1.2	○
`Content-Type`	Content type of the multimedia message (e.g., `application/vnd.wap.multipart.related`).	1.0	●

Table 6.3 MM1 message submission request (`M-send.req`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-send-conf`	1.0	●
`X-Mms-Transaction-ID`	Unique identifier for the submission transaction. The same as the one for the corresponding submission request.	1.0	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.0, 1.1, 1.2, or 1.3.	1.0	●
`X-Mms-Response-Status`	Status code for the submission transaction. The submission request can be accepted or rejected (permanent or transient errors). See status codes in Appendix F.	1.0	●
`X-Mms-Response-Text`	Human readable description of the transaction status.	1.0	○
`Message-ID`	Message unique identifier. This identifier is always provided by the MMSC if the submission request is accepted.	1.0	○
`X-Mms-Content-Location`	Reference to the message stored in the MMBox. This parameter is present only if the three following conditions are fulfilled: - the originator MMSC supports the MMBox feature - the `X-Mms-Store` parameter was present in the corresponding submission request - the `X-Mms-Store-Status` indicates "success." When available, this parameter provides a reference to the message stored in the MMBox (reference used later for message retrieval or view request).	1.2	○
`X-Mms-Store-Status`	MMBox message status. This parameter is present only if the two following conditions are fulfilled: - the originator MMSC supports the MMBox feature - the `X-Mms-Store` parameter was present in the corresponding submission request. When available, this parameter indicates whether or not the submitted message has been successfully stored in the MMBox. See status codes in Appendix H.	1.2	○
`X-Mms-Store-Status-Text`	MMBox message textual status. Textual description qualifying the value assigned to the `X-Mms-Store-Status` parameter.	1.2	○

Recipient addressing parameters (To, Cc, and Bcc) are all optional. However, at least one recipient address shall be provided by the MMS client to the MMSC. Each one of the addressing parameters may appear multiple times.

The client has the possibility of requesting reply charging for the submitted message (from MMS 1.1). However, reply charging may not be supported by the MMSC. If reply charging is not supported, then the MMSC rejects the submission request by providing a confirmation with the appropriate error code.

If the submission request is accepted by the MMSC, then the MMSC provides a unique message identifier (parameter Message-ID of the confirmation). This identifier can later be used for correlating reports and message replies (reply charging) with the original message.

In the event of the submission request not being accepted by the MMSC, the MMSC specifies the reason for rejection in the confirmation. The reason may be of permanent nature (e.g., message badly formatted) or of transient nature (e.g., MMSC is temporarily unavailable).

If the Multimedia Message Box (MMBox) concept is not supported by the originator MMSC, then the submission request MMBox-related parameters (X-Mms-Store, X-Mms-MM-State, and X-Mms-MM-Flags) are ignored by the MMSC.

The submission request PDU is transferred from the MMS client to the MMSC. It is conveyed as part of a WSP/HTTP Post request. Figure 6.9 shows the partial hexadecimal representation of a submission request PDU with the following characteristics:

```
X-Mms-Message-Type: M-send.req
X-Mms-Transaction-ID: 0123456789
X-Mms-MMS-Version: 1.0
From: +33144556677/TYPE=PLMN
To: +33111223344/TYPE=PLMN
Subject: A stay in Velen.
Content-type: application/vnd.wap.multipart.related
    start = <0000>
    type = “application/smil”
```

Contents of the message (partially represented in Figure 6.9) comprise a SMIL scene description (Content-ID: <0000>). This scene description represents two slides, each of them containing an image, an AMR sound, and a short text. The size of the entire MMS PDU is 25 KB and a binary file containing this PDU is available from this book’s companion website.

6.2.2 *Message Notification*

Once a message submission has been accepted by the originator MMSC, the originator MMSC analyses message recipient addresses and identifies corresponding recipient MMSCs. The message is routed forward to all recipient MMSCs. Note that if originator and recipient MMS clients are attached to the same MMSE, then the MMSC plays the role of both originator and recipient MMSCs. Upon receipt of the message, the recipient MMSC builds a compact notification describing the message (class, size, priority, etc.) and delivers it to the recipient MMS client. With the notification, the recipient MMS client can retrieve the

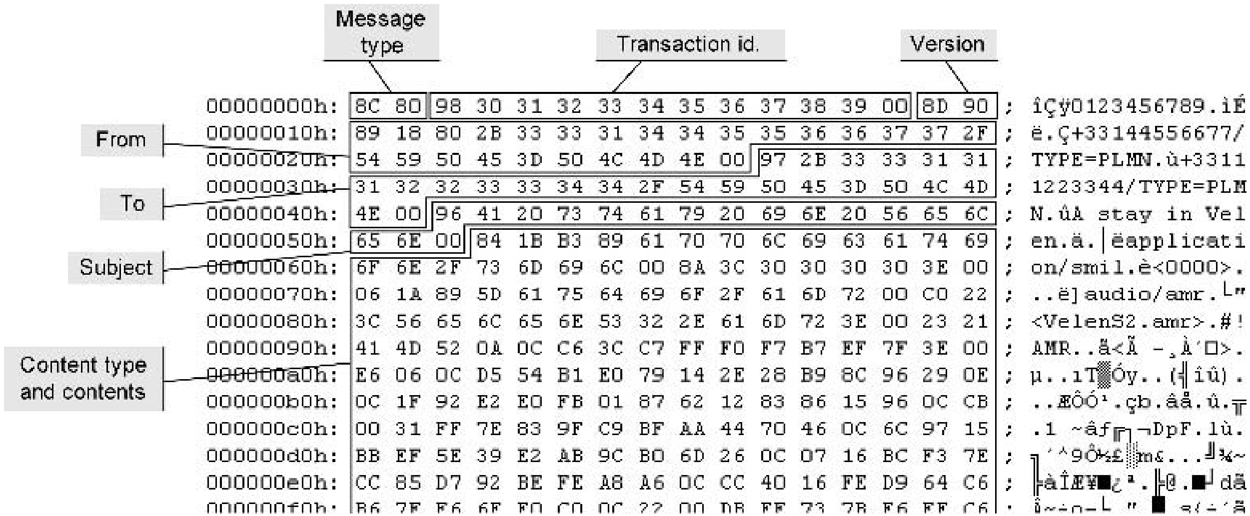


Figure 6.9 Hexadecimal trace dump for submission request (`M-send.req`)

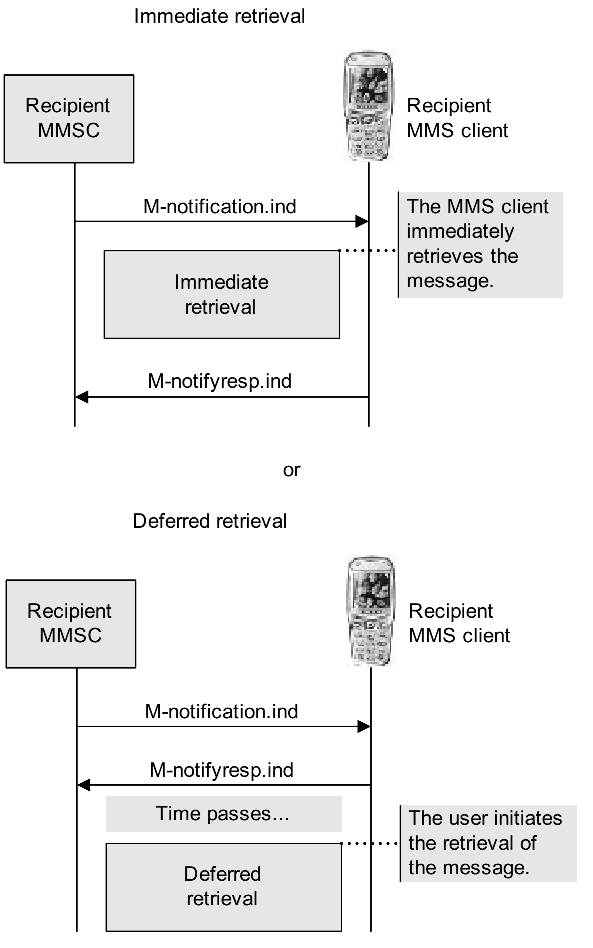


Figure 6.10 MM1 notification transaction flow

corresponding message immediately upon receipt of the notification (immediate message retrieval) or later at the user's own convenience (deferred message retrieval). Figure 6.10 shows the transaction flow for the delivery of a notification from the recipient MMSC to the recipient MMS client.

Parameters composing the notification indication PDU (`M-notification.ind`) are presented in Table 6.4. The notification indication PDU does not contain any body (header only).

Upon receipt of the notification, the MMS client can perform the following actions:

- *Message rejection*: the message is rejected without being retrieved by the recipient MMS client.
- *Immediate message retrieval*: the message is immediately retrieved without user explicit request.

Table 6.4 MM1 notification indication (`M-notification.ind`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-notification-ind`	1.0	●
`X-Mms-Transaction-ID`	Unique identifier for the notification transaction.	1.0	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.0, 1.1, 1.2, or 1.3.	1.0	●
`From`	Address of the originator MMS client (phone number or Email address). This parameter is not present in the notification if the originator requested address hiding.	1.0	○
`Subject`	A short textual description for the message.	1.0	○
`X-Mms-Message-Class`	Message class such as "auto" (automatically generated by the MMS client), "personal" (default), "advertisement," and "informational." Other classes can also be defined in the form of text strings.	1.0	●
`X-Mms-Expiry`	Expiry date (in relative format).	1.0	●
`X-Mms-Priority`	Priority such as "low," "normal" (default), or "high."	1.2	○
`X-Mms-Delivery-Report`	Request for a delivery report. This parameter indicates whether or not delivery reports are to be generated for the message. Two values can be assigned to this parameter: "yes" (delivery report is to be generated) or "no" (no delivery report requested). If the message class is "auto," then this parameter is present in the submission PDU and is set to "no."	1.0	○
`X-Mms-Reply-Charging`	Request for reply charging. The presence of this parameter indicates that reply charging is requested by the message originator. Two values can be assigned to this parameter: "requested" when the originator is willing to pay for the message reply(s) or "requested text only" when the originator is willing to pay for message reply(s) containing text only. In any case, two parameters (reply message size and reply deadline) specify conditions for the message reply to be paid for by the originator.	1.1	○
`X-Mms-Reply-Charging-Deadline`	Reply charging–deadline. This parameter specifies the latest time for the recipient(s) to submit a message reply. This parameter is only present in the PDU if reply charging is requested.	1.1	○
`X-Mms-Reply-Charging-Size`	Reply charging–maximum message size. This parameter specifies the maximum size for message replies. This parameter is only present in the PDU if reply charging is requested.	1.1	○
`X-Mms-Reply-Charging-ID`	Reply charging–identification. This parameter is only inserted in a notification corresponding to a reply message and refers to the original message identifier (`Message-ID` parameter).	1.1	○

(*Continued*)

Table 6.4 (*Continued*)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Size`	Approximate size of the message.	1.0	●
`X-Mms-Stored`	MMBox message availability–the value "yes" assigned to this parameter indicates that the message is stored in the MMBox and is referenced by the value assigned to the `X-Mms-Content-Location` parameter.	1.2	○
`X-Mms-Distribution-Indicator`	Message distribution indicator. This parameter can be present in the notification when the message is sent by a value-added service provider. The value "no" indicates to the recipient that the originator requested the content of the message not to be distributed further.	1.2	○
`X-Mms-Element-Descriptor`	This parameter provides a description of the message main structure (e.g., multipart related, mixed, etc.).	1.2	○
`X-Mms-Content-Location`	Location of the message. This reference can be used by the MMS client for subsequent requests related to the corresponding message.	1.0	●

- *Indication message awaiting retrieval*: the MMS client notifies the user that a message awaits retrieval. In this mode, the user can retrieve the message later at his/her own convenience (deferred message retrieval). With the notification, the user also has the possibility of forwarding the message to other recipients without having to retrieve the message first (from MMS 1.1 only). The user can also instruct the MMS client to delete the notification. In this situation, the message remains in the MMSC message store until its validity period expires and will never be retrieved by the MMS client.

An important parameter of the notification indication PDU is the `X-Mms-Content-Location` parameter. The value assigned to this parameter indicates the location from which the message can be retrieved. An example of the message location that can be assigned to this parameter is:

```
http://mmsc.operator.net/message-id-634
```

The protocol scheme, `http` or `https`, indicates whether or not a secure connection should be established for retrieving the corresponding message as explained in Section 5.24.

The `X-Mms-Message-Size` parameter of the notification indication provides an estimate of the message size. This estimated size may not represent exactly the size of the retrieved message. This is explained by the fact that message contents can be scaled down by the MMSC to meet recipient MMS client rendering capabilities just before the message is retrieved by the recipient MMS client. Furthermore, if the message originates from a value-added service provider, then the message can be replaced by a smaller or larger message between the delivery of the notification and the retrieval of the corresponding message. Consequently, the message size advertised in the notification is usually not used as a reason for automatic message rejection.

A notification indication PDU is conveyed from the recipient MMSC to the recipient MMS client as part of a WAP push ([OMA-WSP], see also Section 1.6.3). For this purpose, the PDU is either encapsulated into one or more SMS messages or, alternatively, conveyed as part of an already established data connection. At the transfer layer of the SMS protocol [3GPP-23.040], an SMS message is structured as a sequence of transfer protocol parameters (parameter names are prefixed with `TP`). For conveying an MMS notification, these parameters are assigned the following values:

`TP-Protocol-Identifier: 0x00`
`TP-Data-Coding-Scheme`: 8-bit data
`TP-User-Data`: as shown in Figure 6.11.

An MMS push (notification or report) always has a push-application identification of 0x04 in the WSP header [OMA-MMS-Enc]. The notification indication PDU is binary encoded according to the rules described in Section 6.2.11. Figure 6.12 shows a hexadecimal trace dump for a notification indication PDU with the following characteristics:

```
X-Mms-Message-Type: M-notification.ind
X-Mms-Transaction-ID: Transaction-123456789-abcdefg
X-Mms-MMS-Version: 1.0
From: +336666666666/TYPE=PLMN
X-Mms-Message-Class: Personal
X-Mms-Message-Size: 120
X-Mms-Expiry: 172800
X-Mms-Content-Location:
    http://mmsc.provider-domain.com:8002/message-123456
```

The notification indication shown in Figure 6.12 is 192 bytes long and, therefore, fits into two SMS message segments (concatenated message). The User Data Header (without UDHL) of the first SMS message segment is shown in Table 6.5.

The User Data Header of the second SMS message segment is similar to that of the first SMS message segment. Only the segment index differs (segment index: `0x02`). Consequently, the User Data Header (without UDHL) of the second SMS message is `0x000306020205040B8423F0`.

The recipient MMS client confirms the reception of the notification indication PDU (`M-notification.ind`) with a notification response indication (`M-notifyresp.ind`). Parameters composing the notification response indication PDU are presented in Table 6.6. The notification response indication PDU does not contain any body (header only).

The parameter `X-Mms-Status` of the notification response indication informs on the status of the message. The following values can be assigned to this parameter:

- *Retrieved*: this status code indicates that the message has been successfully retrieved by the recipient MMS client (immediate retrieval).

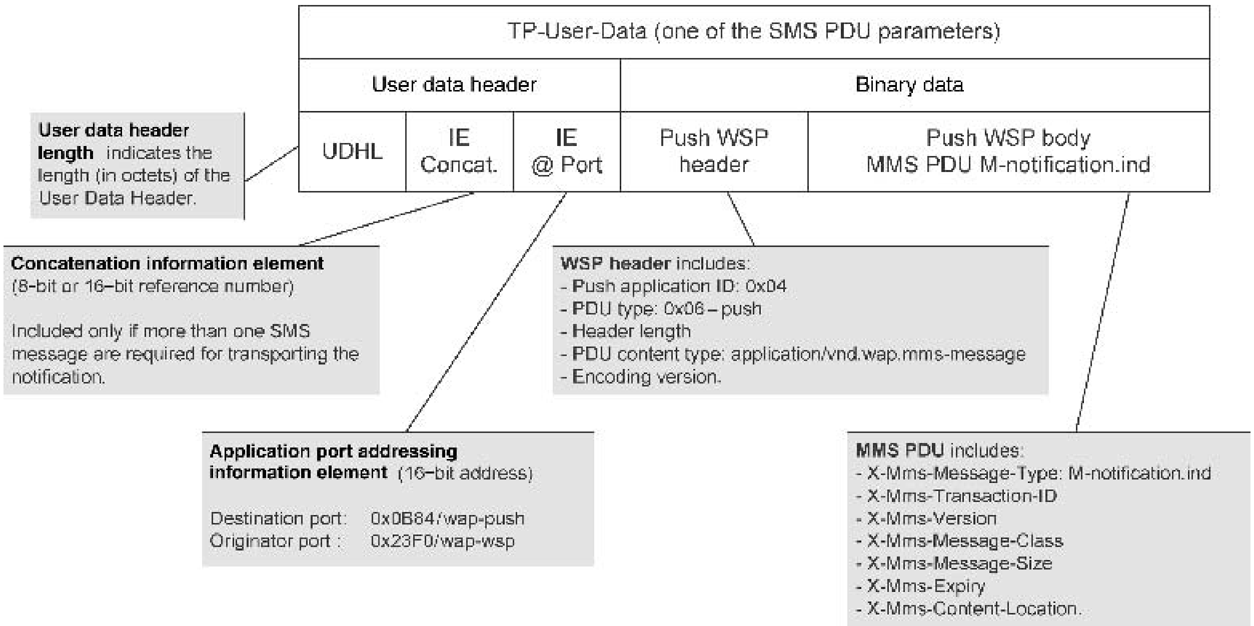


Figure 6.11 Structure of an MMS notification conveyed in an SMS message

WSP Header

```
00000000h: 04 06 37 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 ; ..7application/v
00000010h: 6E 64 2E 77 61 70 2E 6D 6D 73 2D 6D 65 73 73 61 ; nd.wap.mms-messa
00000020h: 67 65 00 45 6E 63 6F 64 69 6E 67 2D 76 65 72 73 ; ge.Encoding-vers
00000030h: 69 6F 6E 00 31 2E 31 00 AF 84 8C 82 98 54 72 61 ; ion.1.1.»âîéÿTra
00000040h: 6E 73 61 63 74 69 6F 6E 2D 49 44 2D 31 32 33 34 ; nsaction-ID-1234
00000050h: 35 36 37 38 39 2D 61 62 63 64 65 66 67 00 8D 90 ; 56789-abcdefg.ìÉ
00000060h: 89 19 80 2B 33 33 36 36 36 36 36 36 36 36 36 36 ; ë.Ç+336666666666
00000070h: 2F 54 59 50 45 3D 50 4C 4D 4E 00 8A 80 8E 01 78 ; /TYPE=PLMN.èÇÄ.x
00000080h: 88 05 81 03 02 A3 00 83 68 74 74 70 3A 2F 2F 6D ; ê.ü..ú.âhttp://m
00000090h: 6D 73 63 2E 70 72 6F 76 69 64 65 72 2D 64 6F 6D ; msc.provider-dom
000000a0h: 61 69 6E 2E 63 6F 6D 3A 38 30 30 32 2F 70 61 74 ; ain.com:8002/pat
000000b0h: 68 2F 6D 65 73 73 61 67 65 2D 31 32 33 34 35 36 ; h/message-123456
```

MMS PDU

Figure 6.12 Hexadecimal trace dump for notification indication (`M-notification.ind`)

Table 6.5 SMS User Data Header of the first SMS message segment

B.E.	Description[1]			
0x00	IEI:	Concatenation 8-bit reference number		(0x00)
0x03	IEDL:	3 bytes (0x03)		
0x06	IED:	Byte 1:	Reference number :	0x06
0x02		Byte 2:	Number of segments:	0x02
0x01		Byte 3:	Segment index:	0x01
0x05	IEI:	Application port addressing scheme, 16-bit address		(0x05)
0x04	IEDL:	4 bytes (0x04)		
0x0B	IED:	Bytes 1...2	Destination port number:	0x0B84
0x84				
0x23		Bytes 3...4	Source port number:	0x23F0
0xF0				

[1] IEI stands for Information Element Identifier, IEDL stands for Information Element Data Length, IED stands for Information Element Data, and B. E. stands for Binary Encoding.

- *Rejected*: this status code indicates that the recipient MMS client rejects the message (e.g., anonymous messages automatically rejected by the recipient MMS client).
- *Deferred*: this status code indicates that the message may be retrieved at a later stage (deferred retrieval).
- *Unrecognized:* this status code is for version management purpose only. For instance, this status code is used by MMS clients if they receive PDUs that are not recognized.

The initial case for element descriptors (parameter `X-Mms-Element-Descriptor`) was to describe, in the notification, media objects contained in a multimedia message to allow a selective retrieval of parts of multimedia messages. However, the latest version of the MMS standard (MMS 1.3, at the time of writing) does not provide support for such selective retrieval. It is, therefore, very unlikely that notifications will contain an element descriptor.

Table 6.6 MM1 notification response indication (`M-notifyresp.ind`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-notifyresp-ind`	1.0	●
`X-Mms-Transaction-ID`	Unique identifier for the notification transaction. The same as the one for the corresponding notification indication.	1.0	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.0, 1.1, 1.2, or 1.3.	1.0	●
`X-Mms-Status`	Status of corresponding message such as "retrieved," "rejected," "deferred," "forwarded," or "unrecognized."	1.0	●
`X-Mms-Report-Allowed`	Indication whether or not the recipient MMS client allows the generation of a delivery report by the recipient MMSC. Possible values are "yes" (default) or "no."	1.0	○

Box 6.1 Is the notification response indication required when notification is conveyed over the SMS bearer?

It has been shown in this section that the MMS notification is conveyed over the SMS bearer or alternatively over a data connection (e.g., GSM data connection or GPRS) if one had previously been established by the MMS client (e.g., message submission or message retrieval). The successful retrieval of the message (immediate retrieval) or the receipt of a positive notification response (deferred retrieval) indicates to the MMSC that the notification indication has been successfully received by the MMS client. Assuming that the notification has been successfully received, the MMSC stops attempting to deliver the notification to the MMS client. Otherwise, the MMSC retransmits the notification after a period of time (period duration not defined in the MMS standards) until the notification is received by the MMS client or until the validity of the corresponding message expires. Theoretically, this behavior should also apply in the situation in which the notification is conveyed over the SMS bearer. Practically, it happens that, in the deferred retrieval case, MMSCs stop attempting to deliver the notification as soon as the SMS message containing the notification has been issued to the device without expecting in return the MMS-level notification response indication from the device. From an operators perspective, this bearer-dependent behavior helps reduce signaling traffic for MMS and this behavior avoids establishing a data connection (of particular interest in the roaming scenario when interoperator contract agreements have not been set). Unfortunately, this deviation from the MMS standards could become the subject of interoperability issues that can lead to message losses.

6.2.3 Message Retrieval

Message retrieval refers to the delivery of a multimedia message from the recipient MMSC to the recipient MMS client. A necessary condition for retrieving a message is that the corresponding notification has been received by the recipient MMS client. The transaction flow for message retrieval is shown in Figure 6.13. The PDU corresponding to the retrieval request is named `WSP/HTTP GET.req` and the corresponding confirmation is named `M-retrieve.conf`. If message retrieval is successful, then the retrieval confirmation PDU body contains the message. According to the retrieving mode, the MMS client may further acknowledge the message retrieval to the MMSC with a retrieval acknowledgment named `M-acknowledge.ind` (deferred retrieval only). As shown in the preceding chapter, a message can be retrieved according to two distinct modes: immediate and deferred retrievals.

With *immediate retrieval*, the recipient MMS client initiates the message retrieval automatically upon reception of the corresponding notification. The user is usually notified that a new message has been received once the message has been completely retrieved by the MMS client. In the immediate retrieval mode, the MMS client indicates to the MMSC that the notification has been successfully received (with the `M-notifyresp.ind` PDU) only after the corresponding message has been successfully retrieved (with the `M-retrieve.conf` PDU). With this method, the role of the `M-notifyresp.ind` PDU is twofold: first, it informs the MMSC that the notification has been successfully processed and second, that the corresponding message has been correctly retrieved. This explains why, in the immediate

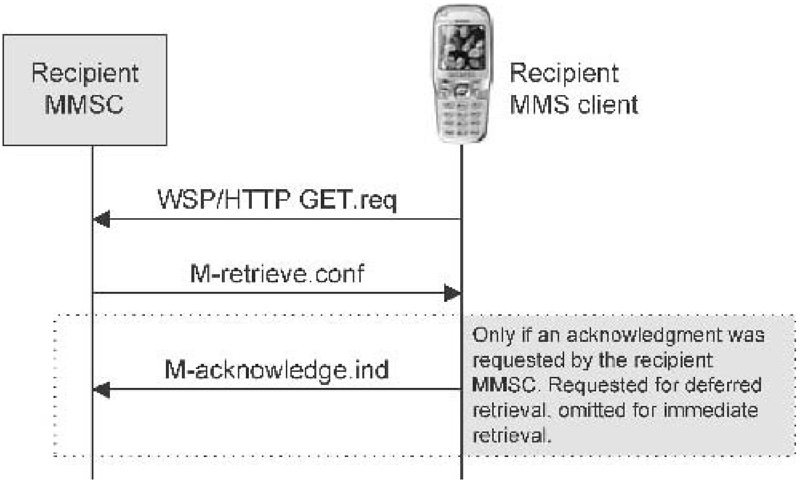


Figure 6.13 MM1 message retrieval transaction flow

retrieval mode, the `M-acknowledge.ind` PDU (used to confirm the message retrieval) is not required. Owing to some transient error conditions, it may happen that message retrieval cannot be performed. In such a situation, the MMSC will retransmit the notification to the MMS client after a given timeout period (dependent on MMSC configuration).

With *deferred retrieval*, the MMS client does not retrieve the message automatically. Instead, the MMS client indicates to the recipient MMSC that the notification has been successfully received and informs the MMSC that the corresponding message may be retrieved at a later stage (e.g., upon user request). At this stage, the user is usually notified that a message awaits retrieval. It becomes the user's responsibility to initiate manually the retrieval of the message. With the deferred retrieval mode, the `M-acknowledge.ind` PDU is used by the MMS client to indicate to the MMSC that the message has been successfully retrieved.

The notification and message retrieval transaction flows are tightly interleaved and differ according to the selected retrieval mode. Figure 6.14 shows the two possible transaction flows.

No dedicated MMS PDU was designed for the message retrieval request. Instead, the existing object retrieval request from the WSP/HTTP protocol is used. This request is named `WSP/HTTP GET.req` and the object that is retrieved is the multimedia message. This message is found at the location provided in the corresponding notification request (`X-Mms-Content-Location` parameter of the `M-notification.ind` PDU). The message is delivered as part of the `M-retrieve.conf` PDU for which parameters are described in Table 6.7. The body of this PDU contains the message being retrieved or an error text if the message cannot be retrieved. Additionally, from MMS 1.1, if the message cannot be delivered by the MMSC, then the message retrieval confirmation PDU indicates an error code (parameter `X-Mms-Retrieve-Status`, see Appendix G). Once the message has been successfully retrieved, the MMS client acknowledges the successful retrieval with the `M-acknowledge.ind` PDU, only if requested by the MMSC (deferred retrieval only). The parameters of the retrieval acknowledgment PDU are described in Table 6.8.

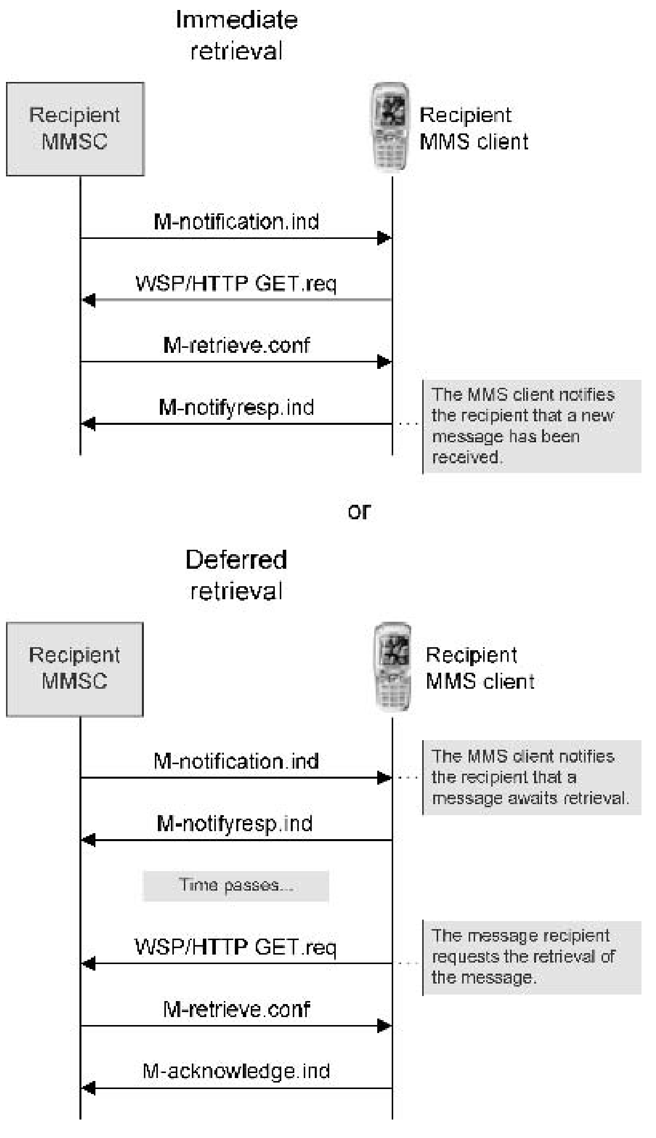


Figure 6.14 MM1 immediate and deferred retrieval transaction flows

From MMS 1.1, the MMSC may indicate the reason for retrieval failure by assigning appropriate values to the parameters `X-Mms-Retrieve-Status` and `X-Mms-Retrieve-Text`. Predefined status codes for the parameter `X-Mms-Retrieve-Status` are listed in Appendix G. The textual description assigned to the `X-Mms-Retrieve-Text` parameter can appropriately be based on codenames defined in [RFC-1893]. In addition, the MMSC also provides an error text as part of the message body, when the `X-Mms-Retrieve-Status` parameter indicates an error, to ensure backward compatibility with MMS 1.0 clients.

Table 6.7 MM1 message retrieval response (`M-retrieve.conf`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-retrieve-conf`	1.0	●
`X-Mms-Transaction-ID`	Unique identifier for the retrieval transaction. This unique identifier is provided by the MMSC only if message retrieval is to be acknowledged with the `M-acknowledge.ind` PDU (deferred retrieval only).	1.0	○
`X-Mms-MMS-Version`	MMS protocol version such as 1.0, 1.1, 1.2, or 1.3.	1.0	●
`Message-ID`	Message unique identifier. This identifier is always provided if the PDU contains the message. It allows the MMS client to correlate read reports and reply messages (reply charging) with the original message.	1.0	© [1]
`Date`	Date and time of latest submission or forwarding of the message (or reception by the MMSC).	1.0	●
`From`	Address of the originator MMS client (phone number or Email address). The "insert token" is not allowed for this PDU. This parameter is not present in the message if the originator requested address hiding.	1.0	○
`X-Mms-Previously-Sent-By`	Address(es) of the MMS client(s) that have previously handled the message (submission or forward).	1.1	○
`X-Mms-Previously-Sent-Date`	Date and time when the message was previously handled by MMS clients (submission or forward).	1.1	○
`To`	One or multiple addresses (phone number or Email address) for message recipient(s). Primary recipients.	1.0	○
`Cc`	One or multiple addresses (phone number or Email address) for message recipient(s). Secondary recipients.	1.0	○
`Subject`	A short textual description for the message.	1.0	○
`X-Mms-Message-Class`	Message class such as "auto" (automatically generated by the originator MMS client), "personal" (default), "advertisement," and "informational." Other classes can also be defined in the form of text strings.	1.0	○
`X-Mms-Priority`	Priority such as "low," "normal" (default), or "high."	1.0	○
`X-Mms-Delivery-Report`	Request for a delivery report. This parameter indicates whether or not delivery reports are to be generated for the message. Two values can be assigned to this parameter: "yes" (delivery report is to be generated) or "no" (no delivery report requested).	1.0	○
`X-Mms-Read-Report`	Request for a read report. This parameter indicates whether or not read reports are to be generated for the message. Two values can be assigned to this parameter: "yes" (read report is to be generated) or "no" (no read report requested).	1.0	○

(*Continued*)

Table 6.7 (*Continued*)

Parameter name	Description	From OMA	St.
`X-Mms-Reply-Charging`	Request for reply charging. The presence of this parameter indicates that reply charging is requested by the message originator. Two values can be assigned to this parameter: "requested" when the originator is willing to pay for the message reply(s) or "requested text only" when the originator is willing to pay for message reply(s) containing text only. In any case, two parameters (reply message size and reply deadline) specify conditions for the message reply to be paid for by the originator.	1.1	○
`X-Mms-Reply-Charging-Deadline`	Reply charging–deadline. This parameter specifies the latest time for the recipient(s) to submit a message reply. This parameter is only present in the PDU if reply charging is requested.	1.1	○
`X-Mms-Reply-Charging-Size`	Reply charging–maximum message size. This parameter specifies the maximum size for message replies. This parameter is only present in the PDU if reply charging is requested.	1.1	○
`X-Mms-Reply-Charging-ID`	Reply charging–identifier. This parameter is inserted in a reply message only and refers to the original message identifier (`Message-ID` parameter).	1.1	○
`X-Mms-Retrieve-Status`	Status code for the retrieval transaction. The retrieval request can be accepted or rejected (permanent or transient errors). See error codes in Appendix G.	1.1	○
`X-Mms-Retrieve-Text`	Description of the transaction status. Descriptions can appropriately be based on the status code names from [RFC-1893].	1.1	○
`X-Mms-MM-State`	MMBox message state. This parameter only appears for a message retrieved from an MMBox and indicates the message state (draft, sent, new, retrieved, or forwarded).	1.2	○
`X-Mms-MM-Flags`	MMBox message flags. This parameter only appears for a message retrieved from an MMBox and indicates the lists of flags associated to the message. The `X-Mms-Flags` parameter appears multiple times if the message is associated to several flags.	1.2	○
`X-Mms-Distribution-Indicator`	Message distribution indicator. This parameter can be present when the message is sent by a value-added service provider. The value "no" indicates to the recipient that the originator requested the content of the message not to be distributed further.	1.2	○
`Content-Type`	Content type of the multimedia message.	1.0	●

[1]In MMS 1.0, the `Message-ID` parameter is optional and not conditional.

Table 6.8 MM1 message retrieval acknowledgement (`M-acknowledge.ind`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-acknowledge-ind`	1.0	●
`X-Mms-Transaction-ID`	Unique identifier for the retrieval transaction. The same as the one for the corresponding retrieval confirmation.	1.0	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.0, 1.1, 1.2, or 1.3.	1.0	●
`X-Mms-Report-Allowed`	Indication whether or not the recipient MMS client allows the generation of a delivery report by the recipient MMSC. Possible values are "yes" (default) or "no."	1.0	○

Note that for the retrieval acknowledgment PDU, the transaction identifier is the same as the one provided by the recipient MMSC for the corresponding retrieval confirmation PDU (`M-retrieve.conf` PDU). Depending on MMSC implementations, the transaction identifier may also be the same as the one provided for the corresponding notification indication PDU (`M-notification.ind` PDU).

The MMSC may indicate that a secure transfer protocol is to be used by the MMS client for the retrieval of the message by indicating so in the corresponding message notification. For instance, this can be done by using the protocol scheme `https` as part of the `X-Mms-Content-Location` parameter of the corresponding notification (e.g., `http://mmsc.operator.net/msgid-6543210`). See also Section 5.24.

6.2.4 Delivery Report

During message submission, the originator MMS client has the possibility of requesting the generation of *delivery report*(s) for the submitted message. If such a request has been made, a delivery report may be generated for each one of the message recipients on the occurrence of the following events:

- Message has been successfully retrieved by the recipient MMS client.
- Message has been deleted (e.g., validity period has expired).
- Message has been rejected by the recipient MMS client.
- Message has been forwarded by the recipient MMS client.
- Message status is indeterminate (e.g., message has been transferred to a messaging system where the concept of delivery report is not fully supported).

Note that a recipient MMS client can deny the generation of delivery reports (parameter `X-Mms-Report-Allowed` for `M-acknowledge.ind` and `M-notifyresp.ind` PDUs). If allowed by the recipient MMS client, the recipient MMSC generates the delivery report and forwards it to the originator MMSC over the MM4 interface. Upon receipt of the delivery report, the originator MMSC delivers it to the originator MMS client over the MM1 interface with the `M-delivery.ind` PDU as shown in Figure 6.15.

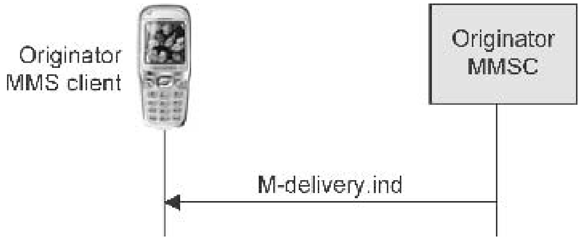


Figure 6.15 MM1 delivery report transaction flow

Parameters of the `M-delivery.ind` PDU are described in Table 6.9.

Delivery and read reports do not contain a transaction identifier since reports are never acknowledged over the MM1 interface. Figure 6.16 shows a hexadecimal trace dump for a delivery report with the following characteristics:

```
X-Mms-Message-Type: M-delivery.ind
X-Mms-MMS-Version: 1.0
Message-ID: Message-12345
To: +33666666666/TYPE=PLMN
Date: Thu, 23 Apr 1998 13:41:37 GMT
X-Mms-Status: Retrieved
```

The delivery report indication shown in Figure 6.16 is 109 bytes long and, therefore, fits into one single SMS message (no need for SMS message concatenation). The User Data Header (without UDHL) of the SMS message is shown in Table 6.10.

Table 6.9 MM1 delivery report indication (`M-delivery.ind`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-delivery-ind`	1.0	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.0, 1.1, 1.2, or 1.3.	1.0	●
`Message-ID`	Identifier of the message to which the report relates. This identifier is used for correlating the delivery report with the original message.	1.0	●
`To`	Address of the message recipient.	1.0	●
`Date`	Date and time the message was retrieved, has expired, etc.	1.0	●
`X-Mms-Status`	Status of corresponding message such as "expired," "retrieved," "rejected," "forwarded," or "indeterminate."	1.0	●

WSP header

```
00000000h: 04 06 37 61 70 70 6C 69 63 61 74 69 6F 6E 2F 76 ; ..7application/v
00000010h: 6E 64 2E 77 61 70 2E 6D 6D 73 2D 6D 65 73 73 61 ; nd.wap.mms-messa
00000020h: 67 65 00 45 6E 63 6F 64 69 6E 67 2D 76 65 72 73 ; ge.Encoding-vers
00000030h: 69 6F 6E 00 31 2E 31 00 AF 84 8C 86 8D 90 8B 4D ; ion.1.1.»äîåiÉïM
00000040h: 65 73 73 61 67 65 2D 31 32 33 34 35 00 97 2B 33 ; essage-12345.ù+3
00000050h: 33 36 36 36 36 36 36 36 36 36 2F 54 59 50 45 2B ; 3666666666/TYPE+
00000060h: 50 4C 4D 4E 00 85 04 35 3F 45 11 95 81          ; PLMN.à.5?E.òü
```

MMS PDU

Figure 6.16 Hexadecimal trace dump for delivery report

Table 6.10 SMS User Data Header of the first message segment

B.E.	Description[1]		
0x05	IEI:	Application port addressing scheme, 16-bit address (0x05)	
0x04	IEDL:	4 bytes (0x04)	
0x0B 0x84	IED:	Bytes 1...2	Destination port number: 0x0B84
0x23 0xF0		Bytes 3...4	Source port number: 0x23F0

[1] IEI stands for Information Element Identifier, IEDL stands for Information Element Data Length, IED stands for Information Element Data, and B.E. stands for Binary Encoding.

6.2.5 Read Report

During message submission, the originator MMS client has the possibility of requesting the generation of *read report*(s) for the submitted message. If such a request has been made, a read report may be generated for each one of the message recipients on the occurrence of the following events:

- Message has been read.
- Message has been deleted without being read.

Unlike delivery reports, it is usually the responsibility of the recipient MMS client to generate the read report when the message has been read by the recipient or deleted without being read. The recipient may deny the generation of a read report (MMS client setting/ according to device features). The MMS client can generate the read report according to the following two different methods:

- The first method, introduced in MMS 1.0, lets the MMS client submit a normal message addressed to the message originator with the `M-send.req` PDU over the MM1 interface. In this case, the message class is set to "auto" and requests for read and delivery reports are disabled. With this method, it is up to the recipient MMS client to include appropriate message content in the message submission PDU body indicating the status of the corresponding message (along with original message subject and message identifier). This first method does not allow the originator MMS client to easily match the received read report with the corresponding submitted message.
- The second method, applicable from MMS 1.1, consists in using a set of two MM1 PDUs dedicated to read reports as shown in Figure 6.17. This method relies on the following three successive steps:

  1. The recipient MMS client provides the read report to the recipient MMSC over the MM1 interface as part of an `M-read-rec.ind` PDU.
  2. Upon receipt of the read report, the recipient MMSC routes forward the read report to the originator MMSC over the MM4 interface.
  3. The originator MMSC provides the read report to the originator MMS client over the MM1 interface as part of an `M-read-orig.ind` PDU.

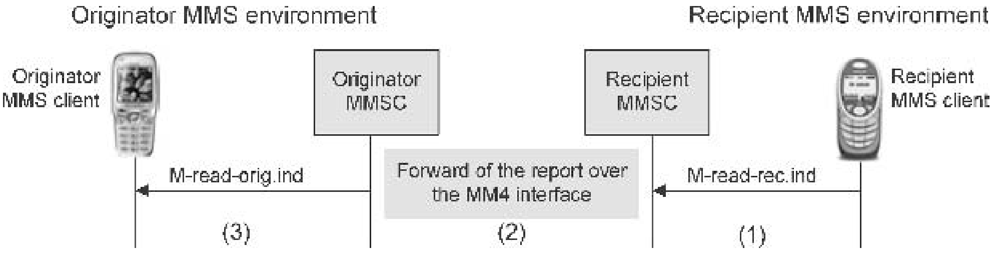


Figure 6.17 MM1 read report transaction flow

Parameters of the `M-read-rec.ind` and `M-read-orig.ind` PDUs are described in Table 6.11.

If the optional `Date` parameter is not present in the `M-read-rec.ind` PDU, then it becomes the responsibility of the recipient MMSC to update the read report with the appropriate date.

There are potential interoperability issues in an environment where communicating MMS entities comply with different methods for the management of read reports. The main risk is for the configuration in which the recipient MMS client and recipient/originator MMSCs comply with MMS 1.1 (or later) and the originator MMSC only complies with MMS 1.0. In this configuration, the originator MMS client may receive a dedicated read-report PDU that it cannot understand. The originator MMSC plays an important role in solving this interoperability issue. When the originator MMSC (complying with MMS 1.1 or later) becomes

Table 6.11 MM1 read-report indications (`M-read-rec.ind` and `M-read-orig.ind`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-read-rec-ind` or : `M-read-orig-ind`	1.1	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.1, 1.2, or 1.3. These PDUs are not defined for MMS 1.0.	1.1	●
`Message-ID`	Identifier of the message to which the report relates. This identifier is used for correlating the read report with the original message.	1.1	●
`To`	Address of the recipient of the read report (message originator).	1.1	●
`From`	Address of the sender of the read report (message recipient). For the `M-read-rec.ind` PDU, an "insert-token" can be used if the sender address is to be provided by the recipient MMSC.	1.1	●
`Date`	Date and time the message was read or was deleted without being read. This parameter is optional for the `M-read-rec.ind` PDU and mandatory for the `M-read-orig.ind` PDU.	1.1	○●
`X-Mms-Read-Status`	Status of corresponding message such as "read" or "deleted without being read."	1.1	●

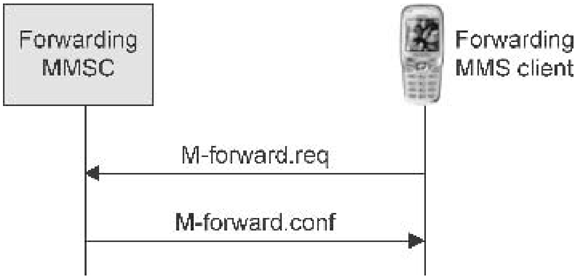


Figure 6.18 MM1 forward transaction flow

aware that the originator MMS client complies with MMS 1.0 only, then it can appropriately convert the dedicated read report into a normal message. The MMSC can identify that the originating MMS client complies with MMS 1.0 only, by means of the User Agent Profile (UAProf) mechanism (MMS version parameter, as described in Section 5.21). Alternatively, the MMS client generating an `M-notifyresp.ind` PDU containing a status value set to "`Unrecognised`" upon receipt of dedicated read report (see Section 6.14) means that the MMS client complies with MMS 1.0 only.

6.2.6 *Message Forward*

Once a notification has been received by the recipient MMS client, the user has the possibility to retrieve or reject the corresponding message. In addition, from MMS 1.1, the recipient MMS client may also support the *forward* of a message that has not yet been retrieved from the MMSC. For instance, a user may wish to forward a large message to an Email server for later viewing from a personal computer. The transaction flow between the MMS client and the recipient MMSC is shown in Figure 6.18. The PDU corresponding to the forward request is named `M-forward.req` and the corresponding confirmation is named `M-forward.conf`.

Parameters of the `M-forward.req` PDU are described in Table 6.12, whereas parameters of the `M-forward.conf` PDU are described in Table 6.13.

Recipient addressing parameters (`To`, `Cc`, and `Bcc`) are all optional in the forward request. However, at least one recipient address shall be provided by the forwarding MMS client to the MMSC.

If the concept of MMBox is not supported by the originator MMSC, then the forward request MMBox-related parameters (`X-Mms-Store`, `X-Mms-MM-State`, and `X-Mms-MM-Flags`) are ignored by the MMSC. When accepting the forwarding request, the MMSC assigns the address of the forwarding MMS client to the `From` parameter of the forwarded message. Optionally, the MMSC assigns the previous address assigned to the `From` parameter to a new instance of the `X-Mms-Previously-Sent-By` parameter and associates a sequence number to this parameter. This sequence number is an increment of the highest sequence number of `X-Mms-Previously-Sent-By` parameters already present in the forwarded message. If there is no `X-Mms-Previously-Sent-By` parameter present in the forwarded message, then the new `X-Mms-Previously-Sent-By` parameter is associated to the sequence number "0" (identifying the original

Table 6.12 MM1 forward request (`M-forward.req`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-forward-req`	1.1	●
`X-Mms-Transaction-ID`	Unique identifier for the forward transaction.	1.1	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.1, 1.2, or 1.3. This PDU is not defined for MMS 1.0.	1.1	●
`Date`	Date and time of latest submission or of forwarding of the message (or reception by the MMSC).	1.1	○
`From`	Address of the forwarding MMS client (phone number or Email address) or "insert token" if address is to be provided by the MMSC.	1.1	●
`To`	One or multiple addresses (phone number or email address) for recipient(s) of the forwarded message. Primary recipients.	1.1	○
`Cc`	One or multiple addresses (phone number or Email address) for recipient(s) of the forwarded message. Secondary recipients.	1.1	○
`Bcc`	One or multiple addresses (phone number or Email address) for recipient(s) of the forwarded message. Secondary recipients/blind copy.	1.1	○
`X-Mms-Expiry`	Expiry date. Default value for this parameter is "maximum."	1.1	○
`X-Mms-Delivery-Time`	Earliest delivery time. Default value for this parameter is "immediate delivery."	1.1	○
`X-Mms-Report-Allowed`	Indication whether or not the forwarding MMS client allows the generation of a delivery report by the forwarding MMSC. Possible values are "yes" (default) or "no."	1.1	○
`X-Mms-Delivery-Report`	Request for a delivery report. This parameter indicates whether or not delivery reports are to be generated for the forwarded message. Two values can be assigned to this parameter: "yes" (delivery report is requested) or "no" (no delivery report requested).	1.1	○
`X-Mms-Read-Report`	Request for a read report. This parameter indicates whether or not read reports are to be generated for the forwarded message. Two values can be assigned to this parameter: "yes" (read report is requested) or "no" (no read report requested).	1.1	○
`X-Mms-Store`	MMBox storage request. This parameter indicates whether the originator MMS client requests to save the message in the originator's MMBox in addition to sending it.	1.2	○
`X-Mms-MM-State`	MMBox message state. If `X-Mms-Store` is set to "yes" or when `X-Mms-Store` is not present and `X-Mms-Content-Location` refers to a message stored in the MMBox, then this parameter indicates the message state to be set in the originator's MMBox. When `X-Mms-Store` is set to "yes" and if this parameter is not present then the message default state in the MMBox is set to "forwarded."	1.2	○

(Continued)

Table 6.12 (*Continued*)

Parameter name	Description	From OMA	St.
X-Mms-MM-Flags	MMBox message flags. This parameter indicates the list of flags associated to a message stored in the MMBox (considered only if X-Mms-Store is set to "yes").	1.2	○
X-Mms-Content-Location	Location of the message to be forwarded (as specified in the corresponding notification).	1.1	○

sender of the message). The example below shows a list of X-Mms-Previously-Sent-By parameters identifying the original sender of the message (number "0," gwenael@lebodic.net) and users who forwarded the message (numbers "1" and "2"):

```
X-Mms-Previously-Sent-By: 0, Gwenael <gwenael@lebodic.net>
X-Mms-Previously-Sent-By: 1, +33612345678/TYPE=PLMN
X-Mms-Previously-Sent-By: 2, +33698765432/TYPE=PLMN
```

Additionally, the MMSC can insert the date and time contained in the Date parameter of the message to be forwarded into a new parameter X-Mms-Previously-Sent-Date and assign a sequence number to that parameter. In that case, the sequence number is the same as the sequence number of the corresponding X-Mms-Previously-Sent-By header field. The MMSC provides a corresponding X-Mms-Previously-Sent-By parameter for each X-Mms-Previously-Sent-Date parameter.

The MMSC may forbid the forwarding of a message containing protected media objects (see Section 5.31).

6.2.7 Storing and Updating a Message in the MMBox

From MMS 1.2, the MMS client has the possibility to store or update a message in an MMBox (only if the concept of MMBox is supported by the MMSC):

- *Store message in MMBox*: the MMS client instructs the MMSC to store a message, which is yet to be retrieved, in the MMBox.
- *Update message in MMBox*: the MMS client instructs the MMSC to update the state and/or flags of a message in the MMBox.

The transaction flow for storing or updating a message in the MMBox is shown in Figure 6.19. The storing/updating request is named M-Mbox-store.req and is acknowledged with the M-Mbox-store.conf confirmation.

Parameters of the M-Mbox-store.req PDU are described in Table 6.14 and parameters of the M-Mbox-store.conf PDU are described in Table 6.15.

The MMSC determines the nature (store or update) of the request by analyzing the X-Mms-Content-Location parameter of the request:

Table 6.13 MM1 message forward confirmation (`M-forward.conf`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-forward-conf`	1.1	●
`X-Mms-Transaction-ID`	Unique identifier for the forward transaction. The same as the one for the corresponding forward request.	1.1	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.1, 1.2, or 1.3.	1.1	●
`X-Mms-Response-Status`	Status code for the forward transaction. The submission request can be accepted or rejected (permanent or transient errors). See status codes in Appendix F.	1.1	●
`X-Mms-Response-Text`	Human readable description of the transaction status.	1.1	○
`Message-ID`	Message unique identifier. This identifier is always provided by the MMSC if the forward request is accepted.	1.1	○
`X-Mms-Content-Location`	Reference to the message stored in the MMBox. This parameter is present only if the three following conditions are fulfilled: - the originator MMSC supports the MMBox feature - the `X-Mms-Store` parameter was present in the corresponding forward request - the `X-Mms-Store-Status` indicates "success." When available, this parameter provides a reference to the message stored in the MMBox (reference used later for message retrieval or view request).	1.2	○
`X-Mms-Store-Status`	MMBox store request status. This parameter is present only if the two following conditions are fulfilled: - the originator MMSC supports the MMBox feature - the `X-Mms-Store` parameter was present in the corresponding forward request. When available, this parameter indicates whether or not the forwarded message has been successfully stored in the MMBox. See status codes in Appendix H.	1.2	○
`X-Mms-Store-Status-Text`	MMBox message textual status. Textual description qualifying the value assigned to the `X-Mms-Store-Status` parameter.	1.2	○

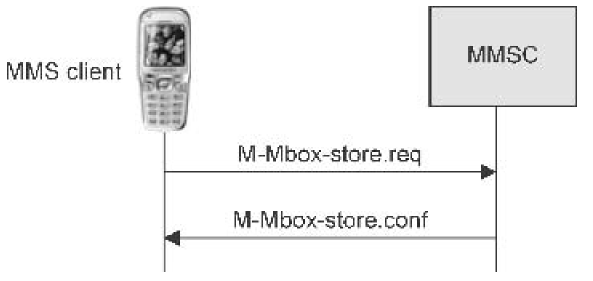


Figure 6.19 MM1 MMBox store/update transaction flow

- If the `X-Mms-Content-Location` refers to a message that is not yet in the user's MMBox, then the MMSC moves the message to the MMBox. The message is associated with the state and flags optionally set in the request. If no state was specified, then the message state is set to "new" by default.
- If the `X-Mms-Content-Location` refers to a message already stored in the user's MMBox, then the MMSC updates the message with the state and flags specified as part of the request.

If the MMSC has successfully processed the request, then it provides a valid reference as part of the `X-Mms-Content-Location` parameter of the confirmation. This reference can be used subsequently for retrieving the message or for obtaining information related to the message.

Table 6.14 MM1 MMBox store/update request (`M-Mbox-store.req`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-Mbox-store-req`	1.2	●
`X-Mms-Transaction-ID`	Unique identifier for the store/update transaction.	1.2	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.2 or 1.3.	1.2	●
`X-Mms-Content-Location`	Reference of the message to be stored or updated in the MMBox.	1.2	●
`X-Mms-MM-State`	MMBox message state–if `X-Mms-Content-Location` parameter refers to a message to be stored (reference retrieved from the `M-notification.ind` PDU), then the default state for the message state is "new."	1.2	○
`X-Mms-MM-Flags`	MMBox message flags. This parameter indicates the list of flags associated to a message stored in the MMBox (considered only if `X-Mms-Store` is set to "yes"). This field may appear multiple times in the PDU.	1.2	○

Table 6.15 MM1 MMBox store/update confirmation (`M-Mbox-store.conf`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-Mbox-store-conf`	1.2	●
`X-Mms-Transaction-ID`	Unique identifier for the store/update transaction. The same as the one for the corresponding MMbox store/update request.	1.2	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.2 or 1.3.	1.2	●
`X-Mms-Content-Location`	This parameter can be present only if the `X-Mms-Store-Status` parameter is set to "success." When available, this parameter indicates a reference to the message in the MMBox. This reference can be used subsequently for retrieving the message or for obtaining information about the message.	1.2	○
`X-Mms-Store-Status`	MMBox store request status. The status of the store/update request such as "success." See status codes in Appendix H.	1.2	●
`X-Mms-Store-Status-Text`	MMBox message textual status. Textual description qualifying the value assigned to the `X-Mms-Store-Status` parameter.	1.2	○

6.2.8 Viewing Information from the MMBox

From MMS 1.2, the MMS client has the possibility to request information about one or more multimedia messages that are stored in the user's MMBox. The information not only includes message parameters (e.g., class, priority, originator address, etc.) but also the message contents if requested (e.g., media objects and scene description). The transaction flow for requesting such information is shown in Figure 6.20. The information-viewing request is named `M-Mbox-view.req` and is acknowledged with the `M-Mbox-view.conf` confirmation.

The identification of messages in the MMBox is performed according to one of the two following methods:

- Identifying the message with its Uniform Resource Locator (URL) assigned to the parameter `X-Mms-Content-Location` (obtained from a previous notification or store transaction).
- Filtering messages with criteria based on the message MMBox flags and state.

For efficiency, the MMS client can retrieve information for all messages meeting the filtering criteria with several viewing transactions. This helps in reducing the amount of information contained in each single viewing transaction confirmation. For this purpose, the MMS client specifies the subset of messages (start and end of range) to be selected for each single viewing transaction. For instance, if 20 messages meet the filtering criteria, then the MMS client would be able to first retrieve the first 5 messages in one view transaction, the next 5 messages in a second view transaction, and so on. In addition, the MMS client

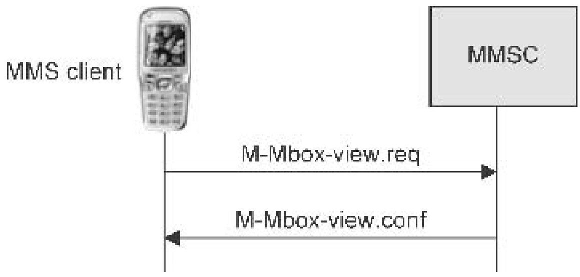


Figure 6.20 MM1 MMBox view transaction flow

indicates, as part of the viewing request, which message parameters/contents are to be returned as part of the viewing confirmation.

Upon reception of the viewing request, the MMSC selects the messages matching the selection criteria in the MMBox. The next step for the MMSC consists of generating the viewing confirmation as a multipart mixed structure for which each body part contains the requested information for one of the selected messages. Additionally, the MMSC provides the MMBox total and quota information as part of the viewing confirmation when requested by the MMS client.

Parameters of the `M-Mbox-view.req` PDU are described in Table 6.16 and parameters of the `M-Mbox-view.conf` PDU are described in Table 6.17. The view request PDU does not contain any body (header only).

If the MMS client view request can be processed successfully, then the body of the confirmation PDU is organized as a multipart structure. For this purpose, the confirmation PDU content type is set to "`application/vnd.wap.multipart.mixed`." Each body part of the structure contains information related to one of the selected messages as shown in Table 6.18.

Figure 6.21 shows how a view confirmation can contain information regarding two selected messages (textual representation).

6.2.9 Uploading a Message to the MMBox

From MMS 1.2, the MMS client has the possibility to upload a message in the MMBox (only if the concept of MMBox is supported by the MMSC). The message may be a new message created by the user (e.g., draft message) or a message previously retrieved from the MMSC. The transaction flow for uploading a message in the MMBox is shown in Figure 6.22. The uploading request is named `M-Mbox-upload.req` and is acknowledged with the `M-Mbox-upload.conf` confirmation.

Parameters of the `M-Mbox-upload.req` PDU are described in Table 6.19 and parameters of the `M-Mbox-upload.conf` PDU are described in Table 6.20. The body of the upload request PDU contains the message to be uploaded in the MMBox. In this context, the uploaded message is described with the parameters presented in Table 6.18. The upload confirmation PDU does not contain any body (header only).

It is recommended to use the `M-MBox-View.req` request (instead of the `WSP/HTTP GET.req`) for retrieving a message that has been uploaded to the user's MMBox with the `M-Mbox-upload.req` request.

Table 6.16 MM1 MMBox view request (`M-Mbox-view.req`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-Mbox-view-req`	1.2	●
`X-Mms-Transaction-ID`	Unique identifier for the view transaction.	1.2	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.2 or 1.3.	1.2	●
`X-Mms-Content-Location`	Reference(s) of the message in the MMBox for which information is requested. This parameter may appear multiple times for obtaining information for several messages.	1.2	○
`X-Mms-MM-State`	MMBox message state. This parameter specifies the state of the message(s) for which information is requested. When this parameter appears multiple times, then the selection is the union of all messages fulfilling at least one of the parameter requirements (e.g., information requested for draft and sent messages/parameter appears twice).	1.2	○
`X-Mms-MM-Flags`	MMBox message flags. This parameter specifies the flag of the message(s) for which information is requested. When this parameter appears multiple times, then the selection is the union of all messages fulfilling at least one of the parameter requirements.	1.2	○
`X-Mms-Start`	Start of selection range. A number indicating the first multimedia message in the selection for which information is returned in the transaction confirmation.	1.2	○
`X-Mms-Limit`	Size of selection range. The number of multimedia messages in the selection for which information is returned in the transaction confirmation.	1.2	○
`X-Mms-Attributes`	Message parameter(s) to be returned. This parameter indicates the list of message parameters (message information) to be included as part of the transaction confirmation. If not present, then an MMSC default set of message parameters is considered. If no default set has been defined, then the set of message parameters composing the notification is used instead.	1.2	○
`X-Mms-Totals`	Request for returning MMBox totals. This parameter indicates whether or not the MMS client requests the total count of messages stored in the user's MMBox to be returned as part of the transaction confirmation. Two values can be assigned to this parameter: "yes" (information on totals is requested) or "no" (no information on totals requested). If not present, then the default value is "no" request.	1.2	○
`X-Mms-Quotas`	Request for returning MMBox quotas. This parameter indicates whether or not the MMS client requests the user's MMBox quotas to be returned as part of the transaction confirmation. Two values can be assigned to this parameter: "yes" (information on quotas is requested) or "no" (no information on quotas requested). If not present, then the default value is "no" request.	1.2	○

Table 6.17 MM1 MMBox view confirmation (`M-Mbox-view.conf`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-Mbox-view-conf`	1.2	●
`X-Mms-Transaction-ID`	Unique identifier for the view transaction. The same as the one for the corresponding MMBox view request.	1.2	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.2 or 1.3.	1.2	●
`X-Mms-Response-Status`	Status code for the view transaction. The view request can be accepted or rejected (permanent or transient errors). See status codes in Appendix F.	1.2	●
`X-Mms-Response-Text`	Human readable description of the transaction status.	1.2	○
`X-Mms-Content-Location`	Copied from the corresponding request–reference(s) of the message in the MMBox for which information is requested. This parameter may appear multiple times for obtaining information for several messages. This parameter appears in the confirmation only if it is present in the corresponding request.	1.2	○
`X-Mms-MM-State`	Copied from the corresponding request–state criteria for selected messages. This parameter may appear several times and appears in the confirmation only if it is present in the corresponding request.	1.2	○
`X-Mms-MM-Flags`	Copied from the corresponding request–flag criteria for selected messages. This parameter may appear several times and appears in the confirmation only if it is present in the corresponding request.	1.2	○
`X-Mms-Start`	Copied from the corresponding request. A number indicating the first multimedia message in the selection for which information is returned. This parameter appears in the confirmation only if it is present in the corresponding request.	1.2	○
`X-Mms-Limit`	Copied from the corresponding request. The number of multimedia messages in the selection for which information is returned. This parameter appears in the confirmation only if it is present in the corresponding request.	1.2	○
`X-Mms-Attributes`	Copied from the corresponding request or specified by the MMSC. This parameter indicates the list of message parameters (message information) included as part of the transaction confirmation.	1.2	○
`X-Mms-Mbox-Totals`	MMBox totals. This parameter indicates the total number of messages or bytes in the user's MMBox. This parameter appears in the confirmation if the parameter `X-Mms-Totals` of the corresponding request is set to "yes." Otherwise it does not appear in the confirmation.	1.2	○
`X-Mms-Mbox-Quotas`	MMBox quotas. This parameter indicates the quotas of the user's MMBox in terms of messages or bytes. This parameter appears in the confirmation if the parameter `X-Mms-Quotas` of the corresponding request is set to "yes." Otherwise it does not appear in the confirmation.	1.2	○

(*Continued*)

Table 6.17 (*Continued*)

Parameter name	Description	From OMA	St.
X-Mms-Message-Count	Number of messages for which information is included in the confirmation.	1.2	○
Content-type	This parameter always appears as the last parameter of the PDU header. If the PDU has no body then the content type "*/*" is used (binary encoded as 0x00). Otherwise, the content type is "application/vnd.wap.multipart.mixed."	1.2	●

6.2.10 Deleting a Message from the MMBox

From MMS 1.2, the MMS client has the possibility to delete one or more messages from the MMBox with a single transaction (only if the concept of MMBox is supported by the MMSC). The transaction flow for deleting one or more messages from the MMBox is shown in Figure 6.23. The delete request is named `M-Mbox-delete.req` and is acknowledged with the `M-Mbox-delete.conf` confirmation.

Parameters of the `M-Mbox-delete.req` PDU are described in Table 6.21 and parameters of the `M-Mbox-delete.conf` PDU are described in Table 6.22. The delete request and confirmation PDUs do not contain any body (header only).

It may happen that the MMSC is unable to delete several or all messages referenced in the deletion request. In order to warn the requesting MMS client, the MMSC includes in the deletion confirmation and the references (`X-Mms-Content-Location` parameter) of messages that could not be deleted along with relevant error codes (`X-Mms-Response-Status` and `X-Mms-Response-Status-Text` parameters).

6.2.11 Parameter Description and Binary Encoding

Protocol data units are binary encoded for being conveyed over the MM1 interface. Each parameter name has an associated assigned number and each parameter value is encoded according to encoding rules specific to the value type (e.g., text string, short integer, etc.). In the context of MMS, the following types are used:

- *Enumeration*: a list of predefined values can be assigned to a parameter of type "enumeration."
- *Short integer*: a value in the range 0...127 (decimal) can be assigned to a parameter of type "short integer." The value is represented with 1 byte for which the most significant bit is set to 1 and the remaining bits represent the integer value.
- *Long integer*: a large integer can be assigned to a parameter of type "long integer." The value is formatted according to the following derivation rule:
  `Long-integer-value = Short-length 1*30 Octet`
  where

Table 6.18 Parameters of the structure containing information related to a viewed message

Parameter name	Description	From OMA	St.
X-Mms-Message-Type	MMS protocol data unit type. Value: M-Mbox-view-desc	1.2	●
X-Mms-Content-Location	Location of the message–this reference can be used by the MMS client for subsequent requests related to the corresponding message. This parameter is only used in conjunction with a view transaction.	1.2	©
Message-ID	Message unique identifier.	1.2	●
X-Mms-MM-State	MMBox message state.	1.2	○
X-Mms-MM-Flags	MMBox message flags. This parameter may appear multiple times.	1.2	○
Date	Date and time of message submission. Note (A).	1.2	○
From	Address of the originator MMS client. Note (A).	1.2	○
To	One or multiple addresses (phone number or Email address) for message recipient(s). Primary recipients. Note (A).	1.2	○
Cc	One or multiple addresses (phone number or Email address) for message recipient(s). Secondary recipients. Note (A).	1.2	○
Bcc	One or multiple addresses (phone number or Email address) for message recipient(s). Secondary recipients/blind copy. Note (A).	1.2	○
X-Mms-Message-Class	Message class. Note (A).	1.2	○
Subject	A short textual description for the message. Note (A).	1.2	○
X-Mms-Priority	Priority such as "low," "normal" (default), or "high." Note (A).	1.2	○
X-Mms-Delivery-Time	Earliest delivery time. Note (A).	1.2	○
X-Mms-Expiry	Expiry time. Note (A).	1.2	○
X-Mms-Delivery-Report	Request for a delivery report. Note (A).	1.2	○
X-Mms-Read-Report	Request for a read report. Note (A).	1.2	○
X-Mms-Message-Size	Approximate size of the message.	1.2	○
X-Mms-Reply-Charging	Request for reply charging. Note (A).	1.2	○
X-Mms-Reply-Charging-ID	Reply charging–identification. This parameter is inserted in a reply message only and refers to the original message identification (Message-ID parameter).	1.2	○
X-Mms-Reply-Charging-Deadline	Reply charging–deadline. Note (A).	1.2	○
X-Mms-Reply-Charging-Size	Reply charging–maximum message size. Note (A).	1.2	○
X-Mms-Previously-Sent-By	Address(es) of the MMS client(s) that have previously handled the message (submission or forward). Note (A).	1.2	○
X-Mms-Previously-Send-Date	Date and time when the message was previously handled by MMS clients (submission or forward). Note (A).	1.2	○
Content-Type	Content type of the multimedia message. Note (A).	1.2	○

Note: (A) In conjunction with a view transaction: the parameter appears only if requested by the MMS client (according to parameter X-Mms-Attributes *of the corresponding view request).*

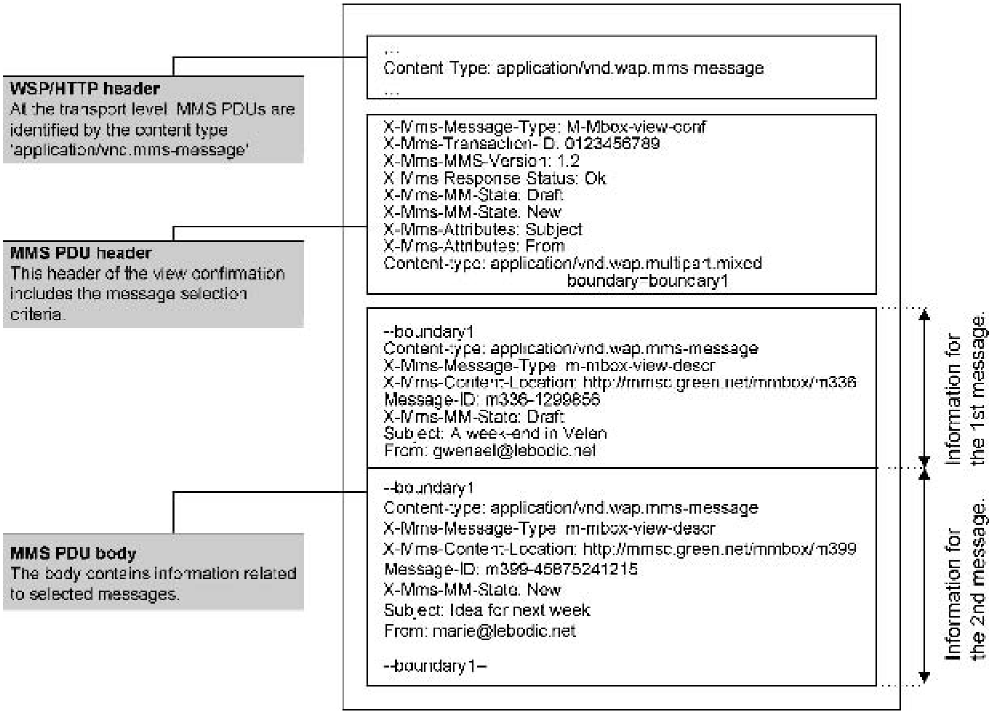


Figure 6.21 View confirmation containing information for two selected messages

- `Short-length` is a 1-byte integer representing the number of following bytes (0···30 bytes).
- `1*30 Octet` is a sequence of up to 30 bytes representing an unsigned integer value with the most significant byte encoded first (big-endian representation).

- *Text string*: values of type "text string" contain only US-ASCII characters. The sequence of characters is null terminated.
- *Encoded string*: values of type "encoded string" contain US-ASCII characters if the character set is not specified, otherwise these strings are encoded according to UTF-8 [RFC-2279]. The string is formatted according to the following derivation rule:

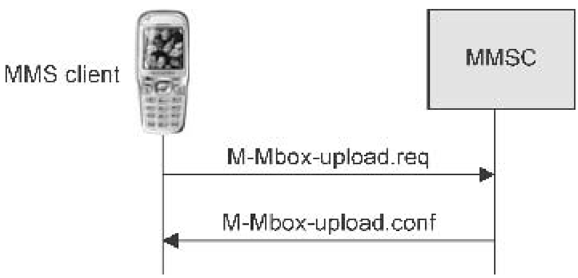


Figure 6.22 MM1 MMBox upload transaction flow

Table 6.19 MM1 MMBox upload request (`M-Mbox-upload.req`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-Mbox-upload-req`	1.2	●
`X-Mms-Transaction-ID`	Unique identifier for the upload transaction.	1.2	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.2 or 1.3.	1.2	●
`X-Mms-MM-State`	MMBox message state. This parameter specifies the state of the uploaded message. If not available, then the default message state is "draft."	1.2	○
`X-Mms-MM-Flags`	MMBox message flags. This parameter indicates the list of flags associated to a message stored in the MMBox (considered only if `X-Mms-Store` is set to "yes"). This field may appear multiple times.	1.2	○
`Content-type`	Content type of the uploaded message.	1.2	●

`Encoded-string-value` = `Text-string` | `Value-length` `Char-set` `Text-string`
where

- `Value-length` is the length of the value in bytes [OMA-WSP]. The value length is either a short length (see long integer derivation rule above) or an integer length spanning over several bytes as defined in [OMA-WSP].

Table 6.20 MM1 MMBox upload confirmation (`M-Mbox-upload.conf`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-Mbox-upload-conf`	1.2	●
`X-Mms-Transaction-ID`	Unique identifier for the upload transaction. The same as the one for the corresponding MMBox upload request.	1.2	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.2 or 1.3.	1.2	●
`X-Mms-Content-Location`	This parameter can be present only if the `X-Mms-Store-Status` is set to "success." When available, this parameter indicates a reference of the message in the MMBox. This reference can be used subsequently for retrieving the message or for obtaining information about the message.	1.2	○
`X-Mms-Store-Status`	MMBox store request status. The status of the store request such as "success."	1.2	●
`X-Mms-Store-Status-Text`	MMBox message textual status. Textual description qualifying the value assigned to the `X-Mms-Store-Status` parameter.	1.2	○

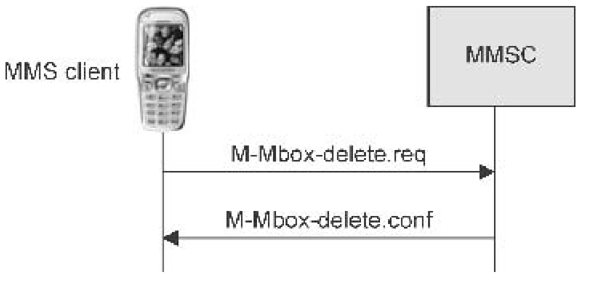


Figure 6.23 MM1 MMBox delete transaction flow

Table 6.21 MM1 MMBox delete request (`M-Mbox-delete.req`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-Mbox-delete-req`	1.2	●
`X-Mms-Transaction-ID`	Unique identifier for the delete transaction.	1.2	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.2 or 1.3.	1.2	●
`X-Mms-Content-Location`	Reference of the message(s) to be deleted. This parameter may appear multiple times if more than one message are to be deleted.	1.2	●

Table 6.22 MM1 MMBox delete confirmation (`M-Mbox-delete.conf`)

Parameter name	Description	From OMA	St.
`X-Mms-Message-Type`	MMS protocol data unit type. Value: `M-Mbox-delete-conf`	1.2	●
`X-Mms-Transaction-ID`	Unique identifier for the delete transaction. The same as the one for the corresponding MMBox delete request.	1.2	●
`X-Mms-MMS-Version`	MMS protocol version such as 1.2 or 1.3.	1.2	●
`X-Mms-Content-Location`	Reference to the message that could not be deleted by the MMSC. This parameter may appear multiple times if more than one message could not be deleted by the MMSC.	1.2	○
`X-Mms-Response-Status`	MMBox deletion request status. The status of the deletion request such as "success." See status codes in Appendix F.	1.2	●
`X-Mms-Response-Status-Text`	MMBox message textual status. Textual description qualifying the value assigned to the `X-Mms-Response-Status` parameter.	1.2	○

- `Char-set` is the character set MIBEnum[1] value registered by IANA.
- `Text-string` is a sequence of null-terminated characters.

For instance, the following hexadecimal sequence of bytes represents the UTF-8 encoded string "Hello:"

0x07	Length of value remaining part (7 bytes–short length)
0xEA	MIBEnum value 0x6A for UTF-8 (106 decimal)
0x48	Character "H"
0x65	Character "e"
0x6C	Character "l"
0x6C	Character "l"
0x6F	Character "o"
0x00	Null character–end of string

- *Absolute or relative date* (GMT): a date is either represented in an absolute or relative format according to the following derivation rule:
  `AR-Date-value = Value-length (Absolute-token Date-value|Relative-token Delta-seconds-value)`
  where
  - `Value-length` is the length of the value in bytes [OMA-WSP].
  - `Absolute-token` is the value 0x80.
  - `Relative-token` is the value 0x81.
  - `Date-value` is a value of type "long integer" representing the number of seconds elapsed since 1970-01-01, 00:00:00 GMT.
  - `Delta-seconds-value` is a value of type "long integer" representing the number of seconds from the reference date and time.

  A date is always conveyed as a GMT date. Such a date usually requires to be converted to a local date before it is provided to the user.

- *Complex structure for From parameter*: values of this type are assigned to the `From` parameter. They are constructed according to the following derivation rule:
  `From-value = Value-length (Address-present-token Encoded-string-value | Insert-address-token)`
  where
  - `Value-length` is the length of the value in bytes.
  - `Address-present-token` is the value 0x80.
  - `Insert-address-token` is the value 0x81.
  - `Encoded-string-value` is a string encoded as shown above.

- *Indexed encoded string*: values of this type are encoded strings prefixed by either a long integer or a short integer (index) as shown by the following derivation rule:

[1] See registered MIBEnum values at http://www.iana.org/assignments/character-sets.

`Indexed-encoded-string-value` = `Value-length Index-value Encoded-string-value`
where

- `Value-length` is the length of the value in bytes.
- `Index-value` is a short or long integer (index) as defined above.
- `Encoded-string-value` is an encoded string as defined above.

• *Indexed date*: values of this type are absolute dates prefixed by either a long or a short integer as shown by the following derivation rule:
`Indexed-date-value` = `Value-length Index-value Encoded-string-value`
where

- `Value-length` is the length of the value in bytes.
- `Index-value` is a short or long integer (index) as defined above.
- `Encoded-date-value` is a value of type "long integer" representing the number of seconds elapsed since 1970-01-01, 00:00:00 GMT.

• *MMBox size*: this type is mainly used for representing MMBox totals and quotas. Values of this type represent a size expressed in terms of number of messages or number of bytes as shown by the following derivation rule:
`Mbox-size-value` = `Value-length (Message-token | Size-token) Integer-Value`
where

- `Value-length` is the length of the value in bytes.
- `Message-token` is the value 0x80 (size is expressed in number of messages).
- `Size-token` is the value 0x81 (size is expressed in number of bytes).
- `Integer-Value` is a short or long integer representing the size (in terms of messages or bytes).

• *Flag command*: a flag command instructs the MMSC to add, remove, or filter a flag for a message stored in the MMBox. A flag command is structured as shown in the following derivation rule:
`Flag-command-value` = `Value-length ( Add-token | Remove-token | Filter-token ) Encoded-string-value`
where

- `Value-length` is the length of the value in bytes.
- `Add-token` is the value 0x80 (command is "add a flag").
- `Remove-token` is the value 0x81 (command is "remove flag").
- `Filter-token` is the value 0x82 (command is "filter flag").
- `Encoded-string-value` is a US-ASCII or UTF-8 string representing the flag.

• *Element descriptor*: a value of type element descriptor describes the main structure of a message (multipart mixed, related, etc.), available from MMS 1.2. Such a value is structured as shown with the following derivation rules:

`Element-Descriptor-value = Value-length Content-Reference-value *(Parameter)`
`Parameter = Parameter-name Parameter-value`
where

- `Value-length` is the length of the value in bytes.
- `Content-Reference-Value` is a text string representing the element reference.
- `Parameter-name` is the name of the parameter as a text string (US-ASCII) or short integer.
- `Parameter-value` is the value assigned to the corresponding parameter.

In encoding of the PDU header parameters, the order of the parameters is not significant, except that `X-Mms-Message-Type`, `X-Mms-Transaction-ID` (when present), and `X-Mms-MMS-Version` parameters must be at the beginning of the PDU header, in that order, and if the PDU contains a multimedia message in the PDU body, then the `Content-Type` parameter must be the last header parameter, followed by message contents.

To guarantee interoperability, OMA recommends maximum lengths for several PDU parameters of the MM1 interface as shown in Table 6.23 [OMA-MMS-Conf].

The list of parameters that can be included in PDUs invoked over the MM1 interface is provided in Table 6.24.

6.3 MM2 Interface, Internal MMSC Interface

The *MM2 interface* is an MMSC internal interface. The interface is required if MMS Relay and MMS Server are provided as two separate entities (e.g., developed by two different manufacturers). However, available commercial implementations usually combine MMS Relay and MMS Server into a single entity, called MMSC. In this situation, the MM2 interface is implemented in a proprietary manner. At the time of writing, no technical realization of the MM2 interface had been specified by standardization organizations.

Table 6.23 Maximum lengths for MM1 parameters

Parameter name	Maximum length (characters)
`Content-ID`	100
`Content-location`	100
`Message-ID`	40
`Subject`	40[1]
`X-Mms-Content-Location`	100
`X-Mms-Response-Text`	30
`X-Mms-Transaction-ID`	40

[1]For the `subject` parameter, 40 characters represent either 40 US-ASCII characters or 40 complex glyphs (e.g., Chinese pictograms).

Table 6.24 MM1 PDU parameters with binary encoding

Parameter name	A.N.	B.E.	Values	Binary encoding
Bcc	0x01	0x81	US-ASCII or UTF-8 string	Type: encoded string
Cc	0x02	0x82	US-ASCII or UTF-8 string	Type: encoded string
Content-type	0x04	0x84	Multipart mixed/related	Type: multipart as defined in [OMA-WSP]
Date	0x05	0x85	In seconds since 1970-01-01, 00:00:00 GMT	Type: long integer
From	0x09	0x89	US-ASCII or UTF-8 string or "insert token."	Type: complex structure for From parameter
Message-ID	0x0B	0x8B	US-ASCII string	Type: text string
Subject	0x16	0x96	US-ASCII or UTF-8 string	Type: encoded string
To	0x17	0x97	US-ASCII or UTF-8 string	Type: encoded string
X-Mms-Attributes	0x28	0xA8		Type: enumeration
			Bcc	0x81
			Cc	0x82
			Content	0xAE
			Content-type	0x84
			Date	0x85
			X-Mms-Delivery-Report	0x86
			X-Mms-Delivery-Time	0x87
			X-Mms-Expiry	0x88
			From	0x09
			X-Mms-Message-Class	0x8A
			Message-ID	0x8B
			X-Mms-Message-Size	0x8E
			X-Mms-Priority	0x8F
			X-Mms-Read-Report	0x90
			Subject	0x96
			To	0x97
			X-Mms-Reply-Charging	0x9C
			X-Mms-Reply-Charging-ID	0x9E
			X-Mms-Reply-Charging-Deadline	0x9D
			X-Mms-Reply-Charging-Size	0x9F
			X-Mms-Previously-Sent-By	0xA0
			X-Mms-Previously-Sent-Date	0xA1
			Additional-headers	0xB0

(*Continued*)

Table 6.24 (*Continued*)

Parameter name	A.N.	B.E.	Values	Binary encoding
X-Mms-Content-Location	0x03	0x83	US-ASCII string	Type: text string
X-Mms-Delivery-Report	0x06	0x86		Type: enumeration
			Yes	0x80
			No	0x81
X-Mms-Delivery-Time	0x07	0x87	Absolute time or relative time from submission time	Type: absolute or relative date
X-Mms-Distribution-Indicator	0x31	0xB1		Type: enumeration
			Yes	0x80
			No	0x81
X-Mms-Element-Descriptor	0x32	0xB2		Type: element descriptor
X-Mms-Expiry	0x08	0x88	Absolute time or relative time from submission time	Type: absolute or relative date
X-Mms-Mbox-Quotas	0x2C	0xAC	Quotas of the MMBox	Type: MMBox size
X-Mms-Mbox-Totals	0x2A	0xAA	Number of messages or bytes stored in the MMBox	Type: MMBox size
X-Mms-Message-Class	0x0A	0x8A	Token text or	Type: enumeration
			Personal	0x80
			Advertisement	0x81
			Informational	0x82
			Auto	0x83
X-Mms-Message-Count	0x2D	0xAD	Number of messages	Type: short or long integer
X-Mms-Message-Size	0x0E	0x8E	Expressed in bytes	Type: long integer
X-Mms-Message-Type	0x0C	0x8C		Type: enumeration
			M-send-req	0x80
			M-send-conf	0x81
			M-notification-ind	0x82
			M-notifyresp-ind	0x83
			M-retrieve-conf	0x84
			M-acknowledge-ind	0x85
			M-delivery-ind	0x86
			M-read-rec-ind	0x87
			M-read-orig-ind	0x88
			M-forward-req	0x89

(*Continued*)

Table 6.24 (*Continued*)

Parameter name	A.N.	B.E.	Values	Binary encoding
			M-forward-conf	0x8A
			M-mbox-store-req	0x8B
			M-mbox-store-conf	0x8C
			M-mbox-view-req	0x8D
			M-mbox-view-conf	0x8E
			M-mbox-upload-req	0x8F
			M-mbox-upload-conf	0x90
			M-mbox-delete-req	0x91
			M-mbox-delete-conf	0x92
			M-mbox-descr	0x93
X-Mms-MM-Flags	0x24	0xA4	Add, remove, or filter flag command	Type: flag command
X-Mms-MM-State	0x23	0xA3		Type: enumeration
			Draft	0x80
			Sent	0x81
			New	0x82
			Retrieved	0x83
			Forwarded	0x84
X-Mms-MMS-Version	0x0D	0x8D		Type: short integer
			MMS 1.0	0x90
			MMS 1.1	0x91
			MMS 1.2	0x92
			MMS 1.3	0x93
X-Mms-Previously-Sent-By	0x20	0xA0	A US-ASCII or UTF-8 string prefixed with a "forward count"	Type: indexed encoded string
X-Mms-Previously-Sent-Date	0x21	0xA1	An absolute date prefixed with a "forward count"	Type: indexed date
X-Mms-Priority	0x0F	0x8F		Type: enumeration
			Low	0x80
			Medium	0x81
			High	0x82
X-Mms-Quotas	0x2B	0xAB		Type: enumeration
			Yes	0x80
			No	0x81
X-Mms-Read-Report	0x10	0x90		Type: enumeration
			Yes	0x80
			No	0x81

(*Continued*)

Table 6.24 (*Continued*)

Parameter name	A.N.	B.E.	Values	Binary encoding
X-Mms-Read-Status	0x1B	0x9B		Type: enumeration
			Read	0x80
			Deleted without being read	0x81
X-Mms-Reply-Charging	0x1C	0x9C		Type: enumeration
			Requested	0x80
			Requested text only	0x81
			Accepted	0x82
			Accepted text only	0x83
X-Mms-Reply-Charging-Deadline	0x1D	0x9D	Absolute time or relative time from submission time	Type: absolute or related date
X-Mms-Reply-Charging-ID	0x1E	0x9E	US-ASCII string	Type: text string
X-Mms-Reply-Charging-Size	0x1F	0x9F	Number of bytes	Type: long integer
X-Mms-Report-Allowed	0x11	0x91		Type: enumeration
			Yes	0x81
			No	0x82
X-Mms-Response-Status	0x12	0x92	See error codes in Appendix F	Type: enumeration
X-Mms-Response-Text	0x13	0x93	US-ASCII or UTF-8 string	Type: encoded string
X-Mms-Retrieve-Status	0x19	0x99	See error codes in Appendix G	Type: enumeration
X-Mms-Retrieve-Text	0x1A	0x9A	US-ASCII or UTF-8 string	Type: encoded string
X-Mms-Sender-Visibility	0x14	0x94		Type: enumeration
			Hide	0x80
			Show	0x81
X-Mms-Status	0x15	0x95		Type: enumeration
			Expired	0x80
			Retrieved	0x81
			Rejected	0x82
			Deferred	0x83
			Unrecognized	0x84
			Indeterminate	0x85
			Forwarded	0x86

(*Continued*)

Table 6.24 (*Continued*)

Parameter name	A.N.	B.E.	Values	Binary encoding
`X-Mms-Store`	0x22	0xA2		Type: enumeration
			`Yes`	0x80
			`No`	0x81
`X-Mms-Stored`	0x27	0xA7		Type: enumeration
			`Yes`	0x80
			`No`	0x81
`X-Mms-Store-Status`	0x25	0xA5	See error codes in Appendix H	Type: enumeration
`X-Mms-Store-Status-Text`	0x26	0xA6	US-ASCII string	Type: text string
`X-Mms-Totals`	0x29	0xA9		Type: enumeration
			`Yes`	0x80
			`No`	0x81
`X-Mms-Transaction-ID`	0x18	0x98	US-ASCII string	Type: text string

6.4 MM3 Interface, MMSC External Servers

The *MM3 interface* allows the MMSC to exchange messages with external servers such as Email servers and SMS centers (SMSCs). This interface is typically based on existing IP-based Email protocols (e.g., SMTP).

When sending a message to an external messaging system, the MMSC converts the multimedia message into an appropriate format supported by the external messaging system. For instance, the exchange of a message between an MMSC and an Internet Email server can be performed by converting the multimedia message from its MM1 binary multipart representation into a text-based MIME representation for transfer over SMTP. In order to receive a message from an external messaging system, the MMSC converts incoming messages to a format supported by receiving MMS clients.

Several mechanisms can be used for discovering incoming messages from external messaging servers. These mechanisms include the following:

- Forwarding of the message from the external messaging server to the MMSC.
- The external messaging server notifies the MMSC that a message is waiting for retrieval. In this configuration, it is the responsibility of either the MMSC or the recipient MMS client to explicitly retrieve the message.
- The MMSC can periodically poll the external messaging server for incoming messages.

Mobile users are often able to send multimedia messages to Internet users. Owing to obvious billing issues, it is usually not possible for Internet users to send messages (received as multimedia messages) to mobile users if the applicable billing model is the one where the sender pays for message delivery. However, it is sometimes permitted for an Internet user to reply once only “free of charge” to a multimedia message.

6.5 MM4 Interface, MMSC–MMSC

The *MM4 interface* allows interactions between two MMSCs. Such interactions are required in the situation in which multimedia messages and associated reports are exchanged between users attached to distinct MMSEs. In this context, a multimedia message is always delivered to the recipient via the recipient MMSC, distinct from the originator MMSC. Note that this mechanism of exchanging messages between MMS users differs from the one usually in place for SMS. For the exchange of an SMS message, the originator SMSC is in charge of delivering directly the message to the recipient(s).[1]

At the time of writing, the only standardized technical realization available for the MM4 interface was the one defined by 3GPP in [3GPP-23.140] (from Release 4). In this technical realization, transactions specified for the MM4 interface are conveyed over the Simple Mail Transfer Protocol (SMTP) as shown in Figure 6.24. Table 6.25 lists the three transactions that can occur over the MM4 interface (6 PDUs).

[1] In rare cases, it happens that the delivery of an SMS message is performed by the recipient SMSC. This configuration is implemented when two networks cannot interwork directly because they rely on incompatible transport technologies.

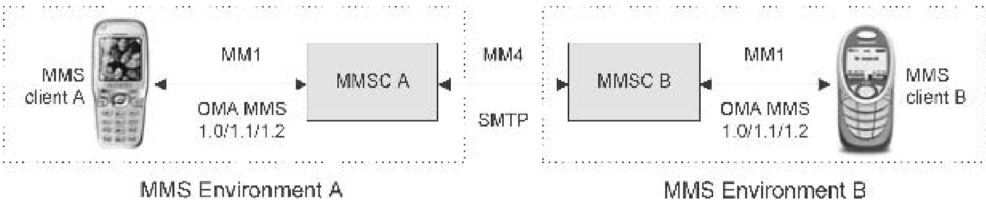


Figure 6.24 Interworking between two distinct MMS environments

As for the MM1 interface, each transaction in the MM4 interface is usually composed of a request and a response/confirmation. The 3GPP naming convention for naming requests and responses is used in this book for description of transactions occurring over the MM4 interface (see Section 6.1).

At the time of writing, MMS national interworking, defined as the interworking between mobile operators providing the service in the same country, has been widely enabled. On the other hand, MMS international interworking, defined as interworking between operators providing the service in distinct countries is still to be achieved on a wide scale.

Note that interworking is not required for roaming users to send messages back to subscribers of their home network. National or international interworking are only required for the exchange of messages between users belonging to different networks.

The transport channel required for interconnecting networks to allow the exchange of messages between distinct MMS environments is typically selected from one of the three possible options below:

- *Leased line*: this option is commonly used for enabling national interworking. The cost of international leased lines is usually too high to make it a commercially acceptable solution for international interworking.
- *IP-VPN*: the establishment of an IP-VPN over Internet is sometimes used to allow the exchange of messages between MMS environments. However, it is difficult to guarantee a certain level of service quality over the Internet.

Table 6.25 List of MM4 transactions/PDUs

Transaction	PDU name	Description	From 3GPP
Routing forward a message	`MM4_forward.REQ`	Message forward request	Rel-4
	`MM4_forward.RES`	Message forward confirmation	
Routing forward a delivery report	`MM4_delivery_report.REQ`	Request for delivery report forward	Rel-4
	`MM4_delivery_report.RES`	Confirmation for delivery report forward	
Routing forward a read report	`MM4_read_reply_report.REQ`	Request for read-report forward	Rel-4
	`MM4_read_reply_report.RES`	Confirmation for read-report forward	

- *GPRS Roaming Exchange* (GRX) : a GRX provides services allowing the interconnection of several GPRS operators. A GRX typically relies on a private Wide Area Network (WAN) interconnecting mobile networks for the exchange of IP traffic including roaming and MMS traffics. Today, multiple GRX providers do offer such commercial services and, therefore, available WANs are often interconnected via peering points in order to achieve global connectivity for mobile operators. Such peering points do exist in Amsterdam and Singapore.

At the SMTP level, two topologies are considered for interconnecting MMSCs (or attached mail transfer agents) as described below:

- *Direct peering*: this topology consists of routing directly messages to the recipient MMSC. Pre-requisites for this direct peering are that routing tables of the originating environment (DNS, MMSC, and/or mail transfer agent look-up tables) are correctly configured and a commercial agreement is in place between the two parties exchanging the message. Direct peering is appropriate when the number of destinations is low (acceptable to achieve national interworking, for instance). However, direct peering does not scale very well when the number of destinations is high. For instance, this topology does not look appropriate for enabling international interworking where hundreds of direct peering connections would have to be established.
- *Hub concept*: this topology consists of having a central MMS gateway, known as *MMS hub*, to which multiple MMS centers get connected according to a star topology. A cascade billing model can be adopted where one commercial agreement is established between the hub provider and the connected operator. This agreement allows the exchange of messages with all available destinations at predefined transit and termination fees. Such a hub service is typically provided by GRX providers and, therefore, the GRX is usually the preferred physical bearer to interconnect an MMSC to the hub. In order to achieve global MMS connectivity, peering is established between available MMS hubs as shown in Figure 6.25.

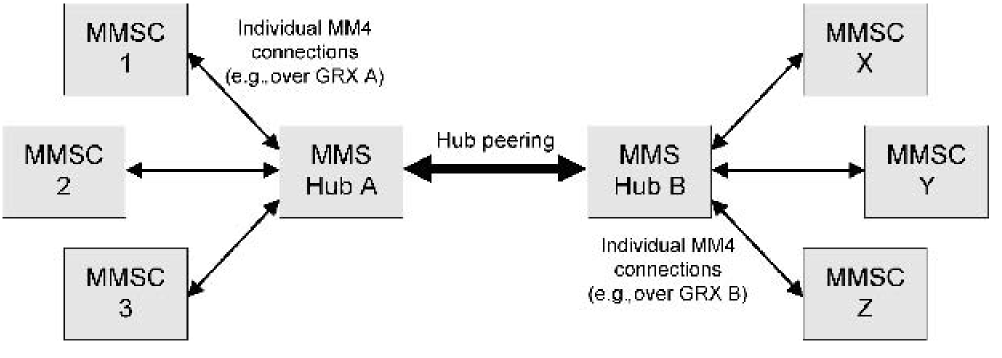


Figure 6.25 MM4 and hub concept

6.5.1 Introduction to SMTP

SMTP is a basic protocol for exchanging messages between Mail User Agents (MUAs). MUA is the name given to applications in charge of managing Email messages exchanged over the Internet. In the Internet domain, SMTP has become the de facto Email transfer protocol for exchanging messages. MUAs have similar responsibilities as those of MMS clients in an MMSE. Specifications of SMTP have been published by the Internet Engineering Task Force (IETF) in [RFC-2821] ([RFC-2821] obsoletes [RFC-821]).

SMTP relies on a model based on an interconnection of Mail Transfer Agents (MTAs). In a typical SMTP transaction, an MTA is the *sender-SMTP* if it originates the SMTP commands, or the *receiver-SMTP* if it handles the received SMTP commands. An MTA can usually play both roles: the sender-SMTP client and the receiver-SMTP server. SMTP defines how senders and receivers can initiate a transfer session, can transfer messages(s) over a session, and can tear down an open session. Note that how the message physically transferred from the sender to the receiver is not defined as part of the SMTP specifications (see Box 6.2). SMTP only defines the set of commands, and corresponding responses, for controlling the transfer of messages over sessions.

Box 6.2 Interworking between distinct environments, GSMA recommendations

Ability to exchange messages between distinct environments is crucial for the success of MMS. To enable this, 3GPP has identified SMTP as a suitable transport protocol for the realization of the interface bridging MMS centers belonging to different MMS providers.

Furthermore, the GSM Association (GSMA) has published recommendations to ensure an efficient interworking over the MM4 interface [GSMA-IR.52]. The interconnection between two mobile networks for the realization of the MM4 interface can be performed over the public Internet (with secure connections) or over direct leased lines such as Frame relay or Asynchronous Transfer Mode (ATM). However, GSMA recommends the use of an alternative solution based on GRX [GSMA-IR.34] as an interconnection and transmission network for MMS traffic. GRX enables multimedia messages to be routed as IP-based traffic between two distinct mobile networks.

In the roaming scenario, the roaming user still uses the home MMSC in order to send and retrieve multimedia messages. This means that a roaming user is not required to change the MMS device settings in order to have access to the service. The visited network is not required to provide support for MMS, but the user should be able to establish a data connection (e.g., GSM Circuit Switched Data (CSD) or GPRS) in order to have access to the service.

SMTP is a stateful protocol, meaning that the sender and the receiver involved in operations over a session maintain a current context for a session. Consequently, commands requested over SMTP have different results according to the session state.

In the context of MMS, SMTP is used to transfer multimedia messages, delivery reports, and read reports between MMSCs. For this purpose, originator and recipient MMSCs are MTAs playing the roles of sender-SMTP and receiver-SMTP, respectively.

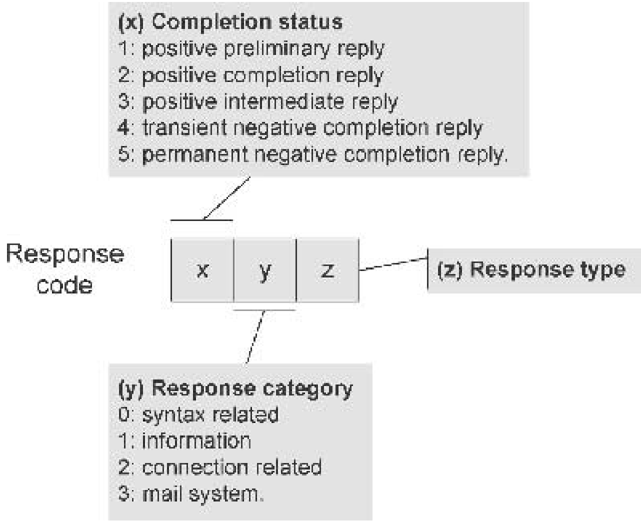


Figure 6.26 Structure of an SMTP response

An SMTP command is a four-letter command such as `HELO` or `DATA`. The response to such a command consists of a three-digit code followed by some optional human readable text. The status code is formatted according to the convention shown in Figure 6.26.

The minimum set of SMTP commands that should be supported by MMSCs is the following:

- `HELO` or `EHLO`: this command (abbreviation for "Hello") is used for initiating a session.
- `QUIT`: this command is used to tear down a session.
- `MAIL`: this command tells the SMTP-receiver that a message transfer is starting and that all state tables and buffers should be re-initialized. This command has one parameter named `FROM`, which identifies the address of the message originator.
- `RCPT`: this command (abbreviation for "Recipient") has one parameter, named `TO`, which identifies the address for one of the message recipients. If the message is sent to several recipients, then this command may be executed multiple times for one single message transfer.
- `DATA`: this command is used for transferring the message itself.

Phone numbers and Email addresses can be used for addressing recipients in an MMSE. With SMTP, only Email addresses are supported for routing purpose. Consequently, phone numbers specified as part of `MAIL` and `RCPT` parameters need to be adapted from original and recipient MMS client addresses. For instance, the phone number +49172287376 for an originator MMS client is converted to a Fully Qualified Domain Name (FQDN) Email address as illustrated in Figure 6.27.

For sending a message to an external MMSE, the originator MMSC needs to resolve the recipient MMSC domain name to an IP address. Two resolution methods have been identified by 3GPP in [3GPP-23.140] Release 4 which are as folllows:

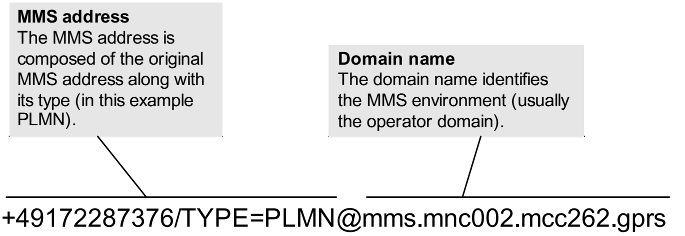


Figure 6.27 SMTP address conversion

- *IMSI-based method:* this method for identifying the recipient MMSC IP address complies with the Mobile Number Portability (MNP) requirements. The MMSC interrogates the recipient Home Location Register (HLR) in order to get the International Mobile Subscriber Identity (IMSI) corresponding to the recipient phone number (via the MM5 interface). From the IMSI, the originator MMSC extracts the Mobile Network Code (MNC) and the Mobile Country Code (MCC) as illustrated in Figure 6.28. With the MNC and MCC, the originator MMSC obtains the recipient MMSC FQDN by interrogating a local database (mapping table) or by constructing the FQDN according to the naming convention defined in [3GPP-23.140] from Release 6. With the MMSC FQDN, the originator MMSC retrieves the MMSC IP address by interrogating a DNS.
- *DNS-ENUM-based method*: IETF has identified a method for locating a device associated with a phone number with a DNS [RFC-2916]. This method, known as DNS-ENUM, allows DNS records to be retrieved using the recipient phone number as a record key. The originator MMSC can retrieve the IP address of the recipient MMSC in the DNS-ENUM records of the message recipients. Such a solution is not widely supported today.

The originator MMSC typically unbundles messages prior to their transfer over the MM4 interface. This means that if the message is sent to two recipients belonging to the same network then the message will be sent twice over the MM4 interface.

MMSCs can support SMTP extensions shown in the next page:

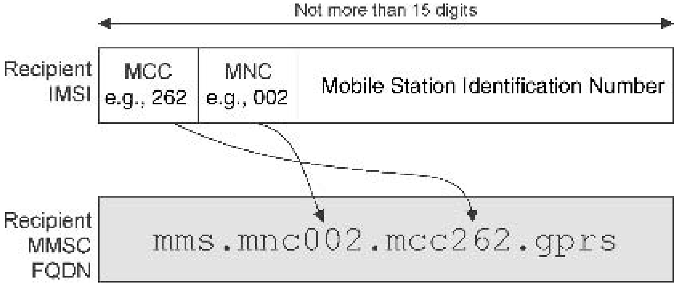


Figure 6.28 3GPP naming convention for MMSC FQDN

- SMTP service extension for message size declaration.
- SMTP service extension for 8-bit MIME transport.

The support of additional commands is also possible, but their use has not yet been covered in 3GPP technical specifications. An example for transferring a message over SMTP between two distinct MMSEs is provided in Section 6.5.5. The following sections present transaction flows for message forward, delivery-report forward, and read-report forward.

6.5.2 Routing Forward a Message

The request for forwarding a message (`MM4_forward.REQ` request) over the MM4 interface allows a multimedia message to be transferred between two MMSEs. If requested by the originator MMSC, the recipient MMSC acknowledges the message forward request with a message forward response (`MM4_forward.RES` response). The transaction flow for the transfer of a message over the MM4 interface is shown in Figure 6.29.

The `MM4_forward.REQ` request is composed of parameters listed in Table 6.26. If requested by the originator MMSC, the recipient MMSC acknowledges the message forward request with a message forward response (`MM4_forward.RES` response). The response is composed of parameters listed in Table 6.27.

Unlike forward requests, the addressing of message forward responses is related to neither the message originator nor the message recipient. Instead, the addressing of a forward response is related to special system addresses. The value to be assigned to the `To` parameter of the response is the value assigned to the `X-MMS-Originator-System` parameter of the corresponding forward request (usually a special system address identifying the originator MMSC). The value to be assigned to the `Sender` parameter is a special address identifying the recipient MMSC. It is suggested that special system addresses should be formatted in the form:

```
system-user@mms.mnc002.mcc262.gprs
```

If the forward request is processed without error by the recipient MMSC, the following value is assigned to the request status code parameter (`X-Mms-Request-Status-Code` parameter):

- `Ok`: this status code indicates that the corresponding request has been processed without errors.

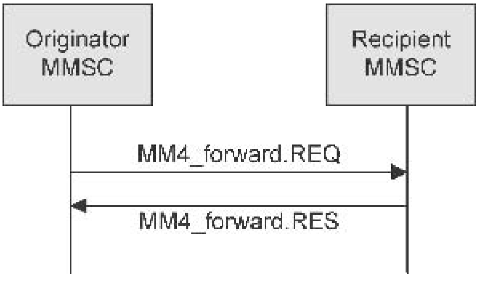


Figure 6.29 MM4 message forward

Table 6.26 MM4 message forward request (MM4_forward.REQ)

Parameter name	Description	From 3GPP	St.
X-Mms-3GPP-MMS-Version	3GPP MMS Version–MMS Version of the MMSC Type: string; Example: 5.2.0	Rel-4	●
X-Mms-Message-Type	Type of MM4 operation (request for the forward of a message); Type: string; Value: MM4_forward.REQ	Rel-4	●
X-Mms-Transaction-ID	Identifier of the forward transaction; Type: string	Rel-4	●
X-Mms-Message-ID	Identifier of the multimedia message being forwarded; Type: string	Rel-4	●
To, Cc	Recipient address(es)–address(es) of the recipient(s) of the original message; Type: string	Rel-4	●
From	Sender address–address of the sender of the original message; Type: string	Rel-4	●
Subject	Message subject; Type: string; Condition (A)	Rel-4	©
X-Mms-Message-Class	Message class; Type: string; Condition (A); Values: Personal, Advertisement, Information, or Auto	Rel-4	©
Date	Date and time when the original message was handled (retrieved, expired, rejected, etc.); Type: date	Rel-4	●
X-Mms-Expiry	Time of expiry of the message; Type: date or duration; Condition (A)	Rel-4	©
X-Mms-Delivery-Report	Whether or not a delivery report is requested by the message originator; Condition (A); Values: Yes or No	Rel-4	©
X-Mms-Originator-R/S-Delivery-Report	Whether or not a delivery report is requested by the originator MMSC; Values: Yes or No (default)	Rel-6	©
X-Mms-Read-Reply	Whether or not a read report is requested; Condition (A); Values: Yes or No	Rel-4	©
X-Mms-Priority	Priority of the message being forwarded; Condition (A); Values: Low, Normal or High	Rel-4	©
X-Mms-Sender-Visibility	Whether or not the sender requested sender details to be hidden from recipients; Condition (A); Values: Hide or Show	Rel-4	©
X-Mms-Forward-Counter	Counter indicating how many times the message has been forwarded; Condition (A); Type: Integer	Rel-4	©
X-Mms-Previously-Sent-By	Address(es) of MMS clients that have handled (submitted or forwarded) the message prior to the manipulation by the MMS client whose address is assigned to the From parameter Type: string with index; Example: 1, armel@armorepro.com 2, gwenael@lebodic.net	Rel-4	○
X-Mms-Previously-Sent-Date-and-Time	Date and time when the message was handled; Type: date with index; Example: 1, Mon Jan 21 09:45:33 2003 2, Wed Jan 23 18:06:21 2003	Rel-4	○

(*Continued*)

Table 6.26 (*Continued*)

Parameter name	Description	From OMA	St.
X-Mms-Ack-Request	Whether or not an acknowledgement for the forward request is requested; Values: Yes or No	Rel-4	○
Sender	Originator address as determined by the SMTP MAIL FROM command; Type: string	Rel-4	●
X-Mms-Originator-System	System address to which the requested forward response should be sent; Condition (B); Type: string	Rel-4	©
Message-ID	Each SMTP request/response has a unique reference assigned to the Message-ID parameter; Type: string	Rel-4	●
X-Mms-Applic-ID	Identification of the recipient application; Type: quoted string	Rel-6	○
X-Mms-Reply-Applic-ID	Identification of the application handling replies (reply path); Type: quoted string	Rel-6	○
X-Mms-Aux-Applic-Info	Auxiliary application addressing information; Type: quoted string	Rel-6	○
X-Mms-Content-Class	Identification of the smallest content class to which the message complies; Values: text, image-basic, image-rich, megapixel, video-basic, video-rich, content-basic, or content-rich	Rel-6	○
X-Mms-Drm-Content	Whether or not the message contains DRM protected contents; Values: yes or no	Rel-6	○
X-Mms-Adaptation-Allowed	Whether or not content adaptation is allowed. If omitted then content adaptation is allowed; Values: yes or no	Rel-6	○
Content-Type	Message content type; Type: string	Rel-4	●
Message body	Message content; Condition (A)	Rel-4	©

Conditions: (A) Available only if provided by the originator MMS client, (B) Required if a forward response is requested.

If errors occurred during the processing of the forward request, the codes listed in Appendix I can be assigned to the request status code parameter of the corresponding response.

6.5.3 Routing Forward a Delivery Report

The request for forwarding a delivery report (MM4_delivery_report.REQ request) over the MM4 interface allows the transfer of a delivery report between two MMSEs. If requested by the recipient MMSC, the originator MMSC acknowledges the request with a response (MM4_delivery_report.RES response). The transaction flow for the transfer of a delivery report over the MM4 interface is shown in Figure 6.30.

The different delivery states that can be reported over the MM4 interface are as follows:

- expired
- retrieved

Table 6.27 MM4 message forward response (MM4_forward.RES)

Parameter name	Description	From 3GPP	St.
X-Mms-3GPP-MMS-Version	3GPP MMS Version–MMS Version of the MMSC; Type: string; Example: 5.2.0	Rel-4	●
X-Mms-Message-Type	Type of MM4 operation (response for the forward of a message) Type: string; Value: MM4_forward.RES	Rel-4	●
X-Mms-Transaction-ID	Identifier of the forward transaction; Type: string	Rel-4	●
X-Mms-Message-ID	Identifier of the multimedia message being forwarded; Type: string	Rel-4	●
X-Mms-Request-Status-Code	Status code of the request to forward the message Values: Ok or error codes defined in Appendix I	Rel-4	●
X-Mms-Status-Text	Optional status text; Type: string	Rel-4	○
Sender	System address–address of the recipient MMSC; Type: string	Rel-4	●
To	System address–address of the originator MMSC; Type: string	Rel-4	●
Message-ID	Each SMTP request; response has a unique reference assigned to the Message-ID parameter; Type: string	Rel-4	●
Date	Date provided by the recipient MMSC; Type: string	Rel-4	●
X-Mms-Request-Recipients	List of recipients to which the status of the response applies. If this field is absent then the status applies to all recipients	Rel-6	©

- deferred
- indeterminate
- forwarded
- unrecognized.

The MM4_delivery_report.REQ request is composed of parameters listed in Table 6.28. The MM4_delivery_report.RES response is composed of parameters listed in Table 6.29.

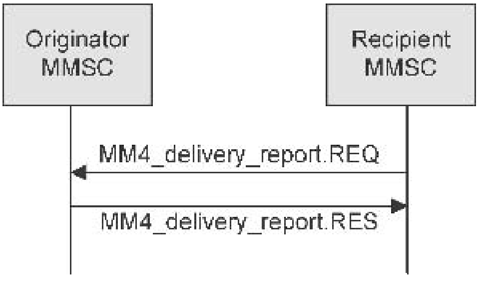


Figure 6.30 MM4 delivery report

Table 6.28 MM4 delivery report forward request (MM4_delivery_report.REQ)

Parameter name	Description	From 3GPP	St.
X-Mms-3GPP-MMS-Version	3GPP MMS Version – MMS Version of the MMSC Type: string; Example: 5.2.0	Rel-4	●
X-Mms-Message-Type	Type of MM4 operation (request for the forward of a delivery report); Type: string; Value: MM4_delivery_report.REQ	Rel-4	●
X-Mms-Transaction-ID	Identifier of the forward transaction; Type: string	Rel-4	●
X-Mms-Message-ID	Identifier of the corresponding multimedia message; Type: string	Rel-4	●
From	Recipient address–address of the recipient of the original message; Type: string	Rel-4	●
To	Sender address–address of the sender of the original message; Type: string	Rel-4	●
Date	Message date and time–date and time when the message was handled (retrieved, expired, rejected, etc.); Type: date	Rel-4	●
X-Mms-Ack-Request	Request for an acknowledgement–whether or not an acknowledgement of the forward request is requested; Values: Yes or No	Rel-4	○
X-Mms-MM-Status-Code	Message status code–status of the corresponding multimedia message; Type: string; Values: Expired, Retrieved, Deferred, Indeterminate, Forwarded, or Unrecognised	Rel-4	●
X-Mms-Status-Text	Text corresponding to the status code; Type: string	Rel-4	○
X-Mms-MM-Status-Extension	Extension for the message status; Type: string; Values: • Rejection-by-mms-recipient: the recipient refused to receive the message • Rejection-by-other-rs: the recipient MMSC refused to deliver the message to the recipient	Rel-6	○
X-Mms-Forward-To-Originator-UA	Indication whether or not the originator MMSC is allowed to forward the delivery report to the message originator; Values: Yes (default) or No	Rel-6	○
Sender	System address–address to which the requested response should be sent; Type: string	Rel-4	●
Message-ID	Each SMTP request; response has a unique reference assigned to the Message-ID parameter; Type: string	Rel-4	●
X-Mms-Applic-ID	Identification of the originator application of the original message; Type: quoted string	Rel-6	○
X-Mms-Reply-Applic-ID	Identification of the recipient application of the original message; Type: quoted string	Rel-6	○
X-Mms-Aux-Applic-Info	Auxiliary application addressing information (as indicated in the original message); Type: quoted string	Rel-6	○

Table 6.29 MM4 delivery report forward response (MM4_delivery_report.RES)

Parameter name	Description	From 3GPP	St.
X-Mms-3GPP-MMS-Version	3GPP MMS Version–MMS Version of the MMSC Type: string; Example: 5.2.0	Rel-4	●
X-Mms-Message-Type	Type of MM4 operation (response for the forward of a delivery report); Type: string; Value: MM4_delivery_report.RES	Rel-4	●
X-Mms-Transaction-ID	Identifier of the forward transaction; Type: string	Rel-4	●
X-Mms-Message-ID	Identifier of the corresponding multimedia message; Type: string	Rel-4	●
X-Mms-Request-Status-Code	Status code of the request to forward the delivery report; Values: Ok or error codes defined in Appendix I	Rel-4	●
X-Mms-Status-Text	Optional status text; Type: string	Rel-4	○
Sender	System address–address of the recipient MMSC; Type: string	Rel-4	●
To	System address–address of the originator MMSC; Type: string	Rel-4	●
Message-ID	Each SMTP request; response has a unique reference assigned to the Message-ID parameter; Type: string	Rel-4	●
Date	Date provided by the recipient MMSC; Type: string	Rel-4	●

For the response, the value to be assigned to the To parameter is the value assigned to the Sender parameter of the corresponding forward request. The value to be assigned to the Sender parameter is the system address of the recipient MMSC.

6.5.4 Routing Forward a Read Report

The request for forwarding a read report (MM4_read_reply_report.REQ request) over the MM4 interface allows a read report to be transferred between two distinct MMSEs. If requested by the recipient MMSC, the originator MMSC acknowledges the request with a response (MM4_read_reply_report.RES response). The transaction flow for the transfer of a read report over the MM4 interface is shown in Figure 6.31.

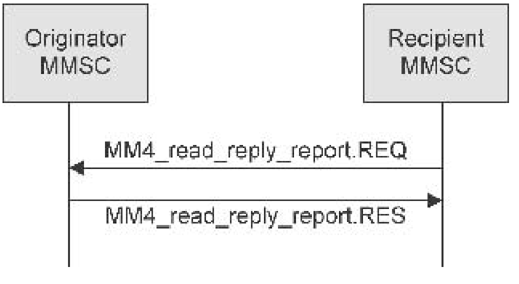


Figure 6.31 MM4 read report

The different read states that can be reported over the MM4 interface are as follows:

- read
- deleted without being read.

The `MM4_read_reply_report.REQ` request is composed of parameters listed in Table 6.30. The `MM4_read_reply_report.RES` response is composed of parameters listed in Table 6.31.

For the response, the value to be assigned to the `To` parameter is the value assigned to the `Sender` parameter of the corresponding forward request. The value to be assigned to the `Sender` parameter is the system address of the recipient MMSC.

Table 6.30 MM4 read-report forward request (`MM4_read_reply_report.REQ`)

Parameter name	Description	From OMA	St.
X-Mms-3GPP-MMS-Version	3GPP MMS Version–MMS Version of the MMSC Type: string; Example: 5.2.0	Rel-4	●
X-Mms-Message-Type	Type of MM4 operation (request for the forward of a read report); Type: string; Value: MM4_read_reply_report.REQ	Rel-4	●
X-Mms-Transaction-ID	Identifier of the forward transaction; Type: string	Rel-4	●
X-Mms-Message-ID	Identifier of the corresponding multimedia message; Type: string	Rel-4	●
From	Recipient address–address of the recipient of the original message; Type: string	Rel-4	●
To	Sender address–address of the sender of the original message; Type: string	Rel-4	●
Date	Message date and time–date and time when the message was handled (read or deleted); Type: date	Rel-4	●
X-Mms-Ack-Request	Request for an acknowledgment–whether or not an acknowledgment of the forward request is requested; Values: Yes or No	Rel-4	○
X-Mms-Read-Status	Message status code–status of the corresponding multimedia message; Type: string; Values: Read or Deleted without being Read	Rel-4	●
X-Mms-Status-Text	Text corresponding to the status code; Type: string	Rel-4	○
Sender	System address–address to which the requested response should be sent; Type: string	Rel-4	●
Message-ID	Each SMTP request; response has a unique reference assigned to the Message-ID parameter; Type: string	Rel-4	●
X-Mms-Applic-ID	Identification of the originator application of the original message; Type: quoted string	Rel-6	○
X-Mms-Reply-Applic-ID	Identification of the recipient application of the original message; Type: quoted string	Rel-6	○
X-Mms-Aux-Applic-Info	Auxiliary application addressing information (as indicated in the original message); Type: quoted string	Rel-6	○

Table 6.31 MM4 read-report forward response (`MM4_read_reply_report.RES`)

Parameter name	Description	From 3GPP	St.
`X-Mms-3GPP-MMS-Version`	3GPP MMS Version–MMS Version of the MMSC Type: string; Example: `5.2.0`	Rel-4	●
`X-Mms-Message-Type`	Type of MM4 operation (response for the forward of a read report); Type: string; Value: `MM4_read_reply_report.RES`	Rel-4	●
`X-Mms-Transaction-ID`	Identifier of the forward transaction; Type: string	Rel-4	●
`X-Mms-Request-Status-Code`	Status code of the request to forward the delivery report; Values: `Ok` or error codes defined in Appendix I	Rel-4	●
`X-Mms-Status-Text`	Optional status text; Type: string	Rel-4	○
`Sender`	System address – Address of the recipient MMSC; Type: string	Rel-4	●
`To`	System address – Address of the originator MMSC; Type: string	Rel-4	●
`Message-ID`	Each SMTP request; response has a unique reference assigned to the `Message-ID` parameter; Type: string	Rel-4	●
`Date`	Date provided by the recipient MMSC; Type: string	Rel-4	●

6.5.5 Example for Message Transfer with SMTP

Figure 6.32 shows the sequence of SMTP instructions required for (1) opening an SMTP session, (2) transferring a message, and (3) tearing down the session. Note that values assigned to `From`, `To`, and `Cc` parameters are not used for routing purpose over SMTP. These values are conveyed transparently over SMTP. Consequently, these values may be formatted as Email addresses or as phone numbers. Values used for routing purpose in SMTP are those assigned to `MAIL` and `RCPT` parameters.

6.6 MM5 Interface, MMSC–HLR

The *MM5 interface* enables interactions between an MMSC and other network entities such as the HLR. Operations that can be invoked over the MM5 interface include the following:

- Interrogating the HLR to obtain routing information for the purpose of forwarding a message from one MMSC to another over the MM4 interface.
- Determination of the recipient or sender location to possibly charge an additional fee for message submissions and/or message retrievals when the user is roaming. Specific application logic may be applied by the MMSC when the user is roaming, e.g., diverting the incoming messages to an Email address when user is roaming.

If the MM5 interface is present in the MMSE, then the interface is usually implemented on the basis of existing Mobile Application Part (MAP) operations. At the time of writing, no

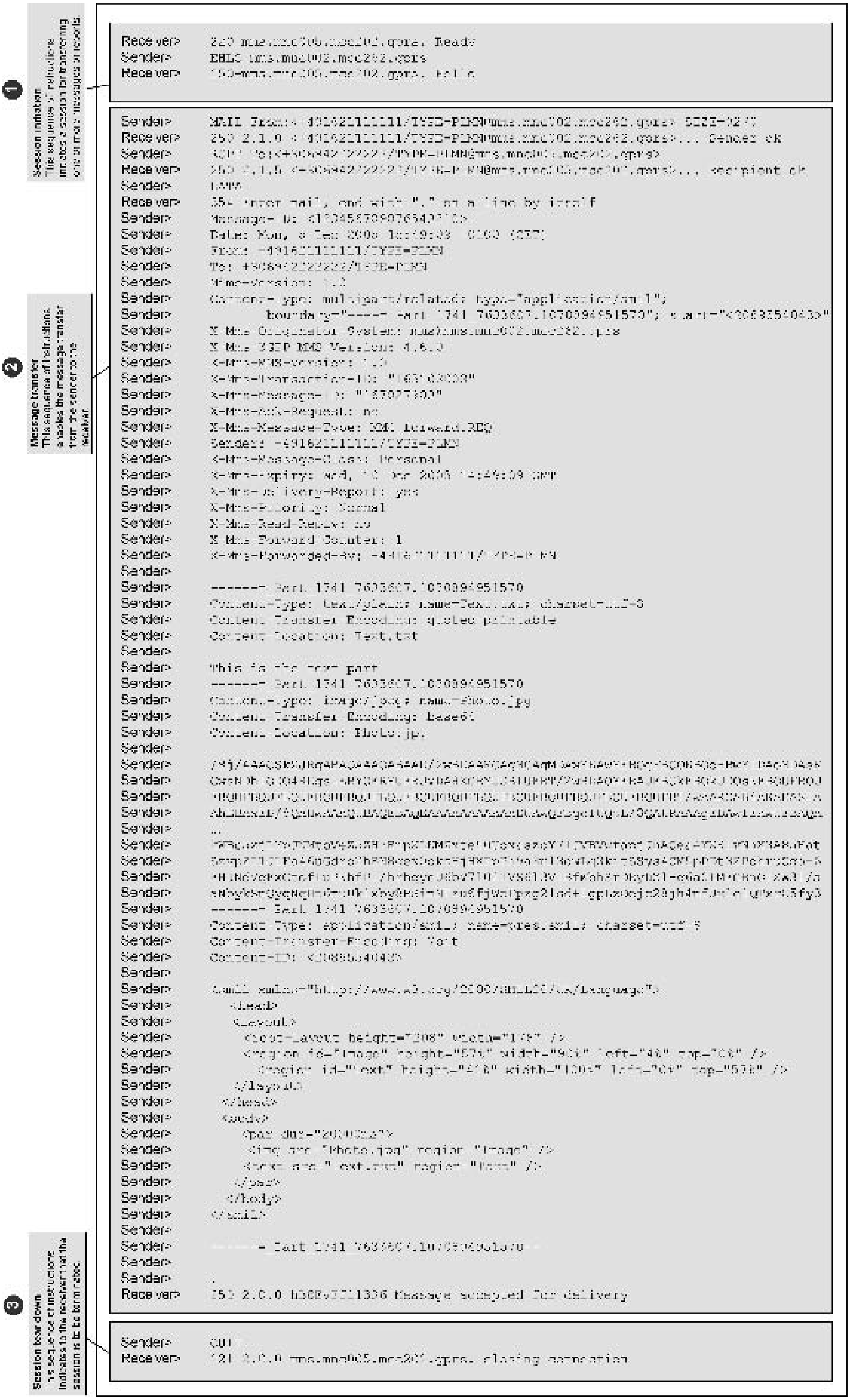


Figure 6.32 Example of message transfer over SMTP

technical realization of the MM5 interface had been specified by standardization organizations.

6.7 MM6 Interface, MMSC–User Databases

The technical realization of the *MM6 interface* allows interactions between the MMSC and external user databases. Generic user information held by the HLR can already be accessed by the MMSC over the MM5 interface. For MMS-specific information, it is common for the MMSC to have an internal user database. If an external user database has to be accessed, it is often realized over the LDAP protocol.

Unfortunately, the MM6 interface has yet to be standardized. Consequently, this interface is not covered in this book.

6.8 MM7 Interface, MMSC–VAS Applications

The *MM7 interface* enables interactions between VAS applications and an MMSC. At the time of writing, the only standardized technical realization for the MM7 interface is the one published by 3GPP in [3GPP-23.140] Release 5 (*stages 2* and *3*) and extended in Release 6. Table 6.32 lists the eight transactions that can occur over the MM7 interface (14 PDUs).

This technical realization is based on the *Simple Object Access Protocol* (SOAP) with HTTP at the transport layer. Figure 6.33 shows a typical network configuration allowing a VAS application to interact with several MMS clients. In this configuration, the VAS application and the MMSC play dual roles of sender and receiver of SOAP messages.

Table 6.32 List of MM7 transactions/PDUs

Transaction	PDU name	Description	From 3GPP
Message submission	`MM7_submit.REQ` `MM7_submit.RES`	Message submission request Message submission response	Rel-5
Message delivery	`MM7_deliver.REQ` `MM7_deliver.RES`	Message delivery request Message delivery response	Rel-5
Cancellation	`MM7_cancel.REQ` `MM7_cancel.RES`	Message cancellation request Message cancellation response	Rel-5
Replacement	`MM7_replace.REQ` `MM7_replace.RES`	Message replacement request Message replacement response	Rel-5
Delivery report	`MM7_delivery_report.REQ` `MM7_delivery_report.RES`	Delivery report request Delivery report response	Rel-5
Read report	`MM7_read_reply_report.REQ` `MM7_read_reply_report.RES`	Read-report request Read-report response	Rel-5
MMSC error	`MM7_RS_error.RES`	Error indication from the MMSC to the VAS application	Rel-5
VASP error	`MM7_VASP_error.RES`	Error indication from the VAS application to the MMSC	Rel-5

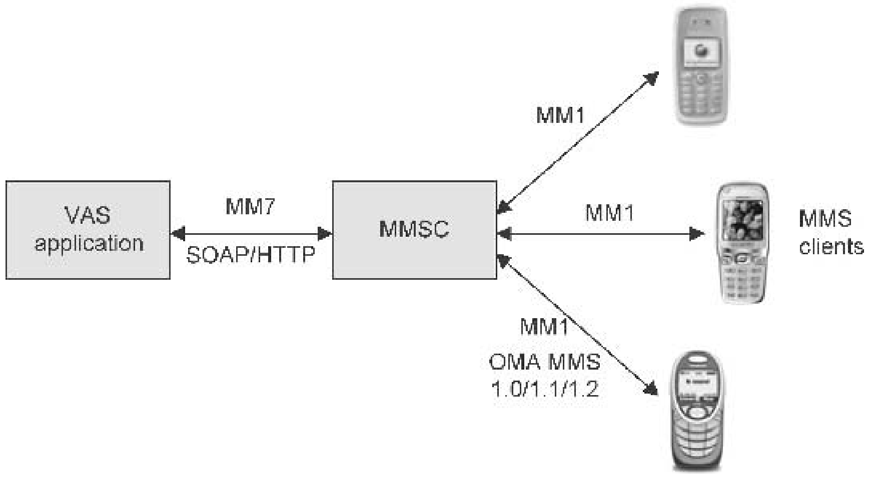


Figure 6.33 Network architecture with a VAS application

In the past, prior to the introduction of the standard MM7 interface, MMSC vendors designed their own proprietary MM7 interfaces. This is the case, for instance, for Nokia MMS centers, which support the External Application InterFace (EAIF). EAIF relies on HTTP where `HTTP Post` requests carry MMS protocol data units between VAS application and MMS center. In EAIF, unlike in 3GPP MM7, protocol data unit are binary encoded as if they were conveyed over the MM1 interface. In EAIF, HTTP headers and Nokia specific extension headers carry some additional information, e.g., routing instructions. Most MMS centers can now be upgraded to support the standard MM7 interface.

As for other interfaces, each transaction invoked over the MM7 interface is composed of a request and a corresponding response. HTTP-level mechanisms[1] can be used in order to authenticate parties communicating over the MM7 interface. Additionally, messages over the MM7 interface can be transported over the Transport Layer Security (TLS) protocol to ensure secure communications.

The support of the MM7 interface is optional for the MMSC. However, if such an interface is supported, then message submission, message delivery, transactions related to the provision of delivery reports, and the management of errors are mandatory transactions to be supported by the MMSC. The support of other transactions such as message cancellation, replacement, and transactions related to the management of read reports is optional for the MMSC.

The three addressing modes, short code, phone number [ITU-E.164], and Email address [RFC-2822], can be used to identify entities communicating over the MM7 interface. Communicating entities include MMS clients, VAS applications, and the MMSC. Addresses can be encrypted or obfuscated over the MM7 interface to maintain user privacy.

[1] For instance, the access authentication mechanism described in [RFC-2617] can be used for authenticating parties communicating over the MM7 interface.

Table 6.33 XML MM7 schema name/version of [3GPP-23.140]

Schema name	Corresponding version of [3GPP-23.140]
REL-5-MM7-1-0	v5.3.0
REL-5-MM7-1-1	v5.4.0
REL-5-MM7-1-2	v5.5.0
REL-5-MM7-1-3	v5.6.0, v5.7.0
REL-5-MM7-1-4	v5.8.0, v5.9.0
REL-5-MM7-1-5	v5.10.0
REL-6-MM7-1-0	v6.1.0, v6.2.0, v6.3.0
REL-6-MM7-1-1	v6.4.0
REL-6-MM7-1-2	v6.5.0
REL-6-MM7-1-3	v6.6.0

SOAP messages are structured according to SOAP technical specifications published by W3C in [W3C-SOAP] and [W3C-SOAP-ATT]. In addition, several versions of the XML schema for formatting MMS-specific SOAP messages are published by 3GPP at the following location:

`http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/`

Several versions of the schema for the MM7 interface are available and each one corresponds to a specific version of the standard [3GPP-23.140] as shown in Table 6.33.

6.8.1 Introduction to SOAP

SOAP is a lightweight protocol for the exchange of information in distributed environments such as the MMSE. All SOAP messages are represented using XML. SOAP specifications consist of the following three distinct parts:

- *Envelope*: this part defines a framework for describing the content of a SOAP message and how to process it.
- *Set of encoding rules*: encoding rules are used for expressing instances of application-defined data types.
- *Convention for representing remote procedure calls*: this convention helps entities in a distributed environment to request services from each other in an interoperable manner.

SOAP may be used over a variety of transport protocols. In the MMSE, for the realization of the MM7 interface, SOAP is used over the HTTP transport protocol. With this configuration, MM7 request messages are transferred in `HTTP Post` requests, whereas corresponding MM7 response messages are transferred as part of `HTTP Response` messages.

A SOAP message, represented using XML, consists of a SOAP envelope, a *SOAP header*, a *SOAP body*, and an optional *SOAP attachment* as shown in Figure 6.34. For messages

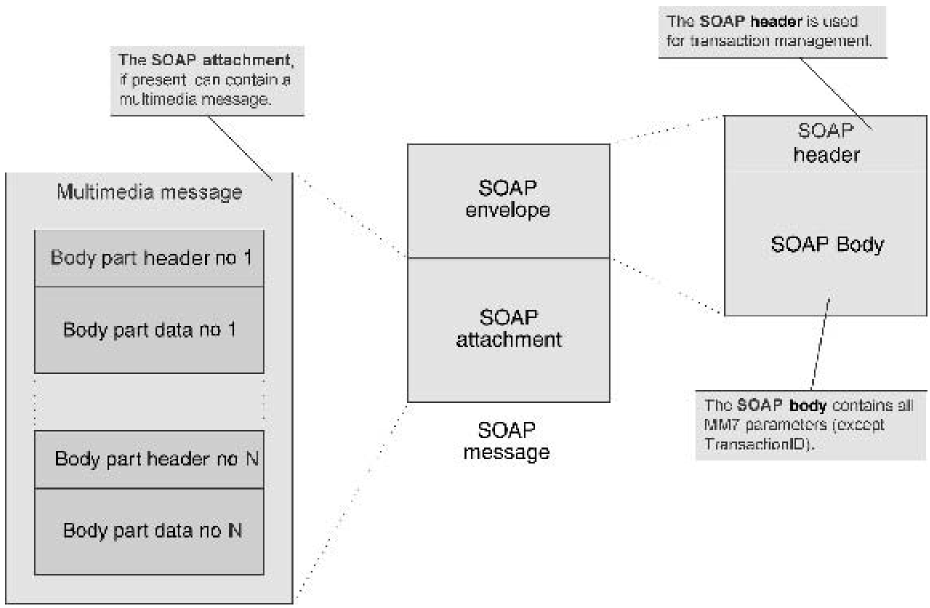


Figure 6.34 Structure of a SOAP message

containing a SOAP envelope only, the media type `text/xml` is used. If the SOAP message also contains an attachment, then the media type `multipart/related` is used and the SOAP envelope is identified with the `Start` parameter of the content type. Each part of the SOAP message has, at least, two parameters `Content-Type` and `Content-ID`.

The SOAP envelope is the first element to appear in `HTTP Post` requests and corresponding responses. The `SOAPAction` parameter is set to the "Null string." The MMSC or the VAS server is identified uniquely with a URL placed in the `host` header field of the `HTTP Post` request.

A request status is provided as part of the corresponding response. The status can be of the following three types:

- *Success or partial success*: this status indicates the successful or partial processing of a request. This status is composed of three parameters: `StatusCode` (numerical code), `StatusText` (human readable textual description), and `Details` (optional human readable detailed textual description). The classification of four-digit error identifiers to be assigned to the `StatusCode` is provided in Figure 6.35 and a full list of defined error codes and corresponding status texts is provided in Appendix J.
- *Processing error*: this status indicates that a fault occurred while parsing SOAP elements. These processing errors include the `faultcode`, `faultstring`, and `detail` elements as defined in [W3C-SOAP].
- *Network error*: this status indicates that an error occurred at the HTTP level.

The 3GPP XML schema defined for the MM7 interface specifies the structure of MMS PDUs embedded in SOAP messages. The following sections describe each one of the 14

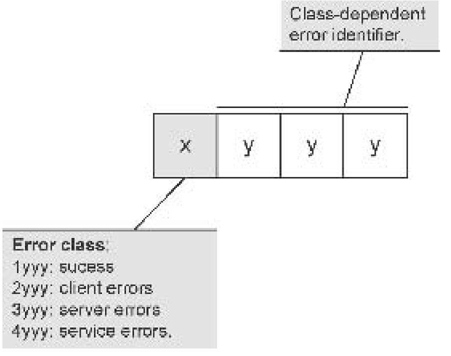


Figure 6.35 Classification of MM7 error codes

PDUs. For this purpose, XML elements composing each PDU are listed and a graphical representation of the overall PDU structure is provided. The graphical representation shows the relationship between the XML schema group elements and associated child elements. The three XML schema grouping operations "`All`," "`Sequence`," and "`Choice`" are used for structuring MMS PDUs. The symbols in Figure 6.36 are used in this book for the graphical representations of XML structures.

6.8.2 Message Submission

Regarding the MM7 interface, *message submission* refers to the submission of a message from an originator VAS application to an MMSC. The message is addressed to a single recipient, to multiple recipients, or to a distribution list managed by the MMSC. If the MMSC

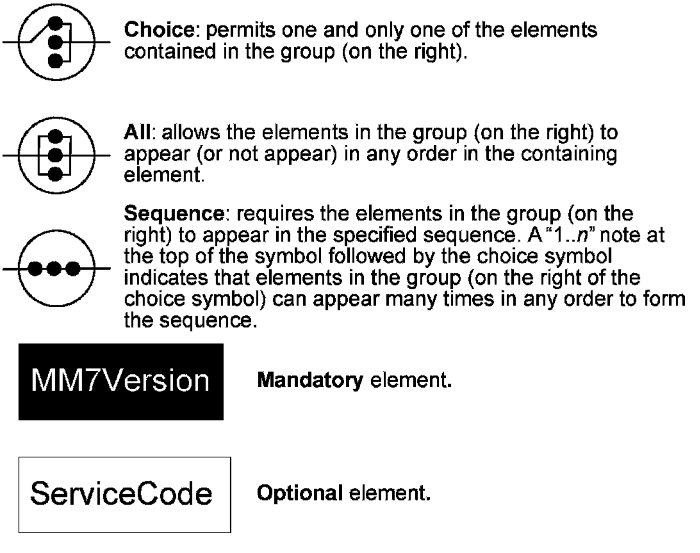


Figure 6.36 Graphical notations for XML schema structures

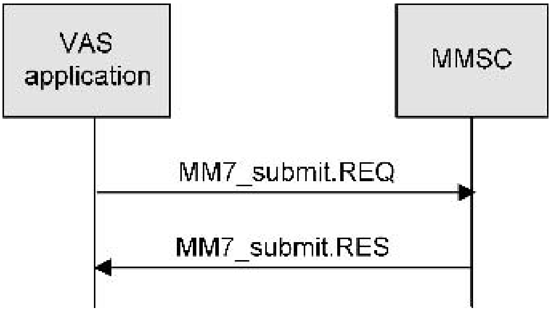


Figure 6.37 MM7 message submission

accepts the submission request, then the MMSC sends back a positive response. This indicates that the submission request is accepted but does not indicate that the message has been successfully delivered to message recipients. The transaction flow in Figure 6.37 shows interactions between the VAS application and the MMSC for the submission of a multimedia message over the MM7 interface.

In tables describing the content of requests and responses for the MM7 interface, the column named "location" (abbreviated "Loc.") indicates whether the corresponding parameter is placed in the SOAP header ("H"), SOAP body ("B"), or SOAP attachment ("A").

The submission request `MM7_submit.REQ` is composed of parameters listed in Table 6.34. The MMSC acknowledges the submission request with the `MM7_submit.RES` response. This response is composed of parameters listed in Table 6.35.

Figure 6.38 shows a graphical representation of the message submission request body.

Figure 6.39 shows a graphical representation of the message submission response body.

Figure 6.40 shows an example of submission request embedded in an `HTTP Post` request, whereas Figure 6.41 shows the corresponding response.

6.8.3 Message Delivery

Regarding the MM7 interface, *message delivery* refers to the delivery of a multimedia message from the MMSC to a VAS application. The MMSC may deliver the message to the VAS application along with a *linked identification*. This identification can be conveyed as part of a subsequent message submission from the VAS application to indicate that the submitted message is related to a previously delivered message. The transaction flow in Figure 6.42 shows interactions between the MMSC and the VAS application for the delivery of a multimedia message over the MM7 interface.

The delivery request `MM7_deliver.REQ` is composed of parameters listed in Table 6.36.

Figure 6.43 shows a graphical representation of the message delivery request body.

The VAS application acknowledges the delivery request with the `MM7_deliver.RES` response. This response is composed of parameters listed in Table 6.37.

Figure 6.44 shows a graphical representation of the message delivery response body.

Table 6.34 MM7 message submission request (MM7_submit.REQ)

Parameter name	Description	From 3 GPP	Loc.	St.
TransactionID	Transaction identifier	Rel-5	H	●
MessageType	<root element of the SOAP body> Message type–type of MM7 operation Value: SubmitReq	Rel-5	B	●
MM7Version	Version[1] of the MM7 interface supported by the VAS application; Example: 5.6.0	Rel-5	B	●
VASPID	<sub-element of SenderIdentification> Identification of the organization providing the value-added service (VAS provider)	Rel-5	B	○
VASID	<sub-element of SenderIdentification> Identification of the value-added service	Rel-5	B	○
SenderAddress	<sub-element of SenderIdentification> Address of the message originator	Rel-5	B	○
Recipients	Address of the message recipient(s)–multiple recipient addresses can be specified	Rel-5	B	●
ServiceCode	Service code for charging purpose	Rel-5	B	○
LinkedID	Linked identification–linkage with another message	Rel-5	B	○
MessageClass	Message class; Type: enumeration Values: Informational, Advertisement or Auto	Rel-5	B	○
TimeStamp	Date and time of message submission	Rel-5	B	○
ExpiryDate	Message time of expiry	Rel-5	B	○
EarliestDeliveryTime	Message earliest time of delivery	Rel-5	B	○
DeliveryReport	Request for a delivery report; Values: True or False	Rel-5	B	○
ReadReply	Request for a read report; Values: True or False	Rel-5	B	○
ReplyCharging	Request for reply charging–parameter has two attributes: replyDeadline and replyChargingSize–presence of this parameter means that reply charging is requested	Rel-5	B	○
replyDeadline	<attribute of ReplyCharging> Reply charging deadline Type: absolute or relative date format	Rel-5	B	○
replyChargingSize	<attribute of ReplyCharging> Reply charging max. message size	Rel-5	B	○
Priority	Message priority; Values: High, Normal, or Low	Rel-5	B	○
Subject	Message subject	Rel-5	B	○
DistributionIndicator	Whether or not the message can be redistributed freely Values: True or False	Rel-5	B	○
ChargedParty	Indication of the party(ies) that should be charged for the cost of handling the message; Values: Sender, Recipient, Both, Neither, or Third Party	Rel-5	B	○

(Continued)

Table 6.34 (*Continued*)

Parameter name	Description	From 3 GPP	Loc.	St.
`ChargedPartyID`	Address of the VAS provider which is expected to pay for the delivery of the message	Rel-6	B	○
`DeliveryCondition`	Specify a condition for the MMSC to deliver the message Type: string; Example: • `MMS capable only`: to deliver the message to MMS capable devices only • `HPLMN only`: to deliver the message only to the subscribers of the network operator in charge of the MMS service.	Rel-6	B	○
`ApplicID`	Identification of the recipient application	Rel-6	B	○
`ReplyApplicID`	Identification of an application to which reply messages, delivery reports and read reports are addressed	Rel-6	B	○
`AuxApplicInfo`	Auxiliary application addressing information	Rel-6	B	○
`ContentClass`	Identification of the smallest content class to which the message complies; Values: `text`, `image-basic`, `image-rich`, `megapixel`, `video-basic`, `video-rich`, `content-basic`, or `content-rich`	Rel-6	B	○
`DRMContent`	Whether or not the message contains DRM protected contents; Values: `true` or `false`	Rel-6	B	○
`Content`	Message content	Rel-5	B	○
`allowAdaptations`	attribute of `Content` Whether or not content adaptations are allowed Values: `True` or `False`	Rel-5	B	○
`Content-type`	Message content type	Rel-5	A	●

[1]From Release 6, the MMS version assigned to this parameter is the version of the specification for which the XML MM7 schema has changed most recently. This avoids having to produce a new schema each time [3GPP-23.140] has evolved without schema updates.

Table 6.35 MM7 message submission response (`MM7_submit.RES`)

Parameter name	Description	From 3GPP	Loc.	St.
`TransactionID`	Transaction identifier	Rel-5	H	●
`MessageType`	<root element of the SOAP body> Message type–type of MM7 operation Value: `SubmitRsp`	Rel-5	B	●
`MM7Version`	Version of the MM7 interface supported by the MMSC Example: `5.6.0`	Rel-5	B	●
`MessageID`	Message identifier generated by the MMSC Condition: available only if the MMSC accepts the submission request	Rel-5	B	©
`StatusCode`	<sub-element of `Status`> Status of the corresponding request completion	Rel-5	B	●
`StatusText`	<sub-element of `Status`> Textual description of the status of the request completion	Rel-5	B	○
`Details`	<sub-element of `Status`> Human readable detailed textual description of the corresponding request status	Rel-5	B	○

6.8.4 Message Cancellation

In the context of value-added services, *message cancellation* refers to the possibility for a VAS application to cancel the delivery of a multimedia message. Upon receipt of such a cancel request, the MMSC cancels the delivery of the associated message to all message recipients to whom the associated notification has not yet been sent out. The transaction flow in Figure 6.45 shows interactions over the MM7 interface between the MMSC and the VAS application for the cancellation of the delivery of a multimedia message.

The cancellation request `MM7_cancel.REQ` is composed of parameters listed in Table 6.38.

Figure 6.46 shows a graphical representation of the message cancel request body.

The MMSC acknowledges the cancellation request with the `MM7_cancel.RES` response. This response is composed of parameters listed in Table 6.39.

Figure 6.47 shows a graphical representation of the message cancellation response body.

6.8.5 Message Replacement

In the context of value-added services, *message replacement* refers to the possibility for a VAS application to replace a previously submitted multimedia message prior to its delivery. Upon receipt of such a replacement request, the MMSC replaces the previously submitted message with the new message specified as part of the replacement request. Only messages that have not yet been retrieved or forwarded can be replaced. The replacement request

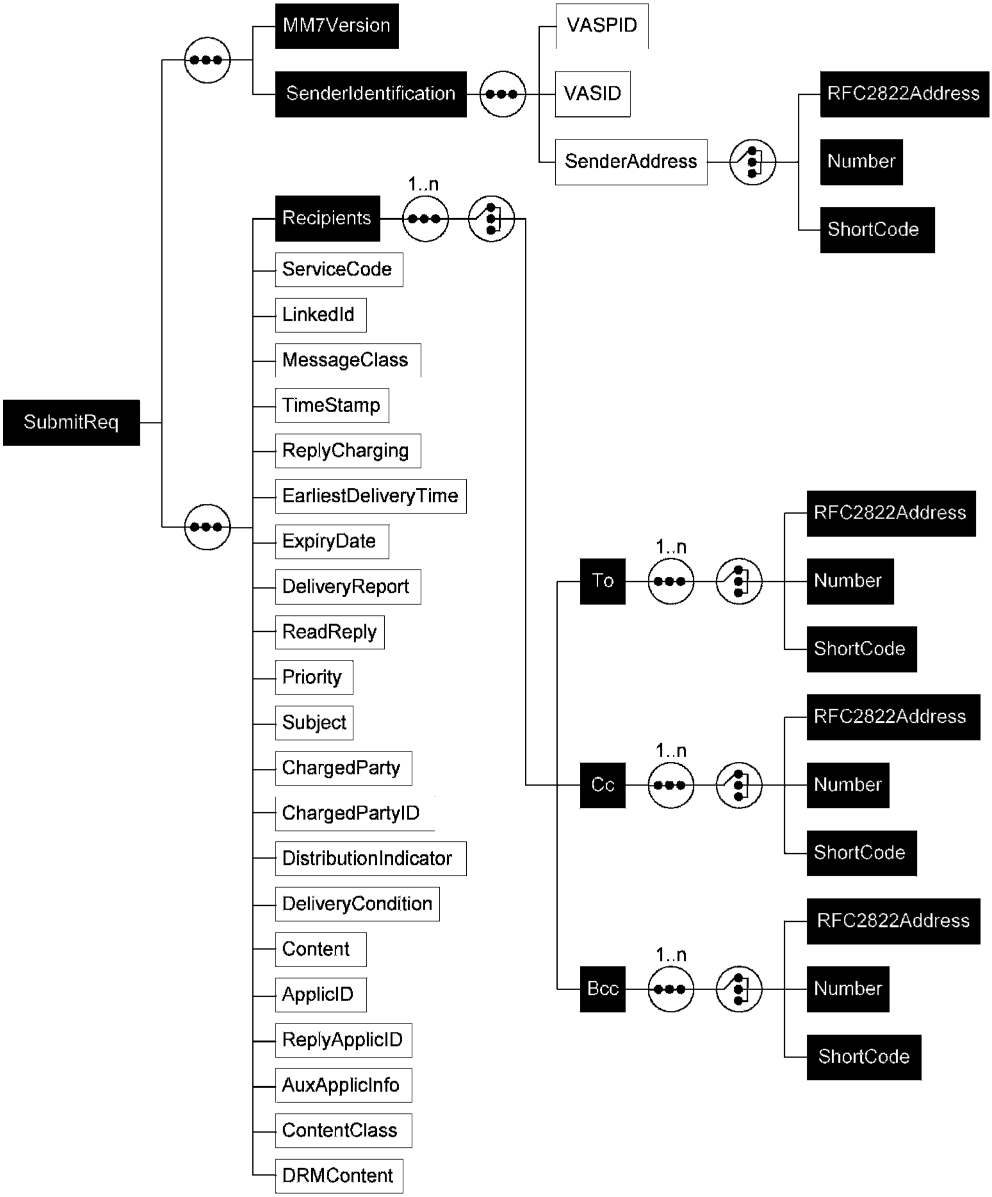


Figure 6.38 Graphical representation of the submission request PDU body

contains a number of parameters that overwrite the ones associated with the previously submitted message. If a parameter was associated with the previously submitted message but is not provided as part of the replacement request, then this parameter is retained for the new message. The transaction flow in Figure 6.48 shows interactions over the MM7 interface between the MMSC and the VAS application for the replacement of a previously submitted multimedia message.

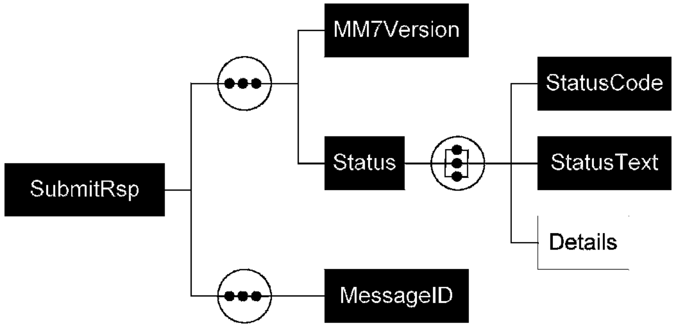


Figure 6.39 Graphical representation of the submission response PDU body

The replacement request MM7_replace.REQ is composed of parameters listed in Table 6.40.

Figure 6.49 shows a graphical representation of the message replacement request body.

The MMSC acknowledges the replacement request with the MM7_replace.RES response. This response is composed of parameters listed in Table 6.41.

The structure of the replacement response body is similar to the one of the cancellation response body. This structure is shown in Figure 6.47.

6.8.6 Delivery Report

In the context of value-added services, a VAS application has the ability to request, as part of a message submission request, the generation of a *delivery report*. If allowed by the message recipient, the MMSC generates a delivery report upon message retrieval, forwarding, deletion, rejection, etc. Note that if the message was submitted to multiple recipients, then several delivery reports may be received by the originator VAS application. The transaction flow in Figure 6.50 shows interactions between the MMSC and the VAS application for the transfer of a delivery report over the MM7 interface.

The request for providing a delivery report MM7_delivery_report.REQ is composed of parameters listed in Table 6.42.

Figure 6.51 shows a graphical representation of the delivery report request body.

The MMSC acknowledges the request with the MM7_delivery_report.RES response. This response is composed of parameters listed in Table 6.43.

Figure 6.52 shows a graphical representation of the delivery report response body.

6.8.7 Read Report

In the context of value-added services, a VAS application has the ability to request, as part of a message submission request, the generation of a *read report*. If allowed by the message recipient, the recipient MMS client generates a read report upon message reading, deletion, etc. Note that, if the message was submitted to multiple recipients, then several read reports

SOAP envelope

SOAP attachment

```
POST /mmsc/mm7 HTTP/1.1
Host: mms.operator.com

Content-Type: Multipart/Related; boundary="PrimaryBoundary"; type=text/xml;
      start="<id_start>"
Content-Length: nnnn
SOAPAction: ""

--PrimaryBoundary
Content-Type: text/xml; charset="utf-8"
Content-ID: <id_start>

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

     <env:Header>
          <mm7:TransactionID xmlns="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-5-MM7-1-3"
          env:mustUnderstand="1" />
               vasp0324-sub
          </mm7:TransactionID>
     </env:Header>

     <env:Body>
          <mm7:SubmitReq xmlns:mm7="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-5-MM7-1-3">
               <MM7Version>5.6.0</MM7Version>
               <SenderIdentification>
                    <VASPID>w-up</VASPID>
                    <VASID>Weather</VASID>
               </SenderIdentification>
               <Recipients>
                    <To>
                         <Number>0147287355</Number>
                         <RFC2822Address>0147287366@operator.com</RFC2822Address>
                    </To>
                    <Cc>
                         <Number>0147243477</Number>
                    </Cc>
                    <Bcc>
                         <Number>0147243488</Number>
                    </Bcc>
               </Recipients>
               <ServiceCode>weather-france</ServiceCode>
               <LinkID>mms00000006</LinkID>
               <MessageClass>Informational</MessageClass>
               <TimeStamp>2002-05-05T09:30:45-01:00</TimeStamp>
               <EarliestDeliveryTime>2002-05-05T09:30:45-01:00</EarliestDeliveryTime>
               <ExpiryDate>P5D</ExpiryDate>
               <DeliveryReport>True</DeliveryReport>
               <Priority>Normal</Priority>
               <Subject>Weather for today</Subject>
               <Content href="cid:weatherpic@w-up.com" ; allowAdptations="true" />
          </mm7:SubmitReq>
     </env:Body>
</env:Envelope>

--PrimaryBoundary
Content-Type: multipart/mixed; boundary="SecondaryBoundary"
Content-ID: <weatherpic@w-up.com>

--SecondaryBoundary
Content-Type: text/plain; charset="us-ascii"

The weather picture.

--SecondaryBoundary
Content-Type: image/gif
Content-ID: <weather.gif>
Content-Transfer-Encoding: base64

R0IGODfg...

--SecondaryBoundary--
--PrimaryBoundary--
```

Figure 6.40 Example of message submission request over the MM7 interface

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf8"
Content-Length: nnnn

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/soap-envelope">
    <env:Header>
        <mm7:TransactionID xmlns:mm7="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-5-MM7-1-3"
        env:mustUnderstand="1">
            vasp0324-sub
        </mm7:TransactionID>
    </env:Header>
    <env:Body>
        <mm7:SubmitRsp xmlns:mm7="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-5-MM7-1-3">
            <MM7Version>5.6.0</MM7Version>
            <Status>
                <StatusCode>1000</StatusCode>
                <StatusText>Message sent</StatusText>
                <MessageID>123456</MessageID>
            </Status>
        </mm7:SubmitRsp>
    </env:Body>
</env:Envelope>
```

Figure 6.41 Example of message submission response over the MM7 interface

may be received by the originator VAS application. The transaction flow in Figure 6.53 shows interactions between the MMSC and the VAS application for the transfer of a read report over the MM7 interface.

The request for providing a read-report `MM7_read_reply_report.REQ` is composed of parameters listed in Table 6.44.

The MMSC acknowledges the request with the `MM7_read_reply_report.RES` response. This response is composed of parameters listed in Table 6.45.

Body structures of the read-report request and response are similar to the ones of the delivery report request and response. Consequently, graphical representations of read-report request and response body structures are, respectively, shown in Figures 6.51 (no `MMStatusExtension` for the read report) and 6.52.

6.8.8 Generic Error Handling

In the situation where the MMSC or the VAS application receives a request, which it cannot process, a generic error notification can be used. The generic error notification always contains the identification of the corresponding request (value assigned to the `Transac-`

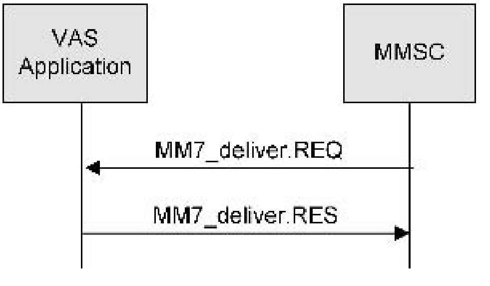


Figure 6.42 MM7 message delivery

Table 6.36 MM7 message delivery request (MM7_deliver.REQ)

Parameter name	Description	From 3GPP	Loc.	St.
TransactionID	Transaction identifier	Rel-5	H	●
MessageType	<root element of the SOAP body> Message type–type of MM7 operation Value: DeliverReq	Rel-5	B	●
MM7Version	Version of the MM7 interface supported by the MMSC Example: 5.6.0	Rel-5	B	●
VASPID	Identification of the organization providing the value-added service (VAS provider)	Rel-6	B	○
VASID	Identification of the value-added service	Rel-6	B	○
MMSRelayServerID	Identification of the MMSC	Rel-5	B	○
Sender	Address of the message sender	Rel-5	B	●
Recipients	Address of the message recipient(s)– multiple recipient addresses can be specified	Rel-5	B	○
LinkedID	Linked identification–linkage with another message	Rel-5	B	○
TimeStamp	Date and time of message submission	Rel-5	B	○
ReplyChargingID	Identification of the reply charging transaction associated to the reply message	Rel-5	B	○
Priority	Message priority/Values: High, Normal, or Low	Rel-5	B	○
Subject	Message subject	Rel-5	B	○
Previouslysentby	Addresses of users who previously sent or forwarded the message	Rel-6	B	○
Previouslysentda-teandtime	Date(s) and time(s) when the message was submitted and forwarded	Rel-6	B	○
SenderSPI	Identification of the service provider (e.g., network operator) originating the message	Rel-6	B	○
RecipientSPI	Identification of the service provider (e.g., network operator) delivering the message to the VAS application	Rel-6	B	○
ApplicID	Identification of the recipient application	Rel-6	B	○
ReplyApplicID	Identification of an application to which reply messages, delivery reports, and read reports are addressed	Rel-6	B	○
AuxApplicInfo	Auxiliary application addressing information	Rel-6	B	○
Content	Message content	Rel-5	B	○
Content-type	Message content type	Rel-5	A	●

tionID parameter) to which it relates. Two generic error notifications can be used as shown in the transaction flows in Figure 6.54.

The MMSC notifies a VAS application of a generic error with the MM7_RS_error.RES response. This response is composed of parameters listed in Table 6.46.

The VAS application notifies the MMSC of a generic error with the MM7_VASP_error.RES response. This response is composed of parameters listed in Table 6.47.

Bodies of the two generic error responses have the same structure as graphically represented in Figure 6.55.

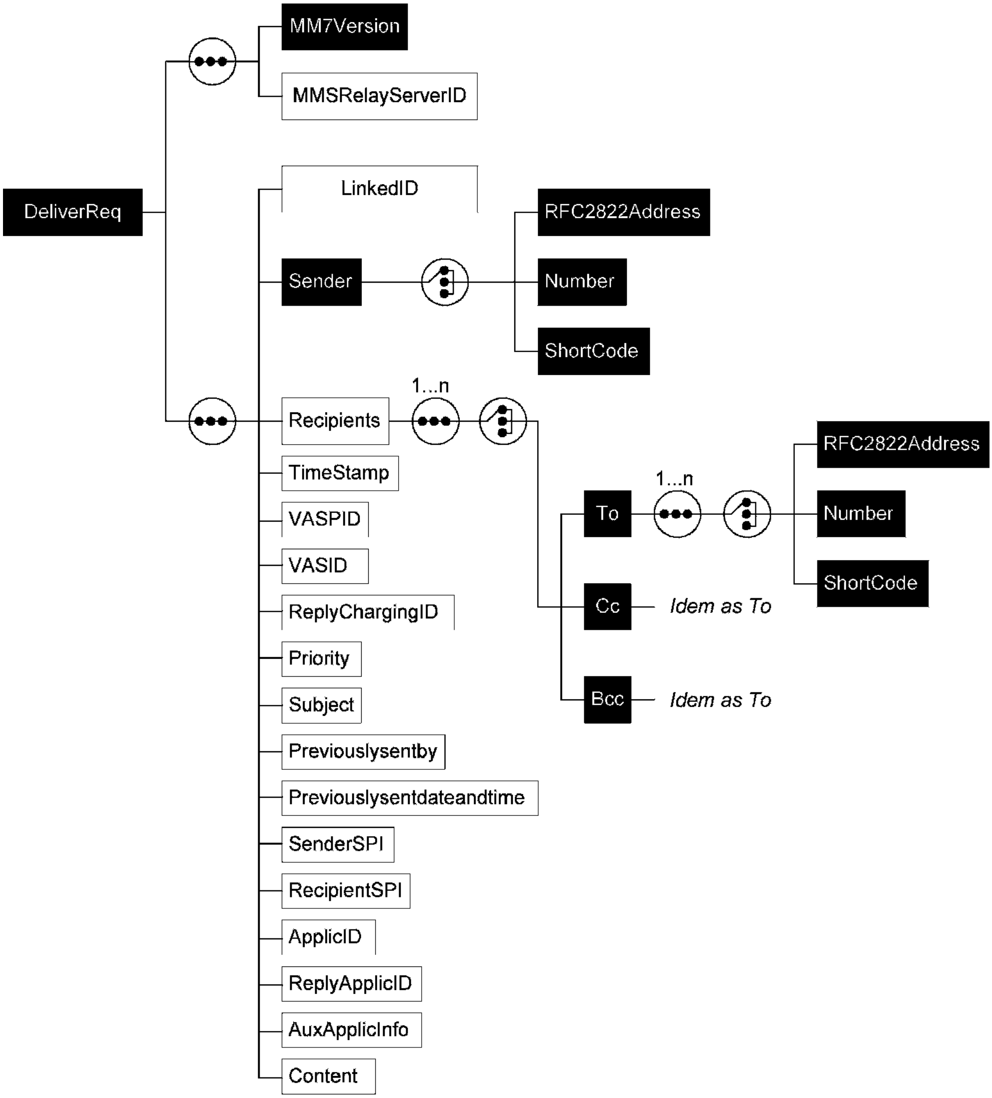


Figure 6.43 Graphical representation of the delivery request PDU body

6.9 MM8 Interface, MMSC–Post-Processing Billing System

The *MM8 interface* interconnects the MMSC and the billing system. 3GPP has standardized charging data records for MMS (see Section 5.23), but a mechanism for conveying these charging data records from the MMSC to the billing system is still to be standardized. Consequently, network operators use existing standard transport protocols or proprietary transport protocols for this purpose.

Table 6.37 MM7 message delivery response (`MM7_deliver.RES`)

Parameter name	Description	From 3GPP	Loc.	St.
`TransactionID`	Transaction identifier	Rel-5	H	●
`MessageType`	<root element of the SOAP body> Message type–type of MM7 operation Value: `DeliverRsp`	Rel-5	B	●
`MM7Version`	Version of the MM7 interface supported by the VAS application/Example: `5.6.0`	Rel-5	B	●
`ServiceCode`	Service code for charging purpose	Rel-5	B	○
`StatusCode`	<sub-element of `Status`> Status of the corresponding request completion	Rel-5	B	●
`StatusText`	<sub-element of `Status`> Textual description of the status of the request completion	Rel-5	B	○
`Details`	<sub-element of `Status`> Human readable detailed textual description of the corresponding request status	Rel-5	B	○

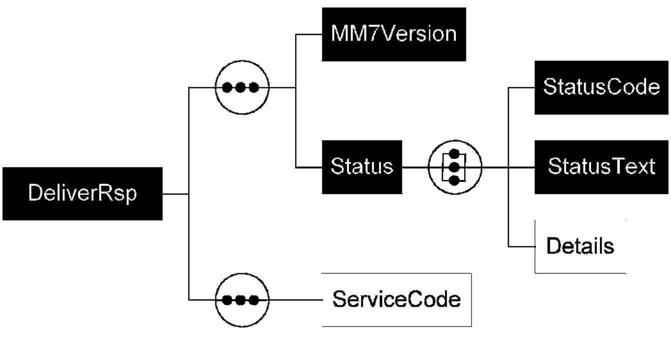


Figure 6.44 Graphical representation of the delivery response PDU body

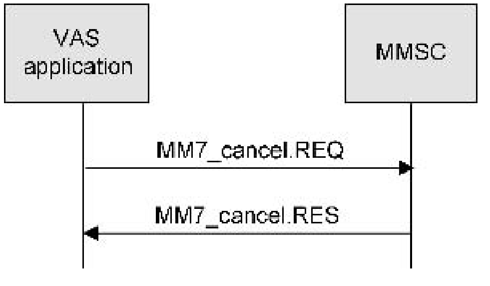


Figure 6.45 MM7 message cancellation

Table 6.38 MM7 message cancel request (MM7_cancel.REQ)

Parameter name	Description	From 3GPP	Loc.	St.
TransactionID	Transaction identifier	Rel-5	H	●
MessageType	<root element of the SOAP body> Message type–type of MM7 operation Value: CancelReq	Rel-5	B	●
MM7Version	Version of the MM7 interface supported by the VAS application/Example: 5.6.0	Rel-5	B	●
VASPID	<sub-element of SenderIdentification> Identification of the organization providing the value-added service (VAS provider)	Rel-5	B	○
VASID	<sub-element of SenderIdentification> Identification of the value added service	Rel-5	B	○
SenderAddress	<sub-element of SenderIdentification> Address of the message originator	Rel-5	B	○
MessageID	Identifier of the message for which the delivery is to be canceled. This identifier was provided by the MMSC in the response of the associated submission request	Rel-5	B	●

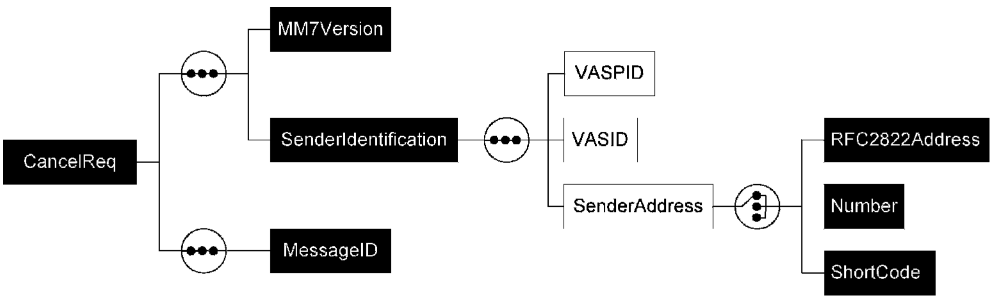


Figure 6.46 Graphical representation of the cancel request PDU body

Table 6.39 MM7 message cancellation response (MM7_cancel.RES)

Parameter name	Description	From 3GPP	Loc.	St.
TransactionID	Transaction identifier	Rel-5	H	●
MessageType	<root element of the SOAP body> Message type–type of MM7 operation Value: CancelRsp	Rel-5	B	●
MM7Version	Version of the MM7 interface supported by the VAS application/Example: 5.6.0	Rel-5	B	●
StatusCode	<sub-element of Status> Status of the corresponding request completion	Rel-5	B	●
StatusText	<sub-element of Status> Textual description of the status of the request completion	Rel-5	B	○
Details	<sub-element of Status> Human readable detailed textual description of the corresponding request status	Rel-5	B	○

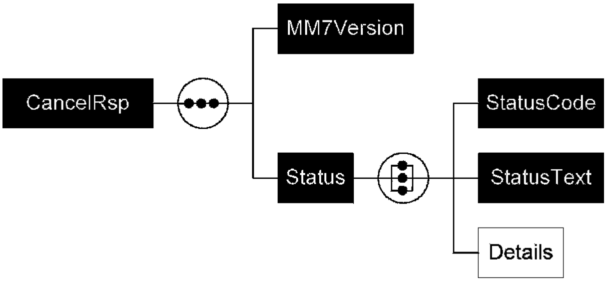


Figure 6.47 Graphical representation of the cancel response PDU body

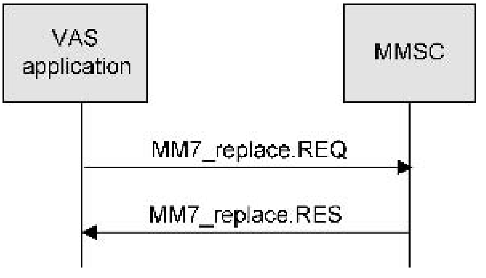


Figure 6.48 MM7 message replacement

6.10 MM9 Interface, MMSC – Online Charging System

The *MM9 interface* enables interactions between the MMSC and an online charging system. With this interface, the MMSC can check whether prepaid customers have sufficient funds in their prepaid account to consume requested services. As for the MM8 interface, the MM9 interface is still to be standardized. Consequently, network operators use existing IP-based protocols (e.g., DIAMETER) or proprietary transport protocols for this purpose.

6.11 MM10 Interface, MMSC – Messaging Service Control Function

The *MM10 interface* allows interactions between the MMSC and a platform implementing a *Messaging Service Control Function* (MSCF). The MMSC requests the MSCF to execute some message-specific service logic that may influence the addressing, routing, and charging of multimedia messages. The MSCF can also access rights for users. This interface is in the process of being standardized but no standard technical realization is available yet.

6.12 STI and MMS Transcoding

It is shown in the preceding chapter that adapting the message contents to the capabilities of the recipient devices can greatly improve the overall user experience. For this purpose, the MMS center may be provided with built-in transacoding capabilities. Owing to the growing complexity of handling high numbers of media types and regarding the high processing

Table 6.40 MM7 message replace request (MM7_replace.REQ)

Parameter name	Description	From 3GPP	Loc.	St.
TransactionID	Transaction identifier	Rel-5	H	●
MessageType	<root element of the SOAP body> Message type–type of MM7 operation Value: ReplaceReq	Rel-5	B	●
MM7Version	Version of the MM7 interface supported by the VAS application/Example: 5.6.0	Rel-5	B	●
VASPID	<sub-element of SenderIdentification> Identification of the organization providing the value-added service (VAS provider)	Rel-5	B	○
VASID	<sub-element of SenderIdentification> Identification of the value-added service	Rel-5	B	○
SenderAddress	<sub-element of SenderIdentification> Address of the message originator	Rel-5	B	○
MessageID	Identification of the message to be replaced. This identification was provided by the MMSC in the response of the associated submission request	Rel-5	B	●
ServiceCode	Service code for charging purpose	Rel-5	B	○
TimeStamp	Date and time of message submission	Rel-5	B	○
EarliestDeli-veryTime	Message earliest request	Rel-5	B	○
ReadReply	Request for a read report/Values: True or False	Rel-5	B	○
allowAdapta-tions	<attribute of Content> Whether or not content adaptations are allowed/ Values: True or False	Rel-5	B	○
DistributionIn-dicator	Whether or not the message can be redistributed freely Values: True or False	Rel-5	B	○
ApplicID	Identification of the recipient application	Rel-6	B	○
ReplyApplicID	Identification of an application to which reply messages, delivery reports, and read reports are addressed	Rel-6	B	○
AuxApplicInfo	Auxiliary application addressing information	Rel-6	B	○
ContentClass	Identification of the smallest content class to which the message complies/Values: text, image-basic, image-rich, megapixel, video-basic, video-rich, content-basic, or content-rich	Rel-6	B	○
DRMContent	Whether or not the message contains DRM protected contents/Values: true or false	Rel-6	B	○
Content	Message content	Rel-5	B	○
Content-type	Message content type Condition: if a message content is available then the content-type must also be present in the PDU	Rel-5	A	©

requirement to convert media types such as video clips, it has become necessary for the MMS center to delegate transcoding activities to a dedicated platform called *transcoder*. Such a transcoder is usually not dedicated to the transcoding of contents for MMS and can be used by other multimedia application platforms such as WAP gateways adapting the contents of browsing traffic, etc. OMA has published the specifications of a *Standard Transcoding*

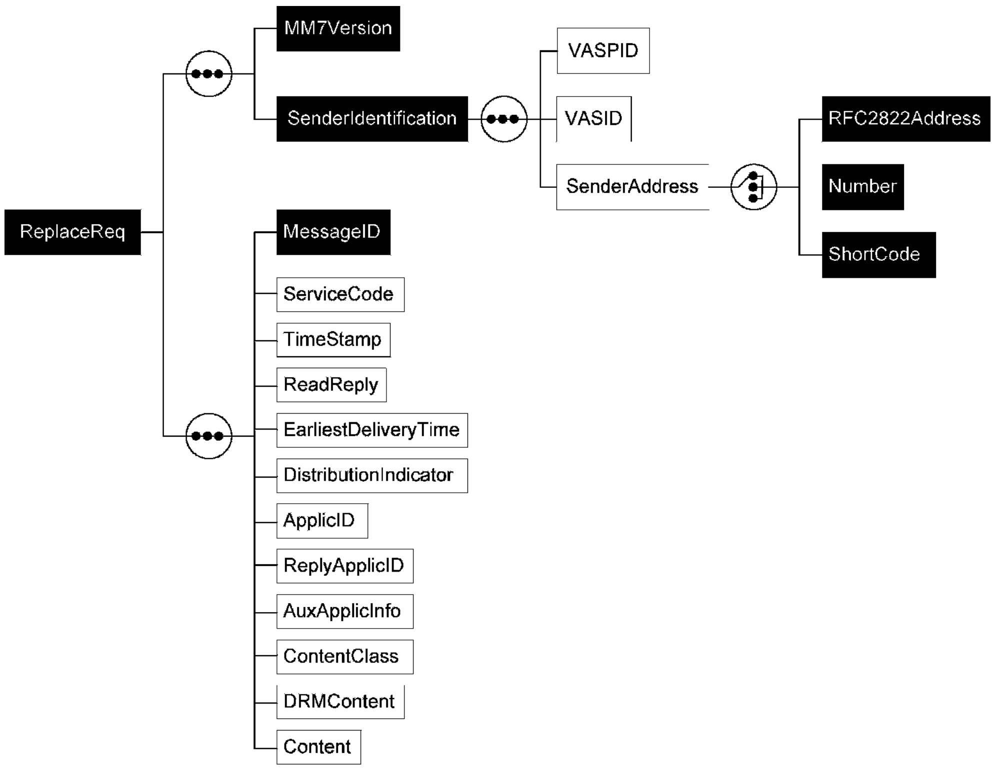


Figure 6.49 Graphical representation of the replace request PDU body

Table 6.41 MM7 message replace response (`MM7_replace.RES`)

Parameter name	Description	From 3GPP	Loc.	St.
`TransactionID`	Transaction identifier	Rel-5	H	●
`MessageType`	<root element of the SOAP body> Message type–type of MM7 operation Value: `ReplaceRsp`	Rel-5	B	●
`MM7Version`	Version of the MM7 interface supported by the VAS application/Example: `5.6.0`	Rel-5	B	●
`StatusCode`	<sub-element of `Status`> Status of the corresponding request completion	Rel-5	B	●
`StatusText`	<sub-element of `Status`> Textual description of the status of the request completion	Rel-5	B	○
`Details`	<sub-element of `Status`> Human readable detailed textual description of the corresponding request status	Rel-5	B	○

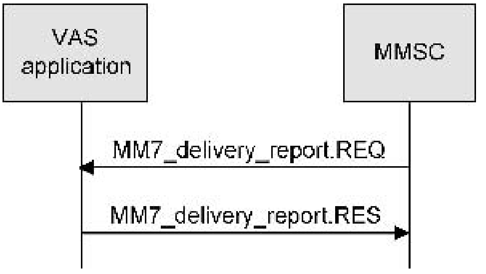


Figure 6.50 MM7 delivery report

Table 6.42 MM7 message delivery report request (`MM7_delivery_report.REQ`)

Parameter name	Description	From 3GPP	Loc.	St.
`TransactionID`	Transaction identifier	Rel-5	H	●
`MessageType`	root element of the SOAP body Message type–type of MM7 operation Value: `DeliveryReportReq`	Rel-5	B	●
`MM7Version`	Version of the MM7 interface supported by the MMSC Example: `5.6.0`	Rel-5	B	●
`MMSRelayServer-ID`	Identification of the MMSC	Rel-5	B	○
`MessageID`	Identification of the message to which the delivery report relates	Rel-5	B	●
`Recipient`	Address of the message recipient	Rel-5	B	●
`Sender`	Address of the VAS application which previously submitted the message	Rel-5	B	●
`TimeStamp`	Date and time of associated message was handled	Rel-5	B	○
`MMStatus`	Delivery status of the associated message Values: `Expired`, `Retrieved`, `Rejected`, `Indeterminate`, `Forwarded`, `Unrecognised`, `Deferred`, or `DeliveryConditionNotMet`	Rel-5	B	●
`StatusText`	Textual description of the status of the request completion	Rel-5	B	○
`MMStatusExten-sion`	Extension for the message status Values: • `RejectionByMMSRecipient`: the recipient refused to receive the message • `RejectionByOtherRS`: the recipient MMSC refused to deliver the message to the recipient	Rel-6	B	○
`ApplicID`	Identification of the originator application of the original message	Rel-6	B	○
`ReplyApplicID`	Identification of an application to which reply messages are addressed	Rel-6	B	○
`AuxApplicInfo`	Auxiliary application addressing information	Rel-6	B	○

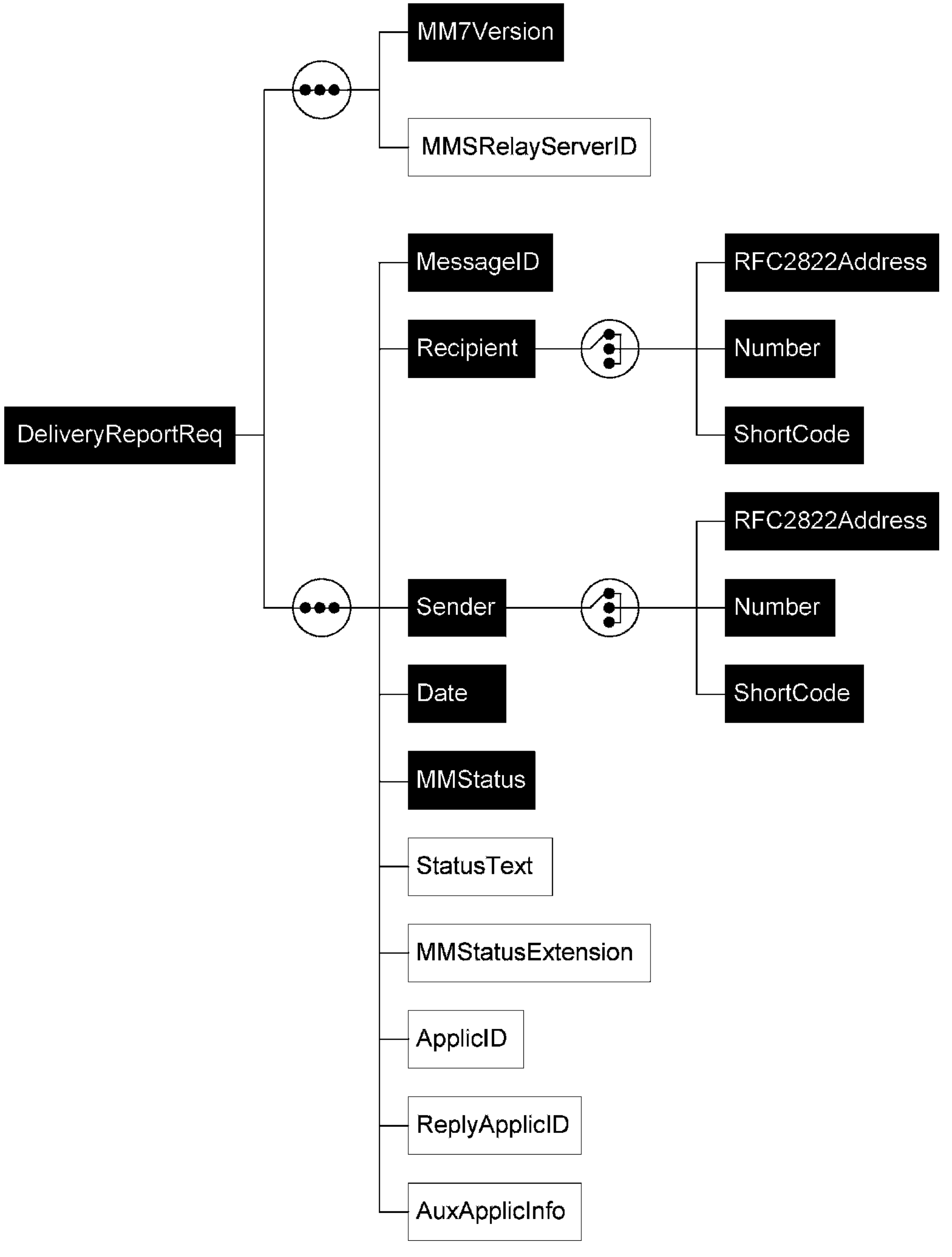


Figure 6.51 Graphical representation of the delivery report request PDU body

Table 6.43 MM7 delivery report response (`MM7_delivery_report.RES`)

Parameter name	Description	From 3GPP	Loc.	St.
`TransactionID`	Transaction identifier	Rel-5	H	●
`MessageType`	<root element of the SOAP body> Message type–type of MM7 operation Value: `DeliveryReportRsp`	Rel-5	B	●
`MM7Version`	Version of the MM7 interface supported by the VAS application/Example: `5.6.0`	Rel-5	B	●
`StatusCode`	<sub-element of `Status`> Status of the corresponding request completion	Rel-5	B	●
`StatusText`	<sub-element of `Status`> Textual description of the status of the request completion	Rel-5	B	○
`Details`	<sub-element of `Status`> Human readable detailed textual description of the corresponding request status	Rel-5	B	○

Interface (STI) [OMA-STI], which defines the way a multimedia application platform shall interact with the transcoder in order to ensure a high level of interoperability between platforms designed by different manufacturers.

Next sections present the transcoding requirements for MMS and describe transactions occurring between an MMS center and a transcoder over the STI interface.

6.12.1 Minor and Major Content Degradations

The preceding chapter presented message content domains and message content classes. MMS capable devices compliant to different classes have different capabilities. If an originator device sends a message containing media objects, which are not understood by the recipient, then it becomes necessary to transcode selected media objects into an appropriate form. In the worst case, if the object is unconvertible then it may be stripped off from the message. Content adaptation is typically performed during the retrieval process according to the User Agent Profile of the receiving device (see also Section 1.6.4). OMA in

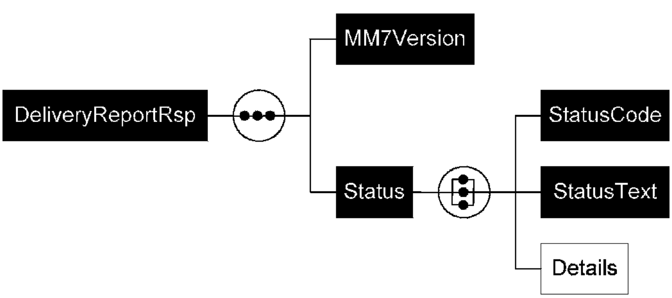


Figure 6.52 Graphical representation of the delivery report response PDU body

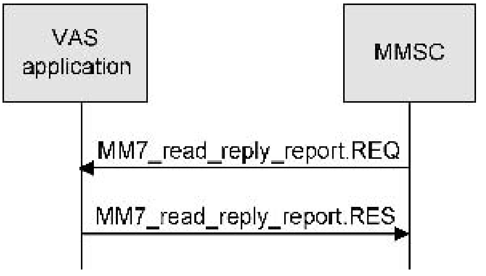


Figure 6.53 MM7 read report

[OMA-ConfDoc] (version 1.2) has categorized content adaptation activities according to the levels of degradation that are applied to media objects:

- *Minor degradation* refers to an activity which mainly adapts the message size, image resolution, sound, and video quality to match the capabilities of the receiving device without degrading significantly the message contents. With minor degradation, the recipient does not perceive a drastic loss of message contents.
- *Major degradation* refers to an activity which changes significantly the message contents, for instance, by removing a media object from the message, converting a video clip into a

Table 6.44 MM7 message read-report request (`MM7_read_reply_report.REQ`)

Parameter name	Description	From 3GPP	Loc.	St.
`TransactionID`	Transaction identifier	Rel-5	H	●
`MessageType`	<root element of the SOAP body> Message type–type of MM7 operation Value: `ReadReplyReq`	Rel-5	B	●
`MM7Version`	Version of the MM7 interface supported by the MMSC Example: `5.6.0`	Rel-5	B	●
`MMSRelayServerID`	Identification of the MMSC	Rel-5	B	○
`MessageID`	Identification of the message to which the read report relates	Rel-5	B	●
`Recipient`	Address of the message recipient	Rel-5	B	●
`Sender`	Address of the VAS application which previously submitted the message	Rel-5	B	●
`TimeStamp`	Date and time of associated message was handled	Rel-5	B	○
`MMStatus`	Delivery status of the associated message Values: `Read`, `Deleted`, or `Indeterminate`	Rel-5	B	●
`StatusText`	Textual description of the status of the request completion	Rel-5	B	○
`ApplicID`	Identification of the originator application of the original message	Rel-6	B	○
`ReplyApplicID`	Identification of an application to which reply messages are addressed	Rel-6	B	○
`AuxApplicInfo`	Auxiliary application addressing information	Rel-6	B	○

Table 6.45 MM7 read-report response (MM7_read_reply_report.RES)

Parameter name	Description	From 3GPP	Loc.	St.
TransactionID	Transaction identifier	Rel-5	H	●
MessageType	<root element of the SOAP body> Message type–type of MM7 operation Value: ReadReplyRsp	Rel-5	B	●
MM7Version	Version of the MM7 interface supported by the VAS application/Example: 5.6.0	Rel-5	B	●
StatusCode	<sub-element of Status> Status of the corresponding request completion	Rel-5	B	●
StatusText	<sub-element of Status> Textual description of the status of the request completion	Rel-5	B	○
Details	<sub-element of Status> Human readable detailed textual description of the corresponding request status	Rel-5	B	○

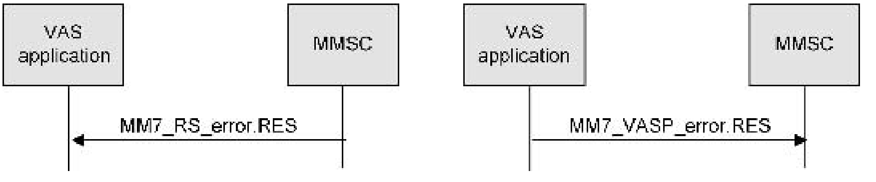


Figure 6.54 MM7 generic errors

Table 6.46 MM7 MMSC error response (MM7_RS_error.RES)

Parameter name	Description	From 3GPP	Loc.	St.
TransactionID	Transaction identifier	Rel-5	H	●
MessageType	<root element of the SOAP body> Message type–type of MM7 operation Value: RSErrorRsp	Rel-5	B	●
MM7Version	Version of the MM7 interface supported by the VAS application/Example: 5.6.0	Rel-5	B	●
StatusCode	<sub-element of Status> Status of the corresponding request completion	Rel-5	B	●
StatusText	<sub-element of Status> Textual description of the status of the request completion	Rel-5	B	○
Details	<sub-element of Status> Human readable detailed textual description of the corresponding request status	Rel-5	B	○

Table 6.47 MM7 VAS application error response (`MM7_VASP_error.RES`)

Parameter name	Description	From 3GPP	Loc.	St.
`TransactionID`	Transaction identifier	Rel-5	H	●
`MessageType`	<root element of the SOAP body> Message type–type of MM7 operation Value: `VASPErrorRsp`	Rel-5	B	●
`MM7Version`	Version of the MM7 interface supported by the VAS application/Example: `5.6.0`	Rel-5	B	●
`StatusCode`	<sub-element of `Status`> Status of the corresponding request completion	Rel-5	B	●
`StatusText`	<sub-element of `Status`> Textual description of the status of the request completion	Rel-5	B	○
`Details`	<sub-element of `Status`> Human readable detailed textual description of the corresponding request status	Rel-5	B	○

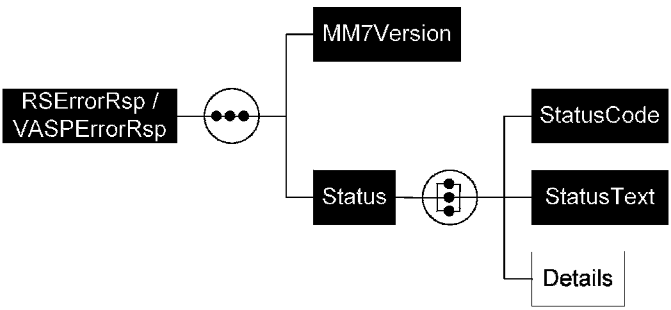


Figure 6.55 Graphical representation of the generic error response PDU bodies

still or an animated image. With major degradation, the recipient typically perceives a significant loss of message contents.

6.12.2 Transcoding Tables

This section presents three transcoding tables indicating for each applicable transcoding activity, whether it is considered as a minor or major adaptation. Table 6.48 presents the

Table 6.48 Audio transcoding

To: From:	AMR-NB	13K	SP-MIDI	General MIDI level 1
AMR-NB	Minor	Minor	N/A	N/A
13K	Minor	Minor	N/A	N/A
SP-MIDI	Major	Major	Major	N/A
General MIDI level 1	Major	Major	Major	Major

Table 6.49 Image transcoding

To: From:	JPEG	GIF87a	GIF89a	WBMP
JPEG	Minor	N/A	N/A	N/A
GIF87a	Minor	Minor	N/A	N/A
GIF89a	Major	Major	Minor	N/A
WBMP	Minor	N/A	N/A	N/A

Table 6.50 Video transcoding

To: From:	H263	MPEG4	JPEG	GIF87a	GIF89a
H263	Minor	N/A	Major	Major	Major
MPEG4	Minor	Minor	Major	Major	Major

audio transcoding. Table 6.49 presents the image transcoding and Table 6.50 presents the video transcoding.

From MMS 1.2, it is mandatory for an MMS system (MMS center with or without external transcoder) to support minor content adaptation. It is optional to support major content adaptation.

The recipient is usually able to retrieve the original message if major content adaptation was applied (e.g., via Web interface). Anyway, the recipient shall always be notified if major content adaptation was applied to the message.

6.12.3 Standard Transcoding Interface

Interactions between the MMS center and a transcoder over the STI rely on a transaction request and a corresponding transaction response as shown in Figure 6.56. The request contains the description for one or more *transaction jobs*. In return, the response contains the results for transaction jobs executed by the transcoder.

As for the MM7 interface, STI relies on SOAP and SOAP messages are encapsulated into `HTTP Post` requests and responses.

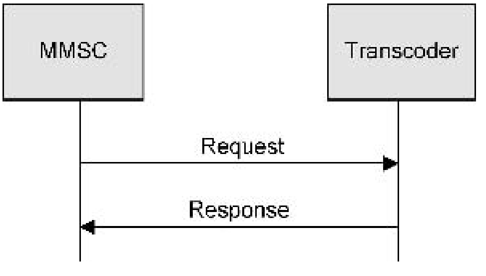


Figure 6.56 STI transaction request/response

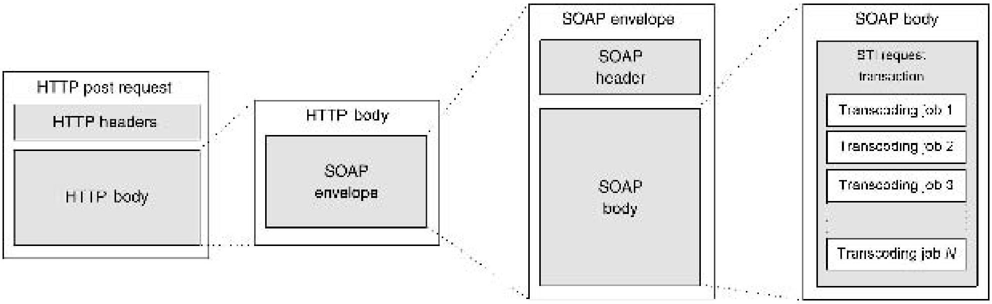


Figure 6.57 STI request transaction without content attachment

6.12.4 STI Request Transaction

The *request transaction* allows the MMSC to request the external transcoder to execute one or more transcoding jobs. The request transaction including the transcoding instructions for each job is encapsulated into the body of a SOAP envelope. The SOAP envelope is inserted in the body of an `HTTP Post` request as shown in Figure 6.57. With this encapsulation, media objects to be transcoded are not included in the request transaction itself; instead transcoding job instructions make references (URL) to an external platform where media objects can be retrieved by the transcoder.

If the transaction is self-contained and also includes media objects to be transcoded, then media objects and the SOAP envelope are included as body parts of a MIME multipart structure as shown in Figure 6.58.

The transcoding job part of the transaction request is composed of two sections: a section describing the *source* (i.e., the media object to be transcoded, e.g., format type, etc.) and another section describing the *target* (i.e., what result is expected, e.g., user agent profile of the receiving device, etc.). As part of the transaction request, the MMS center may refer to a specific transcoding *policy* defining the rules to be followed by the transcoder.

The MMS center can follow two methods for adapting the contents of messages. The first method consists of providing independently media objects to the transcoder and providing specific instructions for each media object contained in the multimedia message. The second

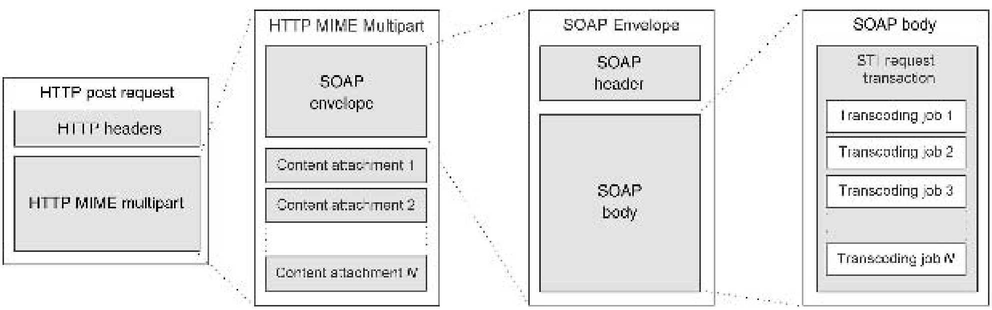


Figure 6.58 STI request transaction with content attachment

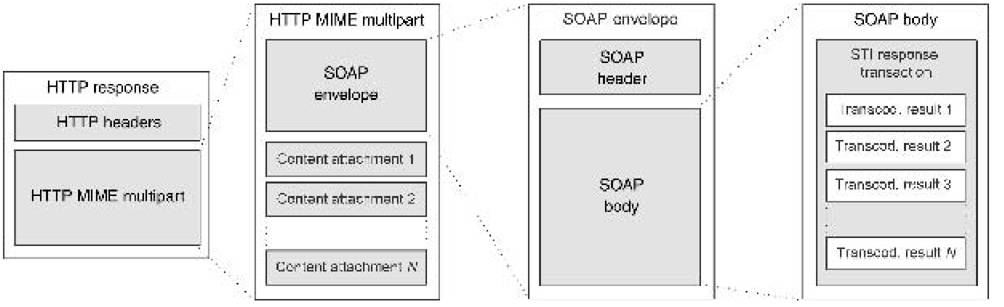


Figure 6.59 STI response transaction

method consists of delegating the entire content adaptation to the transcoder by providing the entire multimedia message as a single transcoding job to the transcoder.

6.12.5 STI Response Transaction

Once a request transaction has been submitted by the MMS center, the transcoder processes the transcoding job(s) and provides the transcoding results back to the MMS center as part of a *response transaction*. This response transaction is encapsulated as part of the `HTTP Response` corresponding to the `HTTP Post` request of the request transaction. The structure of the response transaction is shown in Figure 6.59.

The XML schema specified by OMA for the definition of STI request/response transactions is included as part of the OMA STI standard [OMA-STI].

6.13 Standard Conformance and Interoperability Testing

Standard conformance refers to the adherence of a device implementation to the normative requirements of one or more technical specifications (standards) and *interoperability testing* refers to the process of testing that two or more devices interoperate properly and are in line with the standard requirements.

Once MMS standards have been published, manufacturers can develop devices on the basis of these standards. Of course, devices (MMSC and MMS clients) are implemented with more or less features to match specific market requirements. However, devices claiming conformance to the same standards are expected to interoperate efficiently. This is the prime objective of a standard and a key enabler for the success of any communication service.

Next sections introduce the concepts of standard conformance and interoperability testing. They also describe formal methods that have been set by OMA in order to test interoperability between devices prior to their market launch. The chapter also describes mechanisms that allow devices complying with different versions of the same standard to interoperate in a satisfactory manner.

In March 2004, the Global Certification Forum[1] (GCF) announced that it has initiated the relevant technical work to include an MMS into its certification scheme.

[1]http://gcf.gsm.org

A.2.3 Notification Transaction

Item	Function	Reference	Status	Requirement
MMSCTR-NTF-S-001	Notification Transaction between MMS Proxy-Relay and Receiving MMS Client	7.2	M	MMSCTR-NTF-S-002 AND MMSCTR-NTF-S-003
MMSCTR-NTF-S-002	MMS Proxy-Relay Sending M-Notification.ind to Receiving MMS Client	7.2.1	O	MMSE-S-077 AND MMSCTR-PSH-S-002
MMSCTR-NTF-S-003	Receiving MMS Client Sending M-NotifyResp.ind to MMS Proxy-Relay	7.2.1	O	MMSE-S-077

Figure 6.60 Partial SCR definition – source [OMA-MMS-CTR] (version 1.1)

6.13.1 Static Conformance Requirements

Each OMA specification can include a Static Conformance Requirements (SCRs) definition (appendix section of the standard). An SCR definition for a specification provides the list of mandatory and optional requirements to be fulfilled by the MMS client and/or the MMSC. An SCR definition also expresses dependencies towards other OMA specifications.

Figure 6.60 shows a partial SCR definition extracted from [OMA-MMS-CTR] (version 1.1). This definition identifies optional and mandatory server features for the support of the notification transaction.

An SCR definition is presented as a table with the following columns:

- *Item*: identifier for the feature/requirement (e.g., MMSCTR-NTF-S-001)
- *Function*: short description of the feature (e.g., "Notification Transaction between MMS Proxy-Relay and Receiving MMS Client")
- *Reference*: section number of the OMA specification with more details on the feature (e.g., 7.2)
- *Status*: whether support of the feature is mandatory or optional. "O" stands for optional whereas "M" stands for mandatory
- *Requirement*: other features required by this feature (e.g., MMSCTR-NTF-S-002 AND MMSCTR-NTF-S-003).

The identifier (item) is constructed according to the convention shown in Figure 6.61.

In the context of MMS, the three normative MMS specifications include SCR definitions [OMA-MMS-CTR][OMA-MMS-Enc][OMA-MMS-Conf]. Being informative, [OMA-MMS-Arch] does not contain any SCR definition.

6.13.2 Enabler Implementation Conformance Statement

The Enabler Implementation Conformance Statement (EICS or ICS) is a statement of the capabilities and options supported by a device (MMSC or MMS client) for a given enabler release. It indicates, for an implementation, the set of mandatory and optional features it supports for each specification/standard in an enabler release. The ICS is basically built up from SCR definitions contained in normative documents composing an enabler release. For

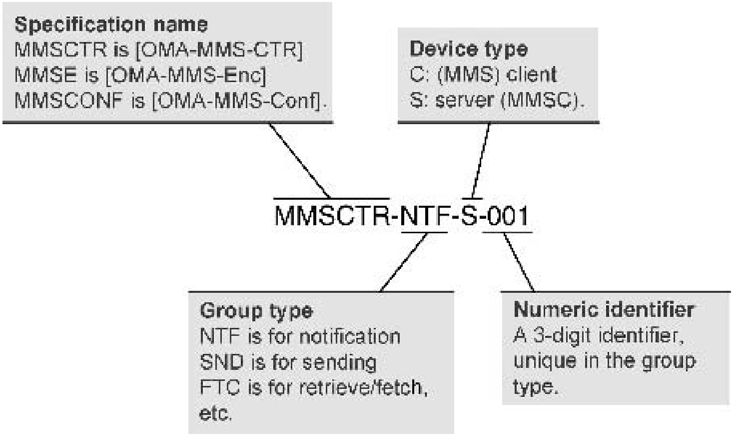


Figure 6.61 Convention for building feature/requirement identifiers

MMS, OMA published an ICS template for the MMS 1.1 enabler release [OMA-MMS-ICS]. It is usually up to the device vendor to complete this template in order to produce the device ICS.

6.13.3 Enabler Test Requirements, Plan, and Specification

As defined in Section 2.9, the OMA IOP working group has the responsibility of testing interoperability and solving identified issues. The OMA IOP sub-working group is in charge of carrying such activities for MMS. For this purpose, this sub-working group produces the following three types of documents:

- *Enabler Test Requirements* (ETRs): the ETR identifies all requirements important enough to warrant attention from an interoperability perspective and identifies any technical features that should be covered by interoperability testing.
- *Enabler Test Plan* (ETP): the ETP defines the test strategy and test methodologies for meeting the requirements in the corresponding ETR. The ETP also defines the scope for testing the corresponding enabler release and identifies the high-level requirements for the test tool, if one is needed.
- *Enabler Test Specification* (ETS): the ETS defines all test cases for the corresponding enabler release, expected inputs and outputs, responses, and behavior for each single test.

Once the ETS has reached a mature stage, the interoperability testing between MMS devices can be organized as defined in the following section.

6.13.4 Interoperability Testing

OMA has selected the following three methods for conducting interoperability testing:

- *OMA hosted test fests*: during OMA hosted test fest, selected device vendors are able to perform one-to-one device testing with other registered MMS devices. Testing is performed against the latest version of the ETS. A number of test reports provide the outcome of the test sessions.

- *Bilateral testing between manufacturers*: this method is similar to an OMA test fest except that it involves only two vendors and tests are usually conducted in the premises of one of the vendors.
- *Testing by an OMA approved test house*: this method consists of mandating a third party to conduct interoperability tests on behalf of OMA members.

Once interoperability tests have been successfully conducted for a given device, then the device vendor can claim that the device conforms to a given MMS enabler release. This of course is a guarantee for the vendor's customers that the device is interoperating efficiently with other devices also conforming to the same enabler release.

6.14 Implementations of Different Versions of the MMS Protocol

As shown in this chapter, each MMS protocol data unit is marked with an MMS version (e.g., versions 1.0 or 1.1). A PDU of higher version may have more parameters than the equivalent PDU marked with a lower version. Furthermore, certain PDU for a given version may not have any equivalent PDU in the protocol of a lower version (e.g., the dedicated PDU for read reports is defined in MMS version 1.1 but is not available in MMS version 1.0). Communicating MMS devices (MMSC and MMS clients) can conform to different versions of MMS (an MMS client conforms to MMS 1.0 communicating with an MMSC conforming to MMS 1.1). Consequently, the MMS standard [OMA-MMS-Enc] (from version 1.1) includes a number of rules for ensuring that such heterogeneous MMS devices interoperate effectively.

The MMS version marking a PDU over the MM1 interface (value assigned to the `X-MMS-Version` parameter) is composed of a major version number and a minor version number as shown in Figure 6.62.

Two scenarios can occur for communicating MMS devices conforming to different versions of the MMS protocol:

1. *Devices conform to the same major version number but to a different minor version number*: an MMS device (MMS client or MMSC) can respond to a received PDU with a PDU marked with a different minor version number but with the same major version number. An MMS device may therefore receive a PDU containing unrecognized

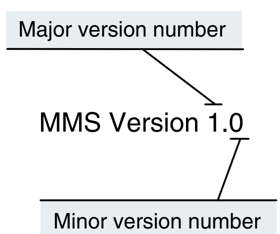


Figure 6.62 Structure of the MMS version

parameters. In this context, a receiving MMS client ignores the unrecognized parameters whereas a receiving MMSC passes unrecognized parameters unchanged (without interpretation). It may happen that the PDU itself is not recognized by the receiving device (unknown value assigned to the `X-MMS-Message-Type` parameter). In this context, the MMS client responds with a `M-NotifyResp.ind` PDU containing a status value set to "`Unrecognised`" whereas the MMSC responds with an `M-Send.conf` PDU with a response value set to "`Error-unsupported-message`."

2. *Devices conform to the different major version numbers*: because the behavior of devices conforming to different major version numbers is it expected to be very different, interoperability between such devices is not ensured. In this context, the MMS client which receives a PDU marked with a major version number which it does not support responds with an `M-NotifyResp.ind` marked MMS version 1.0 with a status value set to "`Unrecognised`." On the other way round, the MMSC which receives a PDU marked with a major version number it does not support responds with the `M-Send.conf` PDU marked MMS version 1.0 with a response status set to "`Error-unsupported-message`." In the case where a receiving device supports multiple major version numbers including the one of the received PDU, it responds to the received PDU with a PDU marked with the same major version number.

Note that, at the time of writing, all MMS devices available on the market (MMSCs and MMS clients) conform to the same major version number which is 1. However, MMS devices conform to protocol versions which can have different minor version numbers (e.g., 1.0, 1.1, 1.2, and now 1.3).

References

Organizations Involved in the Development of Standards and Recommendations

3G Americas	http://www.3gamericas.org/
3G.IP	http://www.3gip.org/
European Telecommunications Standard Institute (ETSI)	http://www.etsi.org/
Fixed Line MMS Forum (F-MMS)	http://www.fixedlinemms.org/
Global Certification Forum (GCF)	http://gcf.gsm.org/
GSM Association (GSMA)	http://www.gsmworld.com/
Global mobile Suppliers Association (GSA)	http://www.gsacom.com/
International Telecommunication Union (ITU)	http://www.itu.org
Internet Engineering Task Force	http://www.ietf.org
IPv6 Forum	http://www.ipv6forum.com/
MIDI Manufacturers Association	http://www.midi.org
Open Mobile Alliance (OMA)	http://www.openmobilealliance.org
SMS Forum	http://www.smsforum.net
Third Generation Partnership Project (3GPP)	http://www.3gpp.org
Third Generation Partnership Project 2 (3GPP2)	http://www.3gpp2.org
UMTS Forum	http://www.umts-forum.org
WAP Forum (now Open Mobile Alliance)	http://www.openmobilealliance.org
Wireless Village (now Open Mobile Alliance)	http://www.openmobilealliance.org
World Wide Web Consortium (W3C)	http://www.w3c.org

Standards

3GPP Documents

[3GPP-01.01] 3GPP TS 01.01, GSM Release 1999 specifications

[3GPP-11.11] 3GPP TR 11.11, Specification of the SIM–ME interface

[3GPP-21.101] 3GPP TS 21.101, 3rd Generation mobile system Release 1999 specifications

[3GPP-21.102] 3GPP TS 21.102, 3rd Generation mobile system Release 4 specifications

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic

ISBN: 0-470-01143-2

[3GPP-21.103] 3GPP TS 21.103, 3rd Generation mobile system Release 5 specifications
[3GPP-21.900] 3GPP TR 21.900, 3GPP working methods
[3GPP-21.905] 3GPP TR 21.905, System aspects, vocabulary for 3GPP specifications
[3GPP-21.910] 3GPP TS 21.910, Multi-mode UE issues: categories, principles, and procedures
[3GPP-22.060] 3GPP TS 22.060, General Packet Radio Service (GPRS), stage 1
[3GPP-22.105] 3GPP TS 22.105, Services and service capabilities
[3GPP-22.121] 3GPP TS 22.121, The Virtual Home Environment (VHE), stage 1
[3GPP-22.127] 3GPP TS 22.127, Open Service Architecture (OSA), stage 1
[3GPP-22.140] 3GPP TS 22.140, Multimedia messaging service, stage 1
[3GPP-22.141] 3GPP TS 22.141, Presence service, stage 1
[3GPP-22.228] 3GPP TS 22.228, Service requirements for the IP multimedia, core network subsystem, stage 1
[3GPP-22.340] 3GPP TS 22.340, IP multimedia system messaging, stage 1
[3GPP-22.940] 3GPP TS 22.940, IMS messaging, stage 1
[3GPP-23.002] 3GPP TS 23.002, Network architecture
[3GPP-23.011] 3GPP TS 23.011, Technical realization of supplementary services
[3GPP-23.038] 3GPP TS 23.038, Alphabets and language-specific information
[3GPP-23.039] 3GPP TR 23.039, Interface protocols for the connection of SMSCs to SMEs
[3GPP-23.040] 3GPP TS 23.040, Technical realization of the short message service
[3GPP-23.042] 3GPP TS 23.042, Compression algorithm for text messaging services
[3GPP-23.060] 3GPP TS 23.060, General Packet Radio Service (GPRS), stage 2
[3GPP-23.101] 3GPP TS 23.101, General UMTS architecture
[3GPP-23.127] 3GPP TS 23.127, Virtual Home Environment/Open Service Architecture, stage 2
[3GPP-23.140] 3GPP TS 23.140, Multimedia messaging service, functional description, stage 2
[3GPP-23.140] 3GPP TS 23.140, Multimedia Messaging Service (MMS), stage 2
[3GPP-23.228] 3GPP TS 23.228, IP multimedia subsystem, stage 2
[3GPP-23.841] 3GPP TS 23.841, Presence service, stage 2
[3GPP-24.011] 3GPP TS 24.011, Point-to-point SMS support on mobile radio interface
[3GPP-26.140] 3GPP TS 26.140, Multimedia messaging service, media formats, and codecs
[3GPP-26.233] 3GPP TS 26.233, Transparent end-to-end packet-switched streaming service, general description
[3GPP-26.234] 3GPP TS 26.234, Transparent end-to-end packet-switched streaming services protocols and codecs
[3GPP-27.005] 3GPP TS 27.005, Use of DTE–DCE interface for CBS
[3GPP-31.102] 3GPP TS 31.102, Characteristics of the USIM application
[3GPP-31.111] 3GPP TS 31.111, USIM Application Toolkit (USAT)
[3GPP-32.200] 3GPP TS 32.200, Telecommunication management, charging management, charging principles
[3GPP-32.235] 3GPP TS 32.235, Telecommunication management, charging management, charging data description for application services
[3GPP-41.102] 3GPP TS 41.102, GSM Release 4 specifications
[3GPP-41.103] 3GPP TS 41.103, GSM Release 5 specifications
[3GPP-43.041] 3GPP TS 43.041, Example protocol stacks for interconnecting SMSCs and MSCs
[3GPP-51.011] 3GPP TS 51.011, Specification of the SIM–ME interface

3GPP2 Documents

[3GPP2-C.P0045] C.P0045–MMS media formats and codecs, 3GPP2, draft
[3GPP2-S.R0064.0] S.R0064-0–Multimedia messaging service, 3GPP2, November 2002
[3GPP2-S0016.000] X.S0016.000–MMS specification overview, 3GPP2, April 2003
[3GPP2-S0016.200] X.S0016.200–MMS stage 2, functional description, 3GPP2, April 2003
[3GPP2-S016.310] X.S0016.310–MMS MM1 stage 3 using OMA/WAP, 3GPP2, April 2003
[3GPP2-S016.340] X.S0016.340–MMS MM4 stage 3 intercarrier interworking, 3GPP2, April 2003
[3GPP2-S016.370] X.S0016.370–MMS MM7 VASP interworking stage 3 specification, 3GPP2, April 2003

ITU Documents

[ITU-E.164] ITU-T E.164, The international public telecommunication number plan, ITU, May 1997

[ITU-E.212] ITU-T E.212, The international identification plan for mobile terminals and mobile users, ITU, November 1998

[ITU-H.263] ITU-T H.263, Video coding for low bit rate communication, ITU, February 1998

[ITU-I.130] ITU-T I.130, Method for the characterization of telecommunication services supported by ISDN and network capabilities of an ISDN, ITU, November 1998

IETF Documents

[RFC-821] Simple Mail Transfer Protocol (SMTP), August 1982

[RFC-822] Standard for the format of ARPA Internet text messages, IETF, August 13, 1982. Note that [RFC-2822] obsoletes [RFC-822]

[RFC-1730] Internet Message Access Protocol (IMAP), version 4, December 1994

[RFC-1766] Tags for the identification of languages, March 1995

[RFC-1806] Communicating presentation information in Internet messages, the Content-Disposition header, IETF, June 1995

[RFC-1889] RTP: a transport protocol for real-time applications, IETF, January 1996

[RFC-1893] Enhanced mail system status codes, IETF, January 1996

[RFC-1939] (STD 0053) Post Office Protocol (POP), version 3, May 1996

[RFC-2026] The Internet standards process–Revision 3, IETF, October 1996

[RFC-2045] Multipurpose Internet mail extensions, part 1: format of Internet message bodies, IETF, November 1996

[RFC-2046] Multipurpose Internet mail extensions, part 2: media types, IETF, November 1996

[RFC-2047] Multipurpose Internet mail extensions, part 3: message header extensions for non-ASCII text, IETF, November 1996

[RFC-2048] Multipurpose Internet mail extensions, part 4: registration procedures, IETF, November 1996

[RFC-2049] Multipurpose Internet mail extensions, part 5: conformance criteria and examples, IETF, November 1996

[RFC-2279] UTF-8, a transformation format of ISO 10646, IETF, January 1998

[RFC-2326] Real-Time Streaming Protocol (RTSP), April 1998

[RFC-2327] Session description protocol, IETF, April 1998

[RFC-2387] The MIME multipart/related content-type, August 1998

[RFC-2392] Content-ID and Message-ID uniform resource locators, IETF, August 1998

[RFC-2396] Uniform Resource Identifiers (URIs): generic syntax, August 1998

[RFC-2557] MIME encapsulation of aggregate documents, IETF, March 1999

[RFC-2616] HyperText Transfer Protocol–HTTP/1.1, June 1999

[RFC-2617] HTTP authentication: basic and digest access authentication, IETF, June 1999

[RFC-2633] S/MIME message specification, version 3, June 1999

[RFC-2821] Simple mail transfer protocol, IETF, April 2001

[RFC-2822] Internet message format, IETF, April 2001

[RFC-2916] E.164 number and DNS, IETF, September 2000

[RFC-3261] Session initiation protocol, IETF, June 2002

[RFC-3267] Real-time transport protocol payload format and file storage format for the AMR and AMR-WB audio codecs, IETF, June 2002

[RFC-3501] Internet message access protocol–version 4 rev1, IETF, March 2003

OMA Document

[OMA-ClientProv] OMA client provisioning enabler release, including:
- Provisioning architecture overview
- Provisioning content

	- Provisioning bootstrap
	- User agent behavior
[OMA-DevMan]	OMA device management enabler release, including:
	- SyncML device management bootstrap
	- Device management conformance requirements
	- Notification initiated session
	- SyncML device management protocol
	- SyncML representation protocol device management usage
	- SyncML device management security
	- SyncML device management standardized objects
	- SyncML device management tree and description
[OMA-DRM]	OMA DRM enabler release
	- Digital rights management
[OMA-DRM-CF]	OMA DRM enabler release
	- Content format
[OMA-MMS-Arch]	OMA multimedia messaging service enabler release
	- Architecture overview
[OMA-MMS-Conf]	OMA multimedia messaging service enabler release
	- Conformance document
[OMA-MMS-CTR]	OMA multimedia messaging service enabler release
	- Client transactions
[OMA-MMS-Enc]	OMA multimedia messaging service enabler release
	- Encapsulation protocol
[OMA-MMS-ICS]	OMA multimedia messaging service enabler release
	- Enabler implementation conformance statement
[OMA-STI]	OMA standard transcoding interface specification
[OMA-UAProf]	OMA user agent profile enabler release
	- User agent profile
[OMA-WSP]	Wireless session protocol

UMTS Forum Documents

[UF-Rep9]	UMTS Forum Report no. 9: The UMTS third generation market–Phase I: structuring the service revenues opportunities, UMTS Forum, October 2000
[UF-Rep13]	UMTS Forum Report no. 13: The UMTS third generation market–Phase II: structuring the service revenue opportunities, UMTS Forum, April 2001

WAP Forum Documents

[WAP-174]	WAP user agent profile, WAP Forum, November 1999
[WAP-190]	Wireless application environment specification, March 2000
[WAP-203]	Wireless session protocol specification, May 2000
[WAP-205]	Multimedia messaging service, architecture overview, WAP Forum, April 2001
[WAP-206]	Multimedia messaging service, client transactions, WAP Forum, June 2001
[WAP-209]	Multimedia messaging service, encapsulation protocol, WAP Forum, June 2001
[WAP-219]	TLS profile and tunneling, WAP Forum, April 2001
[WAP-224]	Wireless transaction protocol, WAP Forum, July 2001
[WAP-229]	Wireless Profiled HTTP (WP-HTTP), WAP Forum, March 2001
[WAP-230]	Wireless session protocol specification, WAP Forum, July 2001
[WAP-237]	Wireless application environment defined media type, WAP Forum, May 2001
[WAP-250]	WAP push architectural overview, WAP Forum, May 2001
[WAP-274]	(draft) WAP MMS architecture overview, version 1.1, WAP Forum, April 2002 (also available from OMA as document titled MMS Architecture)

[WAP-275] (draft) WAP MMS client transactions, version 1.1, WAP Forum, April 2002 (also available from OMA as document titled MMS Client Transactions)

[WAP-276] (draft) WAP MMS encapsulation protocol, version 1.1, WAP Forum, April 2001 (also available from OMA as document titled MMS Encapsulation Protocol)

[WAP-277] XHTML mobile profile, WAP Forum, October 2001

W3C Documents

[W3C-HTML4] (W3C recommendation) HTML 4.01, W3C, December 1999, http://www.w3.org/TR/html4/

[W3C-PNG] (W3C recommendation) Portable network graphics 1.0, W3C, October 1996, http://www.w3.org/TR/REC-png.html

[W3C-SMIL] (W3C recommendation) Synchronized multimedia integration language 2.0, W3C, August 2001, http://www.w3.org/TR/smil20

[W3C-SOAP] (W3C note) Simple object access protocol 1.1, W3C, May 2000, http://www.w3.org/TR/SOAP

[W3C-SOAP-ATT] (W3C note) SOAP messages with attachments, W3C, December 2000, http://www.w3.org/TR/SOAP-attachments

[W3C-sRGB] A standard default color space for the Internet 1.10, W3C, November 1996, http://www.w3.org/Graphics/Color/sRGB

[W3C-SVG] (W3C note) Scalable vector graphics 1.0, W3C, September 2001, http://www.w3.org/TR/SVG

[W3C-XHTML-Basic] (W3C recommendation) XHTML basic profile, W3C, December 2000, http://www.w3.org/TR/xhtml-basic/

Other Documents

[GSMA-IR.34] IR.34–Inter-PLMN backbone guidelines, GSM Association, March 2003

[GSMA-IR.52] IT.52–MMS interworking guidelines, GSM Association, February 2003

[IANA-MIBEnum] Character sets–IANA, http://www.iana.org/assignments/character-sets

[IMC-vCalendar] vCalendar–The electronic calendaring and scheduling format 1.0, Internet Mail Consortium, September 1996, http://www.imc.org/pdi/vcal-10.doc

[IMC-vCard] vCard–The electronic business card format 2.1, Internet Mail Consortium, September 1996, http://www.imc.org/pdi/vcard-21.doc

[IRDA-iMelody] iMelody, specifications for Ir Mobile Communications, version 1.2: Infrared Association, October 2000

[MMA-MIDI] The complete MIDI 1.0 detailed specification, version 96.1, MIDI Manufacturers Association, 1996

[MMA-SP-MIDI] Scalable polyphony MIDI specification and device profiles, version 1.0a, MIDI Manufacturers Association, May 2002

[US-ASCII] Coded Character Set–7-bit American Standard Code for Information Interchange, ANSI X3.4-1986

Appendices

Appendix A SMS TP-PID Values for Telematic Interworking

For enabling SMS interworking with various telematic devices, the following set of protocol identifiers (`TP-Protocol-Identifier`) can be used:

Table A.1 Protocol identifiers for telematic interworking

TP-PID value (hex)	Description
0x20	Type of telematic device is defined by the message destination or originator address.
0x21	Telex (or teletex reduced to telex format).
0x22	Group 3 telefax.
0x23	Group 4 telefax.
0x24	Voice telephone (i.e. conversion to speech).
0x25	European Radio Messaging System (ERMES).
0x26	National paging system (type known to the service center).
0x27	Videotext such as T.100 or T101.
0x28	Teletex, carrier unspecified.
0x29	Teletex, in PSPDN.
0x2A	Teletex, in CSPDN.
0x2B	Teletex, in analog PSTN.
0x2C	Teletex, in digital ISDN.
0x2D	Universal Computer Interface (UCI).
0x2E...0x2F	Reserved (2 values).
0x30	Message handling facility (type known to the service center).
0x31	Public X.400-based message handling system.
0x32	Internet electronic mail (see section).
0x33...0x37	Reserved (5 values).
0x38...0x3E	SC specific use (7 values).
0x3F	GSM or UMTS mobile station. The SMSC converts the short message into a coding scheme which is understandable by the GSM/UMTS mobile station.

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2

Appendix B SMS–Numeric and Alphanumeric Representations

Various numeric values can be assigned to the parameters of an SMS TPDU. In this context, numeric values can be represented in three different ways:

- Integer representation
- Octet representation
- Semi-octet representation.

B.1 Integer Representation

With the integer representation, a numeric value is represented with one or more octets (complete or in fractions). For such a representation, the following rules apply:

1st rule: octets with the lowest octet indices contain the most significant bits.
2nd rule: bits with the highest bit indices are the most significant bits.

The example below shows how the decimal number 987,351 is represented:

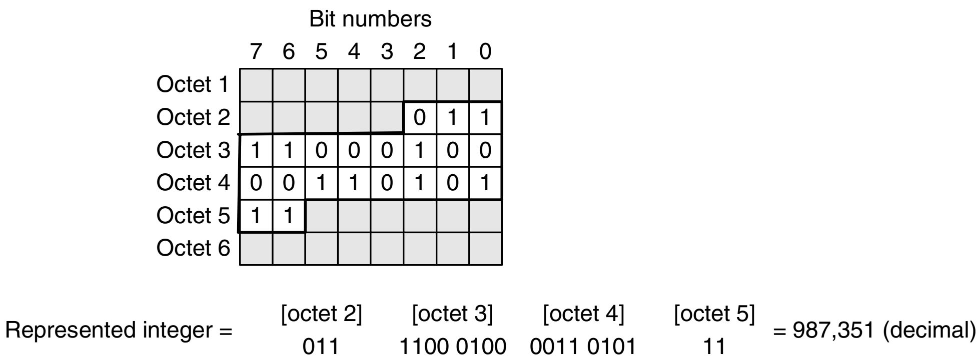


Figure B.1 Integer representation/example

B.2 Octet Representation

With the octet representation, a numeric value is represented with one or more complete octets where each octet represents one decimal digit. The only rule to apply is that octets with the lowest octet indices contain the most significant decimal digits. Each octet can take the following values:

Table B.1 Octet representation

Octet value	Decimal digit
0000 0000	0
0000 0001	1
0000 0010	2
0000 0011	3
0000 0100	4
0000 0101	5
0000 0110	6
0000 0111	7
0000 1000	8
0000 1001	9

All other octet values are reserved. The example below shows how the decimal value 43 is represented:

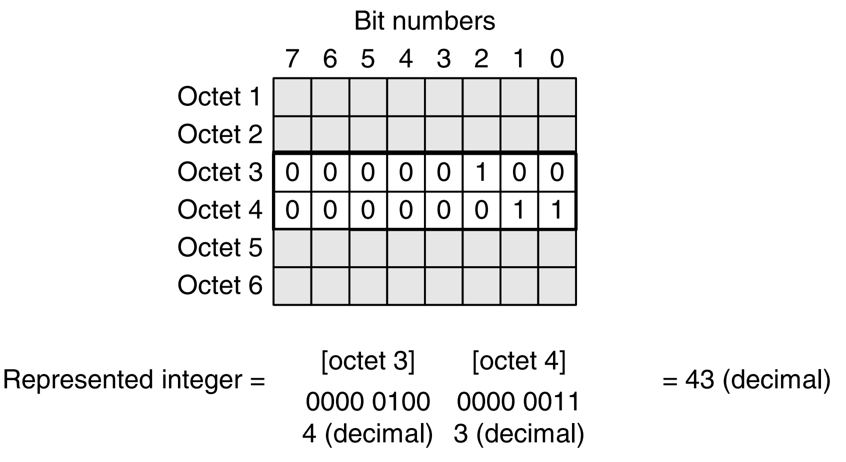


Figure B.2 Octet representation/example

B.3 Semi-Octet Representation

With the semi-octet representation, a numeric value is represented with one or more half-octets (4 bits). For such a representation, the following rules apply:

1st rule: octets with the lowest octet indices contain the most significant decimal digits.
2nd rule: within one octet, the half-octet with bits numbered 0–3 represents the most significant digit.

Each half-octet can take the following values:

Table B.2 Semi-octet representation

Half-octet value	Decimal digit
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	*
1011	#
1100	a
1101	b
1110	c
1111	Used as fill bits

The example below shows how the decimal value 431 is represented with four semi-octets:

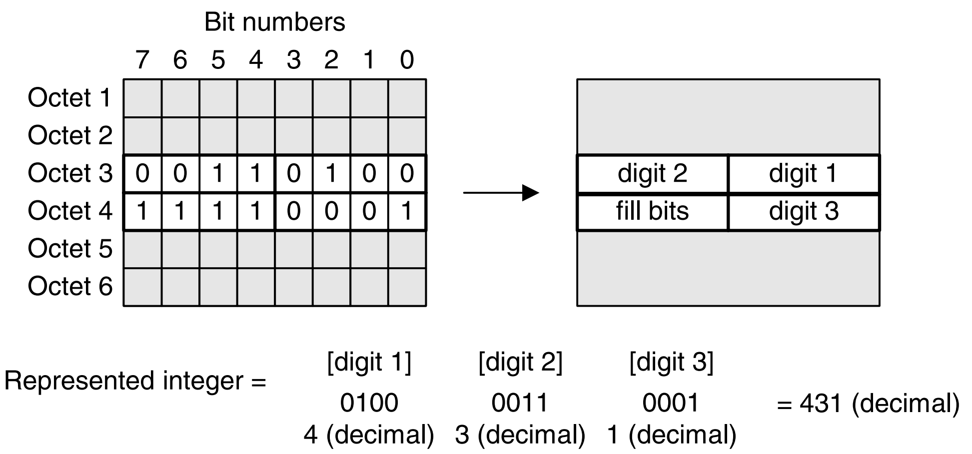


Figure B.3 Semi-octet representation/example

Appendix C SMS–Character Sets and Transformation Formats

C.1 GSM 7-bit Default Alphabet

Table C.1 below presents all the characters in the GSM 7 bits alphabet. Each character is represented with a septet (7 bits) for which the most significant bit is *b7* and the least significant bit is *b1*.

Table C.1 GSM 7 bit alphabet (first table)

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	**b3**	**b2**	**b1**		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	**0**	@	Δ	SP	0	¡	P	¿	p
0	0	0	1	**1**	£	–	!	1	A	Q	a	q
0	0	1	0	**2**	$	Φ	"	2	B	R	b	r
0	0	1	1	**3**	¥	Γ	#	3	C	S	c	s
0	1	0	0	**4**	è	Λ	¤	4	D	T	d	t
0	1	0	1	**5**	é	Ω	%	5	E	U	e	u
0	1	1	0	**6**	ù	Π	&	6	F	V	f	v
0	1	1	1	**7**	ì	Ψ	'	7	G	W	g	w
1	0	0	0	**8**	ò	Σ	(	8	H	X	h	x
1	0	0	1	**9**	ç	Θ	)	9	I	Y	i	y
1	0	1	0	**10**	LF	Ξ	*	:	J	Z	j	z
1	0	1	1	**11**	Ø	Esc*	+	;	K	Ä	k	ä
1	1	0	0	**12**	ø	Æ	,	<	L	Ö	l	ö
1	1	0	1	**13**	CR	æ	−	=	M	Ñ	m	ñ
1	1	1	0	**14**	Å	ß	.	>	N	Ü	n	ü
1	1	1	1	**15**	å	É	/	?	O	§	o	à

Esc*: the escape character indicates that the following character corresponds to an entry in the GSM 7 bits default alphabet extension table as defined below.

Table C.2 GSM 7 bit alphabet (extension table)

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	**b3**	**b2**	**b1**		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	**0**					|			
0	0	0	1	**1**								
0	0	1	0	**2**								
0	0	1	1	**3**								
0	1	0	0	**4**		^						
0	1	0	1	**5**							€	
0	1	1	0	**6**								
0	1	1	1	**7**								
1	0	0	0	**8**			{					
1	0	0	1	**9**			}					
1	0	1	0	**10**	Page break							
1	0	1	1	**11**		Esc*						
1	1	0	0	**12**				[				
1	1	0	1	**13**				~				
1	1	1	0	**14**				]				
1	1	1	1	**15**			\					

Esc*: the escape character indicates that the following character corresponds to an entry in an additional GSM 7 bits default alphabet extension table. At the time of writing this book, such a table had not been defined.

C.2 US-ASCII

The US-ASCII character set [US-ASCII] is widely used for representing text in the Internet domain. This character set can be used for representing the text part of multimedia messages in MMS. The 128 characters composing the US-ASCII character set are listed below:

Table C.3 US-ASCII character set

Decimal	0x00 Hexadecimal	Character	Decimal	0x00 Hexadecimal	Character	Decimal	0x00 Hexadecimal	Character
0	0x00	NUL	43	0x2b	+	86	0x56	V
1	0x01	SOH	44	0x2c	,	87	0x57	W
2	0x02	STX	45	0x2d	−	88	0x58	X
3	0x03	ETX	46	0x2e	.	89	0x59	Y
4	0x04	EOT	47	0x2f	/	90	0x5a	Z
5	0x05	ENQ	48	0x30	0	91	0x5b	[
6	0x06	ACK	49	0x31	1	92	0x5c	\
7	0x07	BEL	50	0x32	2	93	0x5d	]
8	0x08	BS	51	0x33	3	94	0x5e	^
9	0x09	HT	52	0x34	4	95	0x5f	_
10	0x0a	LF	53	0x35	5	96	0x60	‘
11	0x0b	VT	54	0x36	6	97	0x61	a
12	0x0c	FF	55	0x37	7	98	0x62	b
13	0x0d	CR	56	0x38	8	99	0x63	c
14	0x0e	SO	57	0x39	9	100	0x64	d
15	0x0f	SI	58	0x3a	:	101	0x65	e
16	0x10	DLE	59	0x3b	;	102	0x66	f
17	0x11	DC1	60	0x3c	<	103	0x67	g
18	0x12	DC2	61	0x3d	=	104	0x68	h
19	0x13	DC3	62	0x3e	>	105	0x69	i
20	0x14	DC4	63	0x3f	?	106	0x6a	j
21	0x15	NAK	64	0x40	@	107	0x6b	k
22	0x16	SYN	65	0x41	A	108	0x6c	l
23	0x17	ETB	66	0x42	B	109	0x6d	m
24	0x18	CAN	67	0x43	C	110	0x6e	n
25	0x19	EM	68	0x44	D	111	0x6f	o
26	0x1a	SUB	69	0x45	E	112	0x70	p
27	0x1b	ESC	70	0x46	F	113	0x71	q
28	0x1c	FS	71	0x47	G	114	0x72	r
29	0x1d	GS	72	0x48	H	115	0x73	s
30	0x1e	RS	73	0x49	I	116	0x74	t
31	0x1f	US	74	0x4a	J	117	0x75	u
32	0x20	space	75	0x4b	K	118	0x76	v
33	0x21	!	76	0x4c	L	119	0x77	w
34	0x22	"	77	0x4d	M	120	0x78	x
35	0x23	#	78	0x4e	N	121	0x79	y
36	0x24	$	79	0x4f	O	122	0x7a	z
37	0x25	%	80	0x50	P	123	0x7b	{
38	0x26	&	81	0x51	Q	124	0x7c	|
39	0x27	’	82	0x52	R	125	0x7d	}
40	0x28	(	83	0x53	S	126	0x7e	~
41	0x29	)	84	0x54	T	127	0x7f	delete
42	0x2a	*	85	0x55	U			

C.3 Universal Character Set

The ISO has defined in [ISO-10646] the Universal Character Set (UCS), a multi-octet character set for representing most of the world's writing symbols. Two encoding methods are available for USC which are as follows:

- UCS2: a two-octet per symbol encoding
- UCS4: a four-octet per symbol encoding.

UCS2 and UCS4 are difficult to use in systems on 7-bit and 8-bit transport. To cope with these difficulties, USC transformation formats have been developed. These formats are defined in the following sections.

C.4 UCS Transformation Formats

The most commonly used UCS Transformation Formats (UTF) are:

- UTF8 [RFC-2279]: this transformation format is 8-bit aligned and has the key characteristic of preserving the US-ASCII value range. UTF8 encodes UCS2 and UCS4 with one or more octets per symbol (1–6 octets).
- UTF16: this transformation format is 16-bit aligned. UTF16 transforms UCS4 symbols into pairs of UCS2 values.

Symbols represented by one octet in UTF8 are US-ASCII characters. In this configuration, the most significant bit of the octet is set to 0. For other symbols represented with *n* octets, then the *n* most significant bits of the first octet representing the symbol are set to 1. For remaining octets, the most significant bit is set to 1 and the second most significant bit is set to 0. The table below summarizes relationships between UTF8 and UCS4:

Table C.4 Relationships between UTF8 and UCS4

UTF8 octet sequence (binary)	UCS4 range (hexadecimal)	Description
`0xxxxxxx`	`0000 0000 to 0000 007F`	Used to encode US-ASCII symbols over 1 octet
`110xxxxx 10xxxxxx`	`0000 0080 to 0000 07FF`	Used to encode symbols over 2 octets
`1110xxxx 10xxxxxx 10xxxxxx`	`0000 0800 to 0000 FFFF`	Used to encode symbols over 3 octets
`11110xxx 10xxxxxx 10xxxxxx 10xxxxxx`	`0001 0000 to 001F FFFF`	Used to encode symbols over 4 octets
`111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx`	`0020 0000 to 03FF FFFF`	Used to encode symbols over 5 octets
`1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx`	`0400 0000 to 7FFF FFFF`	Used to encode symbols over 6 octets

Appendix D EMS–iMelody Grammar

The iMelody format is used in EMS and MMS messaging systems. The BNF grammar of the iMelody format (version 1.2) is given below:

```
<imelody-objesct> =       "BEGIN:IMELODY" <cr><line-feed>
                          "VERSION:" <version><cr><line-feed>
                          "FORMAT:" <format><cr><line-feed>
                          [ "NAME:" <characters-not-lf><cr><line-feed>]
                          [ "COMPOSER:" <characters-not-lf><cr><line-feed>]
                          [ "BEAT:" <beat><cr><line-feed>]
                          [ "STYLE:" <style><cr><line-feed>]
                          [ "VOLUME:" <volume><cr><line-feed>]
                          "MELODY:" <melody><cr><line-feed>
                          "END:IMELODY" <cr><line-feed>
<version> =               "1.2"
<format> =                "CLASS1.0" | "CLASS2.0"
<beat> =                  "25 | "26" | "27" |...| "899" | "900"
<style> =                 "S0" | "S1" | "S2"
<volume-modifier> =       "V+" |"V- "
<volume> =                "V0" | "V1" | ... | "V15" | <volume-modifier>
<basic-note> =            "c" | "d" | "e" | "f" | "g" | "a" | "b"
<flat-note> =             "&d" | "&e" | "&g" | "&a" | "&b"
<sharp-note> =            "#c" | "#d" | "#f" | "#g" | "#a"
< note> =                 <basic-note> | <flat-note> | <sharp-note>
<octave-prefix> =         "*0" |"*1" |...| "*8"
<duration> =              "0" | "1" | "2" | "3" | "4" | "5"
<duration-specifier> =    "." | ":" | ";"
<rest>=                   "r"
<led>=                    "ledoff" | "ledon"
<vibe>=                   "vibeon" | "vibeoff"
<backlight> =             "backon" | "backoff"
<full-note> =             [ <octave-prefix>] <note><duration> [ <duration-specifier>]
<silence> =               <rest> <duration> [ <duration-specifier]
<repeat> =                "("{ <silence> | <full-note> | <led> | <vib> | <volume> |
                          <backlight> } + "@"<repeat-count> [ <volume-modifier>] ")"
<repeat-count> =          "0" | "1" | "2" |...
<melody> =                { <silence> | <full-note> | <led> | <vib> | repeat> |<volume> |
                          <backlight> } +
<characters-no-lf> =      Any character in the US-ASCII character-set except <line-feed>.
```

Appendix E MMS–Content Types of Media Objects

Table E.1 provides the list of content-types commonly used for characterizing media objects contained in multimedia messages.

Table E.1 Content types of media objects

Format/codec	Content-type	
3GP	video/3gpp	
AMR	audio/amr	audio/x-amr
BMP	image/x-bmp	image/bmp
GIF	image/gif	
JPEG	image/jpeg	image/jpg
MIDI	audio/midi	audio/x-midi
PNG	image/png	
SMIL	application/smil	
SP-MIDI	audio/sp-midi	
SVG	image/svg+xml	
US-ASCII	text/plain; charset=us-ascii	
UTF-8 [RFC-2279]	text/plain; charset=utf-8	
UTF-16	text/plain; charset=utf-16	
vcalendar	text/x-vCalendar	
vcard	text/x-vCard	
WBMP	image/vnd.wap.wbmp	

Appendix F MM1 Interface–Response Status Codes (X-Mms-Response-Status)

The confirmation PDUs `M-send.conf`, `M-forward.conf`, `M-Mbox-delete.conf`, and `M-Mbox-view.conf` include a status code indicating the status of the corresponding request PDU (respectively, `M-send.req`, `M-forward.req`, `M-Mbox-delete.req`, and `M-Mbox-view.req`). The status can indicate that the request has been accepted by the MMSC or has been rejected because of a permanent or transient error. Status codes, as listed in Table F.1, can be assigned to the `X-Mms-Response-Status` parameter of the confirmation PDU.

For each PDU, only a set of error codes are applicable. The applicability of an error code to a particular PDU is shown in the three last columns of Table F.1.

Table F.1 Status codes/Response status (`X-Mms-Response-Status`)

	Name	B.E.	Description	`M-send.conf`	`M-forward.conf`	`M-Mbox-delete.conf` `M-Mbox-view.conf`
	`Ok`	0x80	The request is accepted by the MMSC.	✓	✓	✓
Transient errors	`Error-transient-failure`	0xC0	The request is valid but the MMSC is unable to process it, due to some temporary conditions.	✓	✓	✓
	`Error-transient-sending-address-unresolved`	0xC1	Due to some temporary conditions, the MMSC is unable to resolve an address specified in the request.			
	`Error-transient-message-not-found`	0xC2	Due to some temporary conditions, the MMSC is unable to retrieve the message.		✓	✓
	`Error-transient-network-problem`	0xC3	Due to some temporary conditions, the MMSC is unable to process the request.	✓	✓	✓
	`Error-transient-partial-success`	0xC4	The MMSC was not able to successfully complete the requested action for all the indicated multimedia messages.			✓
Permanent errors	`Error-permanent-failure`	0xE0	An unspecified permanent error occurred during the processing of the request by the MMSC.	✓	✓	✓
	`Error-permanent-service-denied`	0xE1	The request is rejected because of service authentication and authorization failure(s).	✓	✓	✓
	`Error-permanent-message-format-corrupt`	0xE2	The request is badly formatted.	✓	✓	✓
	`Error-permanent-sending-address-unresolved`	0xE3	The MMSC is unable to resolve one of the addresses specified in the request.	✓	✓	
	`Error-permanent-message-not-found`	0xE4	The MMSC is unable to retrieve the message.		✓	

(*Continued*)

Table F.1 (*Continued*)

Name	B.E.	Description	M-send.conf	M-forward.conf	M-Mbox-delete.conf M-Mbox-view.conf
Error-permanent-content-not-accepted	0xE5	The MMSC cannot process the request because of message size, media types, or copyright issues.	✓		
Error-permanent-reply-charging-limitations-not-met	0xE6	The request does not meet the reply charging limitations.	✓		
Error-permanent-reply-charging-request-not-accepted	0xE7	The MMSC supports reply charging but the request is rejected because of incompatibility with service or user profile configurations.	✓		
Error-permanent-reply-charging-forwarding-denied	0xE8	The forwarding request is for a message containing reply charging requirements.		✓	
Error-permanent-reply-charging-not-supported	0xE9	The MMSC does not support reply charging.	✓		
Error-permanent-address-hiding-not-supported	0xEA	The MMSC does not support address hiding.	✓		
Error-unsupported-message	0x88	Only used for version management. Used in a response to an unknown PDU or PDU with different major version number.	✓		✓

The set of error codes introduced in MMS 1.0 [WAP-209] is considered as obsolete for MMS implementations from MMS 1.1 [OMA-MMS-Enc] (version 1.1). Obsolete error codes are supported by MMS entities for ensuring backward compatibility. Obsolete error codes are listed in Table F.2.

Table F.2 Obsolete error codes/response status

	Name	B.E.
Obsolete errors	`Error-unspecified`	0x81
	`Error-service-denied`	0x82
	`Error-message-format-corrupt`	0x83
	`Error-send-address-unresolved`	0x84
	`Error-message-not-found`	0x85
	`Error-network-problem`	0x86
	`Error-content-not-accepted`	0x87

Error codes for which the binary encoding representation is in the range 0xE0. . .0xFF are all considered as permanent errors. Error codes for which the binary encoding representation is in the range 0xC0. . .0xDF are all considered as transient errors.

Appendix G MM1 Interface–Retrieve Status Codes (X-Mms-Retrieve-Status)

The confirmation PDU `M-retieve.conf` can include a status code indicating the status of the corresponding request PDU (`WSP/HTTP GET.req`). The status can indicate that the request has been accepted by the MMSC or has been rejected because of a permanent or transient error. Status codes, as listed in Table G.1, can be assigned to the `X-Mms-Retrieve-Status` parameter of the confirmation PDU.

Table G.1 Status codes/retrieve status (`X-Mms-Retrieve-Status`)

	Name	B.E.	Description
	`Ok`	0x80	The request is accepted by the MMSC.
Transient errors	`Error-transient-failure`	0xC0	The request is valid but the MMSC is unable to process it, due to some temporary conditions.
	`Error-transient-message-not-found`	0xC1	Due to some temporary conditions, the MMSC is unable to retrieve the message.
	`Error-transient-network-problem`	0xC2	The MMSC is unable to process the request due to capacity overload.
Permanent errors	`Error-permanent-failure`	0xE0	An unspecified permanent error occurred during the processing of the request by the MMSC.
	`Error-permanent-service-denied`	0xE1	The request is rejected because of service authentication and authorization failure(s).
	`Error-permanent-message-not-found`	0xE2	The MMSC is unable to retrieve the message.
	`Error-permanent-content-not-unsupported`	0xE3	The message contains contents that the recipient MMS client cannot handle and the MMSC is not able to perform the appropriate content adaptation.

Error codes for which the binary encoding representation is in the range 0xE0...0xFF are all considered as permanent errors. Error codes for which the binary encoding representation is in the range 0xC0...0xDF are all considered as transient errors.

Appendix H MM1 Interface–MMBox Store Status Codes (X-Mms-Store-Status)

The confirmation PDUs `M-send.conf`, `M-forward.conf`, `M-Mbox-store.conf`, and `M-Mbox-upload.conf` include a status code indicating the status of the corresponding request PDU. The status can indicate that the request has been accepted by the MMSC or has been rejected because of a permanent or transient error. Status codes, as listed in Table H.1, can be assigned to the `X-Mms-Store-Status` parameter of the confirmation PDU.

Table H.1 Status codes/store status (`X-Mms-Store-Status`)

	Name	B.E.	Description	M – send.conf M – forward.conf	M – Mbox – store.conf M – Mbox – upload.conf
	`Ok`	0x80	The request is accepted by the MMSC.	✓	✓
Transient errors	`Error-transient-failure`	0xC0	The request is valid but the MMSC is unable to process it, due to some temporary conditions.	✓	✓
Transient errors	`Error-transient-network-failure`	0xC1	The MMSC was not able to handle the corresponding request due to capacity overload.	✓	✓
Permanent errors	`Error-permanent-failure`	0xE0	An unspecified permanent error occurred during the processing of the request by the MMSC.	✓	✓
Permanent errors	`Error-permanent-service-denied`	0xE1	The request is rejected because of service authentication and authorization failure(s).	✓	✓
Permanent errors	`Error-permanent-message-format-corrupt`	0xE2	The request is badly formatted.		✓
Permanent errors	`Error-permanent-message-not-found`	0xE3	The MMSC is unable to retrieve the message.		✓
Permanent errors	`Error-permanent-mmbox-full`	0xE4	The user's MMBox is full.	✓	✓

For each PDU, all error codes are sometimes not applicable. The applicability of an error code to a particular PDU is shown in the two last columns of Table H.1.

Error codes for which the binary encoding representation is in the range 0xE5...0xFF are all considered as permanent errors. Error codes for which the binary encoding representation is in the range 0xC2...0xDF are all considered as transient errors.

Appendix I MM4 Interface–Request Status Codes (X-Mms-Request-Status-Code)

The responses PDUs `MM4_forward.RES`, `MM4_delivery_report.RES`, and `MM4_read_reply_report.RES` (MM4 interface) include a status code indicating the status of the corresponding request PDU. The status can indicate that the request has been accepted by the MMSC or has been rejected due to some errors. Status codes, as listed in Table I.1, can be assigned to the `X-Mms-Request-Status-Code` parameter of the response PDU.

Table I.1 Request status codes (`X-Mms-Request-Status-Code`)

Name	Description
`Ok`	The corresponding request was accepted without errors.
`Error-unspecified`	An unspecified error occurred during the processing or reception of the corresponding request.
`Error-service-denied`	The corresponding request was rejected due to failure of authentication or authorization of the originating MMSC.
`Error-message-format-corrupt`	An inconsistency with the message format was detected when the corresponding request was parsed.
`Error-sending-address-unresolved`	There were no MMS address (`From`, `To`, `Cc`) in its proper format or none of the addresses belong to the recipient MMSC.
`Error-message-not-found`	This status code is obsolete.
`Error-network-problem`	The recipient MMSC was not able to accept the corresponding request due to capacity overload.
`Error-content-not-accepted`	The message content was not accepted due to message size, media type or copyright issues.
`Error-unsupported-message`	The recipient MMSC does not support the corresponding request PDU.

Appendix J MM7 Interface–Status Code and Status Text

All responses over the MM7 interface include a status code (element `StatusCode`) of the corresponding request PDU and an optional status text (element `StatusText`). The status code can indicate that the request has been accepted by the MMSC or has been rejected due to some errors. Errors have been organized into four classes:

- Success or partial success
- Client errors
- Server errors
- Service errors

Status codes, as listed in Table J.1, can be assigned to the `StatusCode` element of response PDUs.

Table J.1 `StatusCode`

Error Class	Status Code	Status Text	Description
Success or success partial	1000	Success	This code indicates that the request was executed completely.
	1100	Partial success	This code indicates that the request was executed partially but some parts of the request could not be completed. Lower order digits and the optional `Details` parameter may indicate what parts of the request were not completed.
Client errors	2000	Client error	Client made an invalid request.
	2001	Operation restricted	The request was refused due to lack of permission to execute the command.
	2002	Address error	The address supplied in the request was not in a recognized format or the MMSC ascertained that the address was not valid for the network because it was determined not to be serviced by this MMSC. When used as part of a response, and multiple recipients were specified in the corresponding submission, this status code indicates that at least one address is incorrect.
	2003	Address not found	The address supplied in the request could not be located by the MMSC. This code is returned when an operation is requested on a previously submitted message and the MMSC cannot find the corresponding message.
	2004	Multimedia content refused	The server could not parse the MIME content that was attached to the SOAP message and indicated by the `Content` element or the content size or media type was unacceptable.
	2005	Message ID not found	This code is returned when an operation is requested on a previously submitted message and the MMSC cannot find the message for the `MessageID` specified or when the VAS application receives a report concerning a previously submitted message and the `MessageID` is not recognized.
	2006	LinkedID not found	This code is returned when a `LinkedID` parameter was supplied and the MMSC could not find the related message.

(Continued)

Table J.1 (*Continued*)

Error Class	Status Code	Status Text	Description
	2007	Message format corrupt	A parameter value format is inappropriate or incorrect.
Server errors	3000	Server Error	The server failed to fulfil an apparently valid request.
	3001	Not Possible	This code is normally used as a result of a cancel or status query on a message that is no longer available for such operations. The MMSC has recognized the message in question, but it cannot fulfil the request because the message is already delivered or status is no longer available.
	3002	Message rejected	Server could not complete the requested service.
	3003	Multiple addresses not supported	The MMSC does not support this operation on multiple recipients. The operation may be resubmitted as multiple single recipient operations.
Service errors	4000	General service error	The requested service cannot be fulfilled.
	4001	Improper identification	Identification header of the request does not uniquely identify the client (either the VAS application or MMSC).
	4002	Unsupported version	The version indicated by the `MM7Version` parameter is not supported.
	4003	Unsupported operation	The server does not support the request indicated by the `MessageType` parameter in the header of the message.
	4004	Validation error	The SOAP and XML structures could not be parsed, mandatory fields are missing, or the message format is not compatible to the format specified. The `Details` parameter may specify the parsing error that caused this status.
	4005	Service error	The operation caused a server (either MMSC or VAS application) failure and should not be resent.
	4006	Service unavailable	This indication may be sent by the server when service is temporarily unavailable, e.g., when server is busy.
	4007	Service denied	The client does not have permission or funds to perform the requested operation.

Acronyms and Abbreviations

3G	Third Generation
3GPP	Third Generation Partnership Project
3GPP2	Third Generation Partnership Project 2
AMR	Adaptive Multi-Rate
AMR-NB	AMR NarrowBand
AMR-WB	AMR WideBand
A.N.	Assigned Number
APN	Access Point Name
ARCH	ARCHitecture, OMA group
ARIB	Association of Radio Industries and Businesses
ARPANET	ARPA wide are NETworking
ASCII	American Standard Code for Information Interchange
AT	ATtention (AT commands)
ATM	Asynchronous Transfer Mode
BARG	Billing and Accounting Roaming Group, GSMA group
B.E.	Binary Encoding
BIFS	Binary Format for Scenes
BNF	Backus-Naur Form
bpm	beats per minute
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CC/PP	Composite Capability/Preference Profiles
CDMA	Code Division Multiple Access
CDR	Charging Data Record
CEK	Content Encryption Key
CEPT	Conférence Européenne des Postes et Télécommunications
CIF	Common Intermediate Format
CLI	Command Line Interface

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2

CLP	Command Line Protocol
CN	Core Network
CO	Cache Operation
CPI	Capability and Preference Information
CR	Change Request
CSCF	Call Session Control Function
CSD	Circuit Switched Data
CSP	Client Server Protocol
CSPDN	Circuit Switched Public Data Network
CWTS	China Wireless Telecommunication Standard
DCF	DRM Content Format
DF	Dedicated File
DID	Document IDentifier
DIG	Developer Interest Group, OMA group
DM	Device Management, OMA group
DNS	Domain Name Server
DRM	Digital Rights Management
DS	Data Synchronization, OMA group
EAIF	External Application Interface
EDGE	Enhanced Data Rate for Global Evolution
EEPROM	Electrically Erasable Programmable Read Only Memory
EF	Elementary File
EFI	External Functionality Interface
EFR	Enhanced Full Rate
EICS	Enabler Implementation Conformance Statement
EIR	Equipment Identity Register
EMS	Enhanced Messaging Service
ENUM	E.164 Number Mapping
ERMES	European Radio Messaging System
ESME	External SME
ETP	Enabler Test Plan
ETR	Enabler Test Requirements
ETS	Enabler Test Specification
ETSI	European Telecommuncations Standard Institute
EWG	Environmental Working Group, GSMA group
FDD	Frequency Division Duplex
FF	Fraud Forum, GSMA group
F-MMS	Fixed Line MMS Forum
FQDN	Fully Qualified Domain Name
GCF	Global Certification Forum
GGSN	Gateway GPRS Support Node
GIF	Graphics Interchange Format
GM	General MIDI
GMT	Greenwich Mean Time
GPRS	General Packet Radio Service
GPS	Global Positioning System

GRX	GPRS Roaming Exchange
GSA	Global mobile Suppliers Association
GSM	Global System for Mobile communication
GSMA	GSM Association
HLR	Home Location Register
HSCSD	High Speed CSD
HSS	Home Subscriber Server
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HTTP over SSL
IAB	Internet Architecture Board
IANA	Internet Assigned Numbers Authority
ICS	Implementation Conformance Statement
IE	Information Element
IED	Information Element Data
IEDL	Information Element Data Length
IEI	Information Element Identifier
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMAP	Interactive Mail Access Protocol
IMEI	International Mobile Equipment Identity
IMPS	Immediate Messaging and Presence Services
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IMT-2000	International Mobile Telecommunications 2000
ISDN	Integrated Services Digital Network
IOP	Interoperability
IOT	Interoperability Testing
IP	Internet Protocol
IREG	Interworking Roaming Expert Group, GSMA group
ISDN	Integrated Services Digital Network
ISOC	Internet SOCiety
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LIF	Location Interoperability Forum
LOC	LOCation, OMA group
LSB	Least Significant Bit
LZSS	Lempel-Ziv-Storer-Szymanski
MAG	Mobile Application Group, OMA group
MAP	Mobile Application Part
MCC	Mobile Country Code
MCEF	Mobile-station-memory-Capacity-Exceeded-Flag
MCOM	Mobile COMmerce, OMA group

MDI	Message Distribution Indicator
ME	Mobile Equipment
MExE	Mobile Execution Environment
MF	Master File
MGCF	Media Gateway Control Function
MGIF	Mobile Games Interoperability Forum
MGW	Media Gateway
MRF	Media Resource Function
MIDI	Musical Instrument Digital Interface
M-IMAP	Mobile IMAP
MIME	Multipurpose Internet Mail Extensions
MIP	Maximum Instantaneous Polyphony
MMA	MIDI Manufacturers Association
MMAP	Mobile Messaging Access Protocol
MMBox	Multimedia Message Box
MMoIP	Multimedia over IP
MMS	Multimedia Messaging Service
MMSC	MMS Center
MMSE	MMS Environment
MMS-IOP	MMS Interoperability group
MMS UA	MMS User Agent
MNC	Mobile Network Code
MNP	Mobile Number Portability
MNRF	Mobile-station-Not-Reachable-Flag
MNRG	Mobile-station-Not-Reachable-for-GPRS
MNRR	Mobile-station-Not-Reachable-Reason
MPEG	Motion Picture Experts Group
MPG	Mobile Protocols Group, OMA group
MS	Mobile Station
MSB	Most Significant Bit
MSC	Mobile Switching Center
MSCF	Messaging Service Control Function
MSISDN	Mobile Station ISDN
MTA	Mail Transfer Agent
MTU	Maximum Transmission Unit
MUA	Mail User Agent
MWI	Message Waiting Indication
MWIF	Mobile Wireless Internet Forum
MWS	Mobile Web Services, OMA group
NSS	Network Subsystem
ODI	Object Distribution Indicator
OMA	Open Mobile Alliance
OPS	Operations and Processes, OMA group
OSA	Open Service Architecture
OSS	Operation Subsystem
OTA	Over The Air

PAP	Push Access Protocol
PCG	Project Coordination Group
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PDC	Personal Data Cellular
PDP	Packet Data Protocol
PDU	Protocol Data Unit
PI	Push Initiator
PIM	Personal Information Manager
PKI	Public Key Infrastructure
PLMN	Public Land Mobile Network
PNG	Portable Network Graphic
PoC	Push-to-talk over Cellular
POP3	Post Office Protocol-3
PPG	Push Proxy Gateway
PRIM	Presence and Instant Messaging Protocol
PSPDN	Packet Switched Public Data Network
PSS	Packet-switched Streaming Service
PSTN	Public Switched Telephone Network
PTT	Push-To-Talk
QCIF	Quarter CIF
QVGA	Quarter VGA
RAM	Random Access Memory
RDF	Resource Description Framework
REL	RELease planning, OMA group
REQ	REQuirement, OMA group
RFC	Request For Comments
RIM	Research In Motion
RNC	Radio Network Controller
ROM	Read Only Memory
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
SAR	Segmentation And Reassembly
SAT	SIM Application Toolkit
SC	SMS Center
SCD	Specification Change Document
SCR	Static Conformance Requirements
SDO	Standard Development Organization
SDP	Session Description Protocol
SEC	SECurity, OMA group
SerG	Service, GSMA group
SG	Security Group, GSMA group
SGSN	Serving GPRS Support Node
SGW	Signaling Gateway
SI	Service Indication
SIM	Subscriber Identity Module

SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIN	Specification Implementation Note
SIP	Session Initiation Protocol
SL	Service Loading
SMAP	Short Message Application Protocol
SMCNP	Server-Mobile Core Network Protocol
SME	Short Message Entity
SMF	Standard MIDI File
SMIL	Synchronized Multimedia Integration Language
SM-MO	Short Message-Mobile Originated
SM-MT	Short Message-Mobile Terminated
SMPP	Short Message Peer to Peer
SMS	Short Message Service
SMSC	SMS Center
SMS-GMSC	SMS gateway MSC
SMS-IWMSC	SMS InterWorking MSC
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SP-MIDI	Scalable Polyphony MIDI
SSL	Secure Socket Layer
SR	Status Report
SS7	Signaling System Number 7
SSL	Secure Socket Layer
SSP	Server-Server Protocol
STI	Standard Transcoding Interface
SVG	Scalable Vector Graphics
Sysex	System exclusive message
SWG	Sub-Working Group
TADIG	Transferred Account Data Interchange Group, GSMA group
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TE	Terminal Equipment
TLS	Transport Layer Security
TP-CD	TP-Command-Data
TP-CDL	TP-Command-Data-Length
TP-CT	TP-Command-Type
TP-DA	TP-Destination-Address
TP-DCS	TP-Data-Coding-Scheme
TP-DT	TP-Discharge-Time
TPDU	Transfer Protocol Data Unit
TP-FCS	TP-Failure-Cause
TP-MMS	TP-More-Message-to-Send
TP-MN	TP-Message-Number
TP-MR	TP-Message-Reference
TP-MTI	TP-Message-Type-Indicator

TP-OA	TP-Originator-Address
TP-PI	TP-Parameter-Indicator
TP-PID	TP-Protocol-Identifier
TP-RA	TP-Recipient-Address
TP-RD	TP-Reject-Duplicates
TP-RP	TP-Reply-Path
TP-SCTS	TP-Service-Center-Time-Stamp
TP-SRI	TP-Status-Report-Indicator
TP-SRQ	TP-Status-Report-Qualifier
TP-SRR	TP-Status-Report-Request
TP-ST	TP-Status
TP-UD	TP-User-Data
TP-UDHI	TP-User-Data-Header-Indicator
TP-UDL	TP-User-Data-Length
TP-VP	TP-Validity-Period
TP-VPF	TP-Validity-Period-Format
TR	Technical Report
TS	Technical Specification
TSG	Technical Specification Group
TTA	Telecommunications Technology Association
TTC	Telecommunications Technology Committee
TWG	Terminal Working Group, GSMA group
UAProf	User Agent Profile
UCI	Universal Computer Interface
UCS	Universal Character Set
UDH	User Data Header
UDHL	User Data Header Length
UDP	User Datagram Protocol
UE	User Equipment
UICC	UMTS IC Card
UMTS	Universal Mobile Telecommunications System
UPI	User Prompt Indicator
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USIM	UMTS Subscriber Identity Module
USAT	USIM Application Toolkit
UTC	Co-ordinated Universal Time
UTF	UCS Transformation Format
UTRAN	Universal Terrestrial Radio Access Network
UWCC	Universal Wireless Communications Consortium
VAS	Value-Added Service
VASP	VAS Provider
VGA	Video Graphics Array
VHE	Virtual Home Environment
VLR	Visitor Location Register
VoIP	Voice over IP

VPN	Virtual Private Network
W3C	World Wide Web Consortium
WAE	Wireless Application Environment
WAP	Wireless Application Protocol
WBMP	Wireless BitMaP
WBXML	WAP Binary XML
WCDMA	Wideband CDMA
WDP	Wireless Data Protocol
WG	Work Group
WIM	Wireless Identity Module
WML	Wireless Markup Language
WP-HTTP	Wireless Profiled HTTP
WP-TCP	Wireless Profiled TCP
WSP	Wireless Session Protocol
WTA	Wireless Telephony Application
WTLS	Wireless Transport Layer Security
WTP	Wireless Transaction Protocol
WV	Wireless Village
WVG	Wireless Vector Graphics
XHTML	eXtensible HTML
XHTML-MP	XHTML Mobile Profile
XML	eXtensible Markup Language

A list of abbreviations and corresponding definitions used in 3GPP specifications is provided in [3GPP-21.905].

Index

Mobile Messaging Technologies and Services Second Edition Gwenaël Le Bodic
 ISBN: 0-470-01143-2