LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY

ADDING VALUE TO MESSAGE MEDIATION IN MULTIOPERATOR MMS NETWORK ARCHITECTURE

The subject of the thesis was approved by the council of the Department of Information Technology on December 11, 2002.

Supervisor: Professor Jari Porras
Supervisor: M.Sc. Toni Suihko
Instructor: M.Sc. Pasi Sirén

Lappeenranta, 26th January 2003

Jani Nurminen
Teknologiapuistonkatu 2 D 53
53850 Lappeenranta

ABSTRACT

Author: Nurminen, Jani

Subject: **Adding Value to Message Mediation in Multioperator MMS Network Architecture**

Department: Information technology

Year: 2003

Place: Lappeenranta

Master's thesis. Lappeenranta University of Technology. 79 pages, 14 figures, 4 tables and 3 appendices.

Supervisor: Professor Jari Porras

Keywords: MMS, mediation, MMSNA, adding value, gateway

Mobile messaging using the Short Messaging Service (SMS) has become a phenomenal success. Currently a successor to SMS is emerging. This service is called Multimedia Messaging Service (MMS). MMS messages can contain various image, text and video formats. Although mediation products such as the Intellitel Messaging Gateway are needed to connect non-interoperable SMS Center protocols together, the SMS-centric mediation approach is not reasonable for an MMS-capable mediator. An MMS-capable mediator has to add value to the mediation process itself in order to thrive and become a successful product.

This thesis explores the problematics of MMS mediation with special emphasis in developing an MMS-capable mediator from an existing SMS mediator product. The reasoning for the need of value addition is explained. Ways to add value to MMS mediation are proposed. Some of these proposed value-adding schemes are used in the implementation part, which is a mediator component meant for connecting to the MM7/VAS interface of the Nokia MMS Center.

TIIVISTELMÄ

Tekijä: Nurminen, Jani

Nimi: **Adding Value to Message Mediation in Multioperator MMS Network Architecture**

Osasto: Tietotekniikan osasto

Vuosi: 2003

Paikka: Lappeenranta

Diplomityö. Lappeenrannan teknillinen korkeakoulu. 79 sivua, 14 kuvaa, 4 taulukkoa ja 3 liitettä.

Tarkastaja: Professori Jari Porras

Hakusanat: MMS, sanomanvälitys, MMSNA, arvonlisäys, yhdyskäytävä

Mobiiliviestintään käytetystä lyhytsanomapalvelusta (Short Messaging Service, SMS) on tullut ilmiömäinen menestys. SMS-palvelulle on tulossa seuraaja, jota kutsutaan monimediaviestipalveluksi (Multimedia Messaging Service, MMS). MMS-viestit voivat sisältää useita erilaisia kuva-, ääni- ja videoformaatteja. Vaikka yhteensopimattomia SMS-keskuksia yhdistämään tarvitaan Intellitel Messaging Gatewayn kaltaisia sanomanvälitystuotteita, ei tällainen SMS-keskeinen lähestyminen sanomanvälitykseen ole MMS-kykyisen sanomanvälittimen kannalta järkevä ratkaisu. MMS-kykyisen sanomanvälittimen täytyy lisätä arvoa itse sanomanvälitysprosessiin, jotta tuotteesta tulisi menestyksekäs.

Tässä työssä käsitellään MMS-sanomanvälityksen ongelmallisuuksia. Erityisesti painotetaan MMS-kykyisen sanomanvälittimen kehittämistä olemassaolevasta SMS-sanomanvälitintuotteesta, sekä selvitetään syitä arvonlisäyksen välttämättömyydelle. Työssä esitetään myös erilaisia arvonlisäystapoja MMS-sanomanvälitykseen. Eräitä esitetyistä arvonlisäystavoista käytetään käytännön osassa toteutetussa Nokian MMS-keskuksen MM7/VAS-rajapintaan kytkeytymään kykenevässä sanomanvälityskomponentissa.

PREFACE

This thesis was made in Lappeenranta for the Department of Information Technology of Lappeenranta University of Technology.

My appreciation and gratitude go to: my parents and other family for supporting me and being there; my teachers; Intellinauts past & present for making everyday work not suck; Jaakko Kiviluoto for online BS and LyX & LaTeX-consultation; M.Sc. Toni Suihko & M.Sc. Pasi Sirén for offering valuable guidance and insight despite being very busy with other work; M.Sc. Ilkka Toivanen for not being a Pointy-Haired Boss and also for making it possible to write this thesis even during the turmoil and chaos of a reorganizing company; and, last but not least, to my instructor Professor Jari Porras, for slowing his lightspeed daily rhythm down to instruct me with this thesis. Also, thanks to Annastiina for bringing happiness. I would like to dedicate this thesis to all the dogs I've known.

6. Riding the Ox Back Home

Mounting the bull, slowly I return homeward.
The voice of my flute intones through the evening.
Measuring with hand-beats the pulsating harmony, I direct the endless rhythm.
Whoever hears this melody will join me.[1]

[1] Kakuan: The Ten Oxherding Pictures

Contents

List of Figures

List of Tables

ABBREVIATIONS

3G	3rd Generation
3GPP	3rd Generation Partnership Program
AMR	Adaptive Multi-Rate
API	Application Programming Interface
CDR	Call Detail Record
CIMD	Computer Interface to Message Distribution
CUG	Closed User Group
CVOPS	C Virtual Operating System
DNS	Domain Name System
EAIF	External Application Interface
EMI	External Machine Interface
EMS	Enhanced Message Service
ENUM	Telephone Number Mapping
ETSI	European Telecommunications Standards Institute
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GW	Gateway
HLR	Home Location Register
HTTP	Hypertext Transfer Protocol
IF	Interface

IMG	Intellitel Messaging Gateway
ISDN	Integrated Services Digital Network
JFIF	JPEG File Interchange Format
JPEG	Joint Photographic Experts Group
LDAP	Lightweight Directory Access Protocol
MAP	Mobile Application Part
ME	Mobile Equipment
MGMT	Management
MIME	Multipurpose Internet Mail Extensions
MMS	Multimedia Messaging Service
MMSC	MMS Center
MMSE	MMS Environment
MMSNA	MMS Network Architecture
MPEG	Moving Pictures Experts Group
MSISDN	Mobile Station International ISDN Number
MTA	Mail Transport Agent
MUA	Mail User Agent
OIS	Open Interface Specification
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PLMN	Public Land Mobile Network
POTS	Plain Old Telephony System
PSTN	Public Switched Telecommunication Network
RFC	Request for Comments

RPC	Remote Procedure Call
SDR	Service Detail Record
SIM	Subscriber Identity Module
SMIL	Synchronized Multimedia Integration Language
SMPP	Short Message Peer-to-Peer
SMS	Short Message Service
SMSC	SMS Center
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SPP	Service Provider Portal
SSL	Secure Sockets Layer
STK	SIM Toolkit
TCP	Transmission Control Protocol
UCP	Universal Computer Protocol
UDH	User Data Header
UMS	Unified Messaging System
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
USA	United States of America
VAS	Value Added Service
WAP	Wireless Application Protocol
WBMP	Wireless Bitmap
WSP	Wireless Session Protocol
WWW	World Wide Web
XML	Extensible Markup Language

1 Introduction

Few if any people could have foreseen the success which became of a pager-like messaging addition built into GSM (Global System for Mobile Communications) phones. This technology, dubbed SMS for Short Message Service, was not planned from the start to become a gold mine for mobile telephone operators worldwide. Nevertheless, SMS messages enjoyed a huge demand, and the market blossomed.

Due to mostly business reasons, many non-interoperable SMSC (SMS Center) products were gradually introduced. The SMSCs could not communicate with one another, and thus could not send messages between two SMSCs unless the SMSCs were from the same manufacturer. There was, however, demand for interoperability. The interoperability issue could be solved by using a message mediator, i.e. a messaging gateway which understands several SMSC protocols. [Sir2001] Using such a mediator SMSCs from different manufacturers could interoperate.

As time went by, SMS matured and many services became available. However, SMS was not well suited for transmission of images or sound, or other non-textual media. Some handset manufacturers introduced their own methods for achieving messages with richer media. Nokia did this with their proprietary Smart Messaging concept, and some time later Ericsson followed with a more open EMS (Enhanced Message Service). These technologies were noncompatible, although both used the UDH (User Data Header) functionality of SMS. [Mob2000]

Meanwhile, development of a technology known as MMS (Multimedia Messaging Service) was started. The aim of MMS was to bring colour images, sounds and video clips to the handset, offering unparalleled richness of media and expressive power. As 3G (3rd Generation) mobile networks were becoming capable of delivering data faster to the users, it made perfect sense to provide such a new kind of messaging service.

Although the MMS specifications, written by large manufacturers and operators, emphasized commitment to openness and interoperability, it seemed that the lessons learnt from SMS were forgotten. It became clear that two MMSCs (MMS Centers) could not interoperate if they were from different manufacturers. Even the protocol interfaces aimed at external applications (e.g. a mail server and other such legacy servers) were incompatible. These problems could be fixed with an MMS-capable message mediator, i.e. implementing MMS protocol interfaces to an existing SMS message mediator.

But is doing such a straightforward port of an SMS message mediator to MMS capability a reasonable approach? Is it enough for a successful product? The argument is that it is not. MMS specifications have continuously progressed towards increasing levels of interoperability via standard protocols and technologies. The demand for a protocol interface converter of today might be obsolete next month. A better strategy is to build an MMS-capable message mediator capable of more than the basic protocol conversion. The mediation functionality should be extended to the domain of messages. Other functionality should also be implemented to make the MMS mediator truly useful and to stand out from the masses. This thesis examines and discusses such ways to add value to MMS mediation.

This thesis begins with a glimpse of the history of mobile messaging. The concept of SMS mediation and a concrete example of an actual SMS mediation product (Intellitel Messaging Gateway) is examined. A tour of MMS is taken while at the same time issues relevant to mediation are studied. Ways to add value to MMS message mediation are proposed. Finally, an implementation utilizing some aspects of the proposed value added functionality is explained and examined.

2 Evolution of Mobile Messaging

The evolution of mobile messaging has been driven towards more and more capable terminals, richer content in messages and new kinds of services. This chapter takes a quick look at the development of mobile messaging. The development path starts with pagers, followed by SMS. It wasn't until SMS came along that the mobile messaging really started to bloom. While waiting for MMS to appear, EMS was proposed as an enhancement to SMS. However, EMS did not manage to convince all mobile terminal manufacturers and thus became a feature found only in some mobile terminals. The emphasis and hopes in the messaging business is now shifting into MMS. It now seems like SMS will be here to stay and is likely to co-exist with MMS for a long time. However, as MMS enables new kind of messaging solutions, services and products, old technologies such as pagers will gradually disappear from the markets.

2.1 Pagers

A pager is a radio device capable of simple messaging functions. The pager device can be one-way, two-way or one-and-half way, which is functionally in between the first and second type. One-way pagers are only capable of receiving messages from the paging network. There is no capability of acknowledging the received messages, thus the network cannot be certain the pager has received the message. [Mot2000]

One-and-half way pagers can receive messages and are able to acknowledge the reception, but they cannot send any messages other than the acknowledgement back to the network. For these pagers, the network can know that the pager device has received the message. Two-way pagers can both receive and send messages. [Mot2000] These devices resemble most the current SMS capable mobile terminals.

Creating services for pagers is not so straightforward. For example, roaming in ordinary pager networks is difficult, as some pagers may use a certain hardcoded frequency which cannot be changed without reprogramming the pager, or replacing the pager altogether. Also, one-way pagers cannot inform the pager network of their location. If the network cannot know the location of the pager, it is not possible to do roaming. [Mot2000] Lack of support for proper roaming limits the services which can be created. However, it is possible to make a nationwide paging service. In the USA (United States of America) this has been accomplished with the use of satellites. [Gar1998]

Other than the aforementioned pager types with different network connection capabilities, there are different kinds of pagers with differing message type and length support, different form factors, different screen capabilities and so on. As the messaging capabilities of pagers are very varied, it limits the services that can be created. For example, a one-way pager capable of showing only numeric messages offers very little room for creating services. Also, the lack of interactivity in one-way pagers is a big hindrance. There is no well-defined minimal set of capabilities, and pagers may be limited in the protocols and network architecture they support, thus complicating the situation even further.

Although pagers have limited support for creation of different kind of services, they suit well their original purpose, which was to offer a notification of a caller by e.g. displaying a number in the pager, or sending a simple message without having to call the recipient. Pagers were successful in the USA and Asia. In Europe, however, pager devices turned out not to be too popular with the exception of United Kingdom. Pagers are slowly going away in favor of GSM phones and other devices capable of more advanced messaging techniques. [Gar1998]

2.2 SMS

SMS has been a part of the GSM infrastructure since phase 1 of the GSM standards, enacted in 1990. The SMS defines mechanisms for relaying short messages between two single points in a GSM network. There is also a broadcast[2] version of SMS, but lack of profitable usage scenarios have hindered its usage. SMS is similar to the paging service, with the exception that SMS always allows for bidirectional messaging. [Sir2001] Especially in Northern Europe mobile messaging became a phenomenal success with the adoption of mobile handsets with SMS capability.

The length of a short message was initially constrained to 160 characters, chosen from a set of 7-bit GSM alphabet. The length stemmed from limitations in existing telecommunications network protocol implementations. [Sir2001] By using the SMS concatenation feature, multiple SMS messages can be ”chained” together, effectively making it possible to send messages longer than the usual 160 characters. [Mob2000]

[2]The same SMS is sent to all terminals within some base station area.

Later versions of the GSM standards, phase 2 and phase 2+, have further enhanced the capabilities of SMS, although from a point of view of the user, little has changed. An example of these enhanced capabilities is the previously mentioned mechanism to concatenate multiple SMS messages together. Yet another example is the STK (SIM Application Toolkit), with which the application stored in the SIM (Subscriber Identity Module) card can use the ME (Mobile Equipment) via various mechanisms which have been defined in the STK specification. The mechanisms include e.g. commanding the ME proactively (displaying text, playing a tone, initiating a voice call etc.), downloading of the ME profile to the SIM, creation of menus, call control mechanisms, autonomous parsing of SMS messages and security mechanisms. [ETS2000]

SMS messages utilize the unused capacity of GSM signalling channels. Because of this, it is possible to send or receive SMS messages during data or voice transmission. [Sch2000] The SMS supports confirmation of delivery. This allows the SMSC (Short Messaging Service Center) to know when the message has been received at the mobile terminal. In case the mobile handset has been turned off, the short message will be delivered when the handset comes back online. These features alone make SMS superior to most pager solutions.

Nokia has further enhanced SMS with their Smart Messaging technology. Smart Messaging is based on the mobile handset interpreting certain messages in a specific way. This allows e.g. picture messages and ringtones to be delivered to the handset. Smart Messaging is Nokia-specific technology, but other manufacturers have developed similar technologies, such as EMS, of their own. These technologies use SMS messages with the UDH (User Data Header) facility of SMS. [Mob2000]

Although a capability such as changing the ringtone of the handset may seem pointless and vain, such personalization of the handset has become very popular. From the user's point of view, being able to download new ringtones adds value to the handset in two ways: the handset will have a ringtone which suits the user's preferences, taste in music or lifestyle. Also, the ringtone is likely to be different from other user's ringtones, allowing the handset to be distinguished from other handsets in crowded places, e.g. in public transportations and stores.

SMS also supports extra features such as compression of messages and binary messages. [Mob2000] Some of the most common and generally considered to be useful services (because of many users) are for example the SMS weather service and SMS ticket service for local mass transit systems.

2.3 EMS

The Enhanced Message Service (EMS) allows messages to contain simple media such as melodies, pictures, sounds, animations, formatted text and normal text. EMS uses the same SMS Centers and the signalling channel as SMS to provide the EMS service. [Mob2000] Technically, EMS uses the UDH facility of SMS by encoding e.g. text formatting instructions to the header part of the message, and then interpreting this header in a certain way in the mobile handset.

Compared to SMS, EMS enables the short messages to have more complex content than regular SMS. In other aspects it is quite similar, as the same infrastructure which is used with SMS can be used with EMS messages. [Mob2000] However, EMS requires the handset to support EMS messages. A wide support for EMS would logically have required all (or almost all) handset manufacturers to implement EMS. However, Nokia decided not to go with EMS at all [Mob2000], but instead chose to focus their attention on MMS. Nokia's reluctancy to support EMS was partly because Nokia had already created their proprietary Smart Messaging technology, which works similarily to EMS and in a sense competes with it. Lack of support from Nokia was a setback for EMS, which was seen as an intermediate phase technology, an ”appetizer” before MMS was to arrive in a usable form.

EMS allows text to be displayed with different typefaces, font sizes and alignments. EMS also supports monochromatic pictures, which are from 16x16 pixels to 96x64 pixels. Predefined sounds with names like ”Ding”, ”TaDa” and ”Notify” are specified. There is also support for user-defined sounds. Predefined animations describing various emotions are stored in the handset. User-defined animations can be sent from one handset to another. [Mob2000]

2.4 MMS

Mobile messaging technologies have continuously progressed towards messages with more expressive power. Whereas plain SMS offered only short textual messages and simple pictures, ringtones and sounds with manufacturer-specific extensions, MMS allows using the myriad of media formats already used in the Internet.

Although SMS was originally meant to be a pager-like facility in GSM, it turned out to be much more succesful than pager services. EMS was introduced due to purely business-based reasons in an attempt to follow the success of SMS. EMS tried to bring more to SMS but it wasn't really a worthwhile idea to have bold and italic text and bigger fonts, as text is just text, even if it looks a bit different. MMS does something new by combining images, sound and text in one messaging format.

It has been predicted that still images will become the biggest application of MMS messages due to built-in (and external) cameras in mobile handsets. [Mob2000] The camera allows the user to act as a content provider. This is an excellent premise for the operator: as the content is created by the users themselves, less investment for operator-mandated content creation is needed. Also, as the images have obvious value to their creator, it is likely that the user wants to share such images with his social group. Sending images back and forth creates message traffic, which creates revenue for the operator (assuming the operator does make a positive amount of money per message). The volume of MMS messages is expected to surpass the volume of the SMS messages. This kind of vision is why still images are thought (and hoped) to be the the biggest application for MMS messages.

MMS allows also streaming content. Streaming, if supported by the network, can be used for example to view news broadcasts or movie trailers. If the cameras in handsets will support taking pictures rapidly, or properly support digital video, then MMS streaming could be used to view a video clip of whoever had left such a video message to the answering machine. This capability, although certainly possible with MMS, but unlikely to appear among the first wave of services, more than surpasses the early 1990s attempts at making a consumer video phone which utilized POTS (Plain Old Telephony System, the ordinary telephone network).

Although in theory MMS messages could be very large in size, in practice this is not so. Also, the set of supported media formats is limited. Most of the limitations of MMS message content depends not directly on the handsets themselves but making sure that the messages interoperate between the handsets. This interoperability of messages means that a Nokia phone can send e.g. a digital image to an Ericsson phone without trouble. To ensure interoperability, major manufacturers[3] have created an MMS conformance document. [CMG2002] The document specifies interoperability issues such as the maximum message size that is to be supported by all terminals, the minimum

[3]CMG, Comverse, Ericsson, Logica, Motorola, Nokia, Siemens and Sony Ericsson.

set of supported media formats and the specification of the limited subset of SMIL (Synchronized Multimedia Integration Language) language used in MMS terminals. For example, the document specifies that handsets must support messages at least up to 30 kb in size. The MMS architecture and the overall system will be dealt with in more detail in chapter 4.

2.5 Future Directions

What lies in the future is largely unknown, but conclusions can be drawn by examining the evolution of mobile messaging both at the level of the content and the technology itself. Mobile terminals have become more and more lighter and are filled with more features. Battery life has increased over the years. The form factors of mobile terminals have developed towards bigger and more colourful screens. In some cases, the headset can be connected with a wireless link (e.g. Bluetooth) to the handset. New kinds of applications, such as games, have become possible due to the increased processor power and memory available in the handsets. The same mobile terminal which can show text and sound can also show streamed full-colour video.

Wireless network technology, which is what ultimately enables mobile messaging, has become faster and more capable. For example, the possibility of streaming content allows the handset to view content that would otherwise never fit the limited memory and storage space of the handset. Although the handsets develop all the time, so does the media. Streaming will be useful even with devices with larger memory and storage space.

Wireless connectivity will be especially fast in places where it makes sense from the business point of view to invest in such infrastructure, i.e. in metropolitan areas, malls, airports and other such places which are continuously visited by many people. On the other hand, in places with many people (and thus many handsets) mobile handsets might be able to build an ad-hoc network and therefore avoid using the elements of network infrastructure, like messaging centers, as such. In such a case, the available network bandwidth could be preserved for those applications which truly need it, and messaging to handsets within the same area could be done by using the ad-hoc network. Mobile messaging in ad-hoc network might also have some interesting peer-to-peer applications, such as a mobile handset serving some globally relevant data to other handsets (e.g. latest news).

The development has been going towards more complex media: from plain text to simple images and animations to bigger, more colourful pictures and actual video. Will surfing the Internet soon be possible from the mobile handset, just like it is done at a desktop machine today? Will surfing using the mobile handset work without any proxies or other machines which dumb down the content, and will there be compromises in the rendering of the web pages, e-mails and other content? When looking at the evolution of mobile messaging, such a scenario of an all-encompassing Internet access as it is today seems plausible. However, Internet technologies are not in a technological standstill, but also develop constantly, and mobile technologies will keep on lagging somewhat behind. Despite this, mobile Internet access is possible already today with a PDA (Personal Digital Assistant), a mobile phone and a dialup. Bringing web surfing and the Internet to the mobile handset is something of a Holy Grail of mobile messaging. Indeed, whoever can make it happen with total freedom of surfing and seamless access to many kinds of media, will surely find an enormous amount of customers willing to pay for the convenience of an omnipresent Internet access.

In conclusion, the future of mobile messaging is going towards more complex content and more capable handsets, whereas the messaging at network level is going towards standardized protocols. Handset access to the network is evolving towards an IP-based solution. Making message mediation products which enhance basic message mediation capability by handling the content of messages in various ways add value to the mediator product. This is because basic message mediation itself is becoming an off-the-shelf product and the evolution of messaging is going towards more complex messages. These facts make it necessary for the mediator to enhance the basic mediation functionality by handling also at least some parts of the content domain.

3 Message Mediation

What is mediation? A dictionary defines the word *mediation* as ”to effect by action as an intermediary”, ”to act as intermediary agent in bringing, effecting, or communicating” and ”to transmit as intermediate mechanism or agency” [Mer2002]. In a nutshell, message mediation is the scenario in which some element, called the ”mediator”, is located (in a logical sense) between two elements, receives messages from the other element, possibly does something to the message itself or extracts information from the message, and then passes the message to the other element. The receiving and sending element can be the same, in which case the mediator acts like a filter.

Message mediation is needed, for example, when two elements want to send messages but do not use the same messaging protocol, or when the elements do not use the same message content representation. The message mediator acts as a ”translator” between these two elements and makes it possible for the elements to communicate with one another. Figure 1 illustrates the basic principle of message mediation. In the picture, servers A, B and C use non-interoperable protocols. The servers are connected to a message mediator (solid lines). The mediator understands the protocols used with all servers. The mediator routes and converts messages from one protocol to another, creating in essence transparent logical connections between the servers (dashed lines) which actually go through the mediator. Although not shown in the picture, the mediator can do many other things beside moving messages from one place to another.

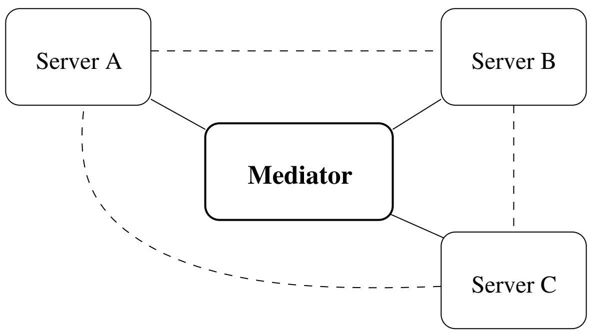


Figure 1: Principle of message mediation.

In this thesis, an individual protocol-to-protocol gateway is called a message mediator or gateway. A collection of message mediators is called a gateway. The gateway can also be referred by the name message mediator when it is seen as a black box which, on a high level, functions just like an individual message mediator: protocol messages come in and out comes messages using possibly a different protocol.

3.1 SMS Message Mediation

Message mediation is needed in SMS when the elements which are trying to communicate with one another are for example SMSCs or value added service providers. One aspect of SMS message mediation is to ensure that the network infrastructure can work together, i.e. that SMSCs from different manufacturers and the protocols used by the SMSCs interoperate. SMS message mediation can also be used to better integrate the SMSC to the operator's existing systems.

Additionally, through an SMS message mediation device interfaces for content providers can be provided. These interfaces are often quite simplified to ease the programming of new services. For example, the HTTP (Hypertext Transfer Protocol) protocol could be used as an interface for content providers [Sir2001], making an SMS to be sent with a HTTP POST operation. SMSC protocols are not the best protocols to offer directly to content providers due to the overall complexity and low-level technical details of the protocols. Offering a simplified interface for content providers is also a matter of convenience, which leads to faster development times for services.

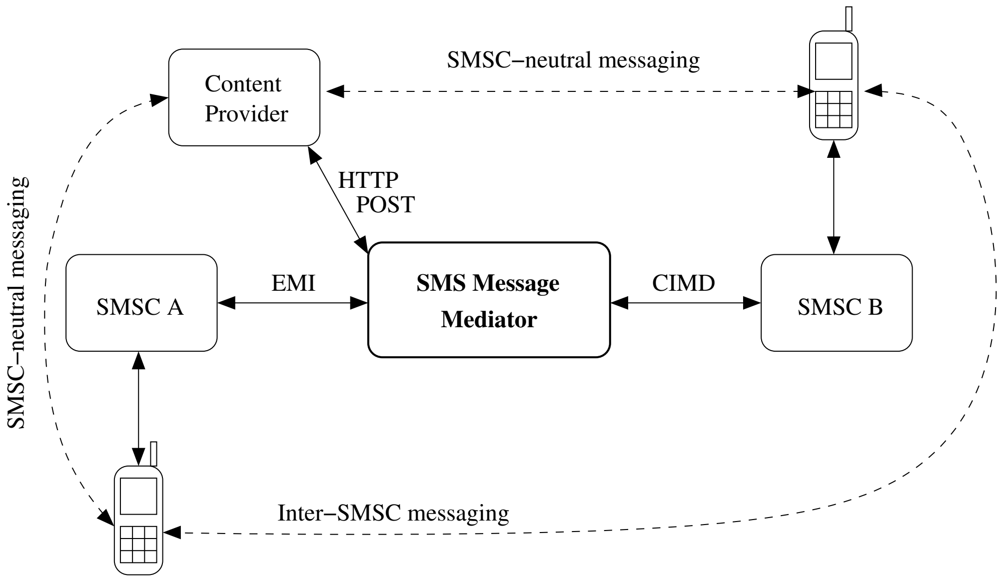


Figure 2: An illustrated case of SMS message mediation.

Figure 2 illustrates an SMS message mediation case. A solid line denotes an actual connection, whereas the dashed line denotes a logical connection. The mediator connects two non-interoperable SMSCs together, allowing inter-SMSC messaging. Also, SMSC-independent messaging is enabled to the content provider. The content provider also has simple access to the SMSCs using the HTTP POST operation. EMI (External

Machine Interface) and CIMD (Computer Interface for Message Distribution) are used in the illustration to denote non-interoperable SMSC protocols. These are actual SMSC protocols which will be discussed in the next section.

The justification for SMS message mediation stems from four facts: SMSC protocols between manufacturers are incompatible, SMSC performance is limited and the billing and security functionalities of the SMSCs nearly always need improvements. These facts are examined in detail next.

3.1.1 Incompatible SMSC Protocols

There are many SMSC manufacturers around the world. Many of these have a proprietary SMSC protocol which is primarily supported. Some of the SMSCs support additional protocols, but this functionality is usually a separately priced item, and operators are reluctant to pay for those. [Sir2001] This creates the situation of incompatible SMSC network protocols between major SMSC manufacturers. An SMS message mediator can act as an intermediate between two SMSCs, converting one protocol to another and thus making these SMSCs communicate with one another.

Although there are some similarities, there are also many differences among the SMSC protocols. For example, EMI [4] from CMG, CIMD from Nokia, SMPP (Short Message Peer-to-Peer) from Logica and OIS (Open Interface Specification) from Sema are all alike in the sense that they can use TCP (Transmission Control Protocol) and X.25 as the bearer protocol. As for the differences, CIMD and EMI have no time zone information in the timestamps, whereas SMPP and OIS do. EMI needs two connections to the SMSC, whereas CIMD uses one connection only. SMPP supports one or two connections, for one-way and two-way connections, respectively. Unlike other protocols, SMPP can function in transceiver mode, which enables it to use one connection for two-way traffic. Unlike CIMD and EMI, SMPP has no separate message for delivery notifications. [Sir2001]

[4] Also called UCP/EMI. UCP is Universal Computer Protocol, an ETSI protocol [Sir2001].

3.1.2 Performance-bound SMSC

There are limits as to how many messages a single SMS Center can handle. SMS Centers are expensive, and naturally the operators will try to get along with as little of these centers as possible. However, having too few SMSCs handling high-volume message traffic is disastrous. For example, during events such as New Year, people often send SMS greetings to family and friends. If the messages cannot get through in time, the users are reluctant to send such messages at all. Naturally, the operator cannot get money from unsent messages.

The reasons for performance limitations do not necessarily stem from the hardware used in the SMS Centers, but it is sometimes the SMSC protocols which are at fault. For example, the EMI protocol, when using synchronous stop-and-wait windowing, sends a message and expects to get a reply along the same connection. [Sir2001] If there is much traffic and the network latencies are high, each message reserves a connection for some time. Often the connections to the SMSC are limited, so that one service provider is not allowed to do many connections to the SMSC. This results in a loss of throughput of the messages, even though the SMSC could handle more messages. Adding multiple SMSCs will not help the situation much. A message mediator can help in this situation by making it possible to do load balancing between multiple SMSCs. Also, priorization of messages can be done, based on e.g. the sender or recipient number. For example, the service number of the operator can be given a higher priority than regular messages, so that the operator's service number replies very fast after a message has been sent to it.

3.1.3 Billing Issues

The data gathered by old SMSCs might not be enough to allow the billing of value added services. [Sir2001] An SMS message mediator can create billing tickets from the messages which pass through the mediator, thus circumventing the limitations of the SMSC. If the operator has multiple SMSCs, there might not be a single point and single format for the billing data. This makes billing difficult to do in practice. A better and more scalable solution than some kludgish script-hack is to use an SMS message mediator, which can act as a single point from which the billing data can be gathered. Also, an SMS message mediator can gather always more information and possibly integrate better with existing billing systems than the SMSC is capable of.

3.1.4 Security Issues

An SMSC might implement protocols which have minimal access control functionality. This is one of the reasons why operators prefer not to allow direct connections to SMSCs. An SMS message mediator can help in this problem by offering more elaborate authentication and access control mechanisms, possibly with encrypted communications and support for different encryption algorithms. The security argument applies especially to the interfaces offered to content providers and other non-operator third parties. [Sir2001]

3.2 Intellitel Messaging Gateway

The aforementioned problems with SMSCs can be remedied with a suitable SMS message mediator. Mediation in general is one aspect of the concept of network convergence. The vision of Intellitel is network convergence, and the company has focused on enabling convergence between different networks [Kem2001]. Thus, creating an SMS message mediator was a suitable business opportunity for Intellitel.

Intellitel Messaging Gateway (IMG) is Intellitel's solution to SMS message mediation. The gateway is built over the Intellitel ONE[5] platform, which offers, among other things, high availability mechanisms, system doubling and ticket generation. Management functionalities are provided via SNMP (Simple Network Management Protocol) and a WWW (World Wide Web) interface. Intellitel Messaging Gateway has installations around the world.

An overview of IMG can be seen in figure 3. The interfaces used with applications and content or service networks are located in the upper part of the picture. These interfaces are HTTP (Hypertext Transport Protocol) and SMTP (Simple Mail Transfer Protocol). At the bottom of the picture are interfaces towards wireless networks, with SMSC protocols and their corresponding vendors. The supported network interfaces are based on TCP/IP or X.25. The architecture of the gateway is modular and new network interfaces to support new protocols or network technologies (shown as "NEW" interfaces in the picture) can be added relatively easily. Tickets for billing can be created using the SDR (Service Detail Record) interface and transferred to the billing system using the built-in FTP (File Transfer Protocol) mechanism.

[5] Despite the initial appearance, ONE is not an acronym.

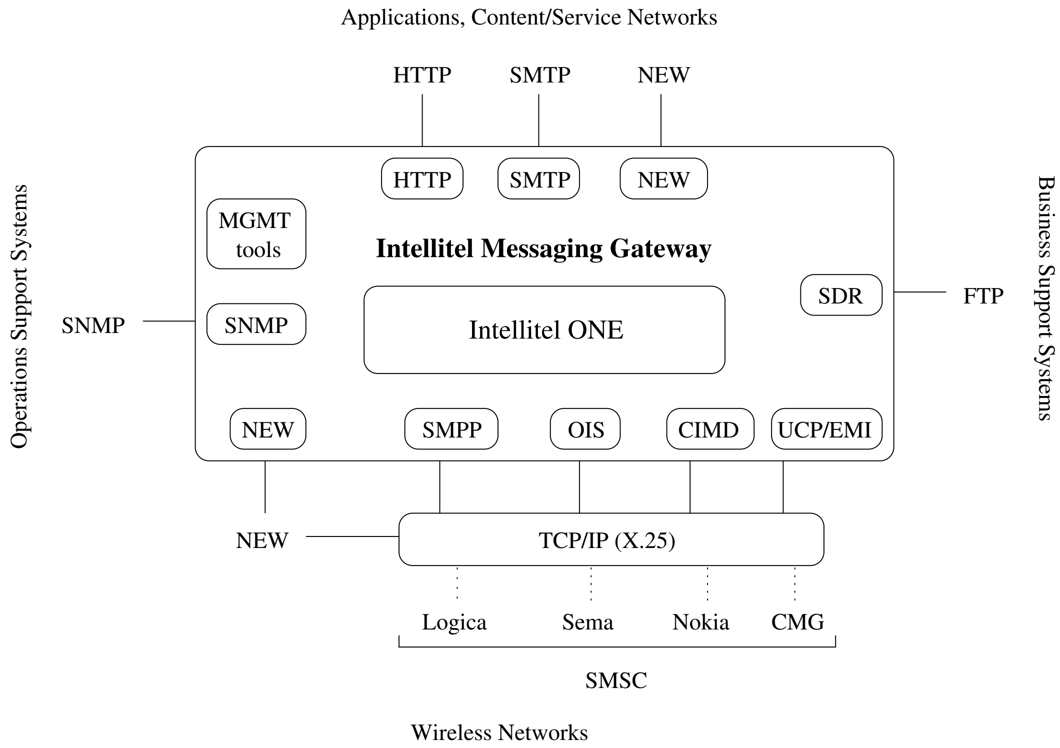


Figure 3: Overview of the Intellitel Messaging Gateway.

3.2.1 Existing Functionality

In addition to the content or service interfaces, network interfaces and the SNMP management interface, IMG has additional functionality [Int2002], some of which will be examined here. Some or all of the functionality described here is required in a carrier-grade mediator product. For this reason it is relevant to observe these functionalities.

For the programmer, the gateway offers functionality such as libraries for configuration, logging, alarms and statistics through the underlying Intellitel ONE platform. The configuration functionality is used to read certain user-configurable parameters. Logging is used to give information about the program execution in the form of log files. Alarms can be used to monitor certain events, such as rejected messages. Alarms can be raised in trouble situations, thereby notifying the administrators to be aware of the situation. Statistics is used to provide information about the amount of message traffic, message types and so on.

The billing facility is used to generate SDR tickets from traffic logs. A flexible parsing tool allows the customization of the information included in the tickets for better integration into existing billing systems. [Int2002] It is likely that some amount of customization is needed to the parsing tool, as there is no well-defined billing ticket standard and thus the existing billing interfaces can be very different from each other.

IMG supports hot stand-by clustering, wherein the system is doubled and the other side runs active while the other side is in stand-by mode. If an irrecoverable fault is detected, a system switchover is done. During a switchover, the active and stand-by sides change, the previously active side becoming a stand-by side, and the side which was in stand-by becomes the new active side. [Int2002] The administrators can then investigate the nature of the trouble which caused the other machine to fail, while the services provided by the gateway run uninterrupted. The switchover can also be done in a controlled fashion. This can be used when there are e.g. hardware upgrades.

The ingress and egress flow of SMS messages can be configured using the routing facility. [Int2002] The messages can be routed based on the protocol used, the first word of the message (keyword), or via a regular expression match of the sender and/or recipient. With the routing facility, it is trivial to implement simple gatewaying such as SMS messages to HTTP requests, e.g. from OIS to HTTP and vice versa.

Load balancing is used to balance MT (Mobile Terminated) messages internally in IMG when multiple logical SMSC connections have been configured via the routing facility. The purpose is to control the message flow and spread it evenly to reduce the possibility of bottleneck situations. The balancing algorithm can be round robin or something more sophisticated based on information from message queue lengths, transmission latency times and connection states. [Int2002b]

Additional functionality provided by the gateway which is only mentioned here includes persistent storage facility, distribution lists, support for Nokia Smart Messaging, SSL-protected connections, throughput limiter and support for configurable message segmentation. Any TCP-based connection can be configured to use SSL to protect the connection. [Int2002b]

3.2.2 Existing Protocol Interfaces

IMG has existing protocol interfaces for the aforementioned SMSC protocols to provide connectivity to the SMS networks. Also, there are service and content interfaces which provide HTTP and SMTP protocol functionality. These two protocols are collectively called content protocols. The architecture of the protocol interfaces is modular. New protocol interfaces can be written and basically ”plugged” into the system, as long as various conventions needed by the different system core processes are heeded. Table 1 shows the protocol interface names, their types and versions. The FTP interface used with SDR tickets for billing purposes is not a protocol interface as such, but a part of the billing related mechanisms of the management tools.

Table 1: Protocol interfaces of the Intellitel Messaging Gateway. [Int2002]

Name	Type	Versions
SMPP	SMSC	3.3 and 3.4
OIS	SMSC	6.0
CIMD	SMSC	CIMD2
UCP/EMI	SMSC	3.1.2, 3.5 and 4.0
HTTP	Content	1.0 and 1.1
SMTP	Content	Minimal RFC2821
SNMP	Management	v1, v2c and v3

As there is support for different versions for certain protocols, it is also possible in some cases to do a protocol converter from old to new and vice versa. This functionality can be useful in situations where the new protocol is not backwards compatible. It is also possible to do protocol conversion between content protocols, i.e. between SMTP and HTTP.

4 Multimedia Messaging Service

Multimedia Messaging Service (MMS) is a logical progression of SMS. Whereas SMS offers text-only short messages, MMS messages can contain many kinds of media, such as text, images and sound. This chapter examines the inner workings of MMS in detail.

There is hope within the telecommunications industry that MMS will become as big a success as SMS was. Indeed, manufacturers and operators have placed heavy expectactions to the success of MMS. Although MMS provides, from a technical point of view, a new kind of depth to mobile messaging, technical merits are not the main reason why operators are viewing MMS as their saviour. The operators are under pressure to succeed financially. This pressure originates partly from the sales of UMTS (Universal Mobile Telecommunications System) licenses, for which the operators have paid huge amounts of money. The pressure to succeed may make operators blind to the feedback received from users, especially if the feedback is negative. If something truly worthwhile is not provided to the user, there is a risk of turning MMS into another WAP (Wireless Application Protocol).

The problematic situation with WAP was nicely summed up in a report by Mobile Lifestreams Ltd. [Mob2000]:

> That is why nobody in the WAP value chain is making money - the WAP handset vendors are giving the networks volume products so that the networks can dump them on the market at heavily subsidized prices to try to build a critical mass of handset users. Meanwhile there are so many WAP Gateway vendors that the platforms are giving away free. The content providers give their content away free because there is no way for them to make money from it unless they do a deal with a network to list that service on their portal which the networks expect to be paid for. The venture capitalists and invest bankers that allow these uneconomics to continue by suspending all logic and rationale can fuel these follies for a temporary period, but it is fundamentally impossible for an industry to be a long term success if nobody is making any money from it.

Also in the eyes of the consumer, WAP was a failure. This was because despite all the hype that was generated, from the point of view of an average end-user WAP was a miserable experience of over-expensive services tied to a ”mobile portal” with a clumsy user interface. Selling pure technology to the end user simply did not work. Few users resulted in little revenue. Little by little the hype around WAP silently faded. However, WAP itself has not died, but instead has a big role in MMS. For example, the coding scheme used with WAP messages is used also with MMS messages. Also, the interface between the mobile handset and the MMSC is implemented with WAP.

The biggest difference between MMS messages and the messages used with existing mobile messaging schemes such as SMS are the numerous media types supported by MMS. The media types shall be examined next.

4.1 Media Types

Messages with pictures are expected to be among the biggest uses for MMS messages, and this makes the media types for still images especially important. The pictures need not come only in the MT direction, because some of the latest MMS phones, such as the Nokia 6650, have a built-in digital camera [Nok2002c], making it possible to send digital images also from MO (Mobile Originated) direction.

Other than still images, the supported media types include video, audio and plain text. For example, the MMS Conformance Document defines that MMS messages shall use still image coding formats such as GIF87a (Graphics Interchange Format), GIF89a, WBMP (Wireless Bitmap) and baseline JPEG (Joint Photographic Experts Group) with JFIF (JPEG File Interchange Format) as the exchange format. For text, the encoding formats are us-ascii, utf-8 and utf-16 with explicit Byte Order Mark. [CMG2002] For audio, the format to be supported is AMR (Adaptive Multi-Rate) [CMG2002] although the 3GPP (3rd Generation Partnership Project) document specifying MMS Media Formats and Codecs specifies also MPEG-4 (Moving Pictures Experts Group) to be used for non-speech audio.

It should be noted that the MMS conformance document refers to an outdated 4.5.0 version of the MMS functional description document when specifying the set of supported media formats. Although the MMS conformance document is not an official specification, it reflects the views and commitments of the major manufacturers. There are differences between the media formats in the latest MMS functional description document and the conformance document, although basically the media formats are not too different.

All of these media formats are standard formats which are already used in the Internet. By using such standard formats, interoperability with the content available in existing systems (e.g. Internet) is made much easier.

4.2 Structure of a Multimedia Message

An MMS message is an octet stream, composed of one or more parts. The encoding of the stream is specified in WAP-209 MMS Encapsulation [WAP2002] and WAP-230 WSP specifications [WAP2001]. On a high level, the message consists of MMS headers followed by one or more subparts. Each subpart contains their own MMS headers and a multimedia object. [WAP2002] Figure 4 illustrates the structure of the message. Overall, the scheme is similar to MIME (Multipurpose Internet Mail Extensions) multipart messages. Contrary to MIME messages, the body part of each element is a binary stream without a special encoding. As there is no special form of encoding for the content, there is no CONTENT-TRANSFER-ENCODING header applicable. There is, however, a CONTENT-TYPE header, which specifies the media type and, indirectly, also how the body part is to be handled.

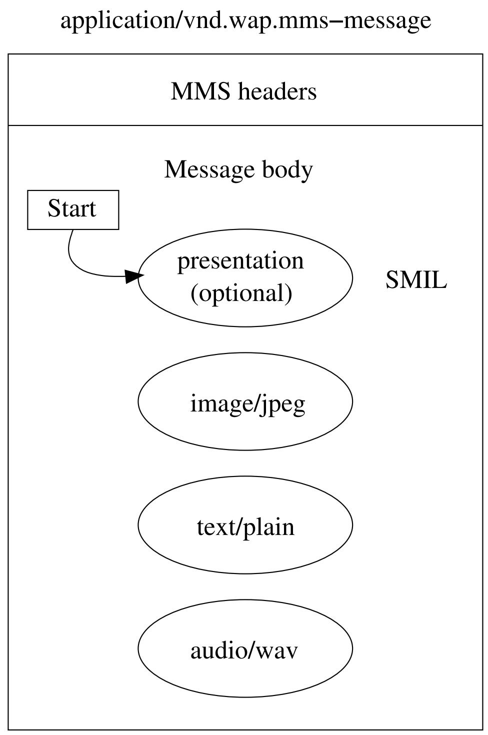


Figure 4: Model of MMS data encapsulation. [WAP2002] Augmented.

There can be a special part called the presentation part, which is implemented as a subset of the XML-based (Extensible Markup Language) SMIL 2.0 language. [ETS2002a, CMG2002] SMIL 2.0 is a W3C Recommendation. SMIL can be used to create interactive multimedia presentations by specifying the temporal behaviour of the overall presentation, associating hyperlinks with media objects and describing the layout of the presentation. [W3C2001] A more concrete example of an actual SMIL presentation part is given in Appendix 3.

The presentation part controls the presentation of each of the multimedia object parts. For example, if the MMS message consists of three images, then the presentation part could tell the MMS handset how long each part is shown in order to achieve a rendering resembling a slide show. [WAP2002, CMG2002] The presentation part is optional. If there is a presentation part, it has to be the first part of the message, or alternatively, the position of the presentation part must be set with the Start-parameter. If there is no presentation part, the MMS User Agent implementation in the MMS handset determines how the media object is handled. [WAP2001] For example, some handsets might show a picture for a short time only, whereas some might display it until a button has been pressed.

There are two kinds of messages, those consisting of only one part, and those consisting of many parts. These messages are called, respectively, single part and multipart messages. Single part messages contain one object only, for example, an image. Single part messages do not have a presentation part at all. Thus the presentation of the single part depends on how the MMS handset chooses to interpret it. The situation is similar for multipart messages. For multipart messages without a presentation part, each part is shown according to the logic in the MMS handset (e.g. each part sequentially).

When the multimedia message is a multipart message and there is a presentation part available to control the displaying of the message, the multimedia capabilities of the handset are utilized fully. Although full SMIL offers quite a lot of control over the media objects, at first the MMS handsets will support only a simplified subset of SMIL, which limits the complexity of the multimedia messages. It is likely that a more comprehensive SMIL support will be added later. [WAP2001, CMG2002]

4.3 Architecture

The Multimedia Messaging Service consists of MMS elements and Multimedia Messaging interfaces. A collection of MMS elements providing full Multimedia Messaging Service to the user is called an MMS Network Architecture (MMSNA). This includes interworking between MMSCs of different operators. When the MMS elements are under control of a single administration, such as an operator, and interworking is not done, the collection of such elements is called an MMS Environment (MMSE) [ETS2002]. The MMSNA consists of more than one interoperating MMSEs.

Figure 5 shows the MMS reference architecture. The MMS elements inside an MMSE communicate using interfaces which are named MM1 to MM8. The MMS functional description [ETS2002] describes these interfaces, although not all interfaces are defined on a detailed level. This fact was especially emphasized in the earlier revisions of the MMS functional description. One of the results of ambiquities and lack of strict technical description of the interfaces was the emergence of non-interoperable manufacturer specific external interfaces [Nok2002, Eri2001].

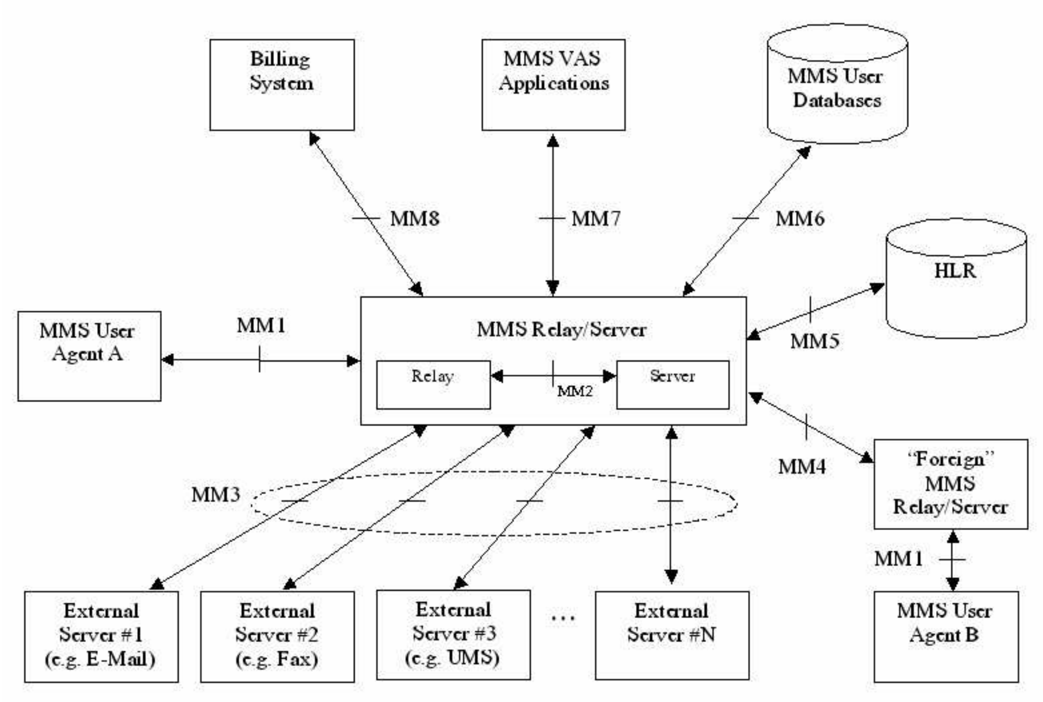


Figure 5: MMS reference architecture. [ETS2002]

All MM-interfaces, except MM2 (which is internal to the MMSC), connect from the MMS Relay/Server to some MMS element, and this makes MMS Relay/Server the most central element in the MMS architecture. Note that an MMS Center is not an actual element of the architecture as such, but instead a logical element, which consists of a combination of an MMS Relay and an MMS Server. If the elements are physically separated, the MMS Relay and MMS Server communicate using the MM2 interface [ETS2002]. For an MMS Center, the MM2 interface is internal. Depending on the architecture of the MMS Center, such interface may not actually exist at all.

4.3.1 MMS Relay/Server aka MMS Center

The MMS Relay and MMS Server elements can be physically distributed separate elements. However, the MMS functional description deals only with the MMS Relay/Server as an aggregate element. [ETS2002] It does not specify the separation of the MMS Relay/Server into MMS Relay and MMS Server elements. Thus, in practice these elements are always combined to form an MMS Relay/Server, which can, for all practical purposes, also be called an MMS Center. In this thesis, the preferred term is MMS Center, although MMS Relay/Server is used when talking in the context of the MMS functional description, as the specification uses the term MMS Relay/Server exclusively.

The MMS Relay/Server is responsible for storage and notification, reports, and general handling and successful delivery of multimedia messages. Address translation is also done by MMS Relay/Server. The MMS Relay/Server may also generate billing information in the form of CDR (Call Detail Record) tickets when receiving or submitting messages, but this functionality is not mandatory. [ETS2002] The MMS functional description also defines a non-exhaustive list of additional functionality which the MMS Relay/Server should support.

One of these functionalities is the optional support for a network-based persistent storage of messages. Such mechanism is called an ”MMBox” by the MMS functional description. The MMBox can manifest itself to the user as, for example, a web-based interface allowing the user to view and manipulate the stored MMS messages. [ETS2002] MMBox can be used in situations when the user's handset does not support MMS. In such a case, the MMS messages can be configured to be automatically stored into the MMBox. Using these mechanisms, the user can effectively receive and view MMS messages, even without MMS support in handset. The user does, however, require access to a web browser.

As the MMS Relay/Server is a central point in the MMS architecture, it has to implement all of the MM interfaces. However, the MMS functional description does not specify every MM interface in detail. The definitions of MM2, MM5, MM6 and MM8 interfaces are insufficient. The MM3 interface is described but lacks details which may hinder implementation in some cases. All this leaves room for manufacturer-specific implementations and interpretations. Therefore, despite the few interfaces which are specified in proper detail for technical implementations, a given MMSC of some manufacturer is not modular nor standardized enough to replace the MMSC implementation of another manufacturer. In other words, whoever manufactures the MMSC, defines and controls the structure and technical details of the entire MMSE up to a large degree.

Despite the MMSE partly being a manufacturer-specific black box, the relevant interfaces to allow interworking between MMSEs have been specified at a detailed enough level in the specification. The interworking between MMSEs is implemented using the MM4 interface, as shown in figure 6. According to the MMS functional description, the MM4 interface uses the SMTP protocol as the transport protocol. [ETS2002] This allows the creation of an MMSNA, as the MMSEs making up the MMSNA can talk to each other with a standardized transport protocol. Many manufacturers have probably already developed an SMTP protocol implementation in-house for use in other products.

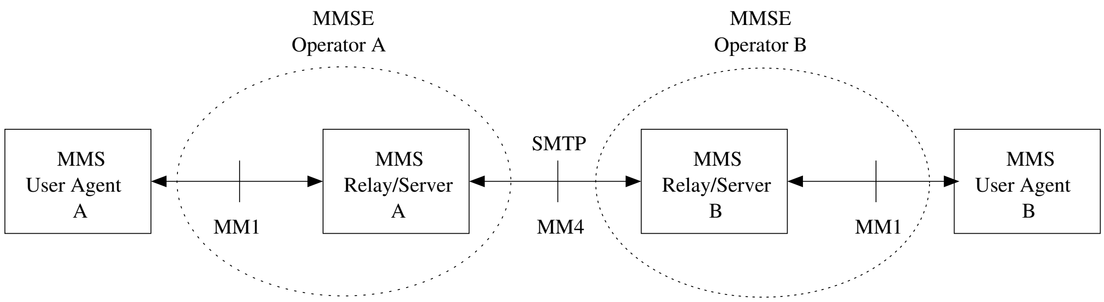


Figure 6: MMSE interworking. [ETS2002] Augmented.

Having a standardized protocol as the transport protocol is a good thing from the point of view of the manufacturers. First, the development time for a new product can be made shorter if there already exists a usable SMTP implementation. Second, interworking between different MMSCs is likely to be more easier to specify and implement, and thus well-functioning interworking is easier to achieve than with a protocol made completely from scratch. Also, an SMTP-based interworking can use the existing MTA infrastructure, although in practice a closed network is more likely. Had the interworking followed the same model as the SMSCs with their proprietary non-interoperable protocols, MMS messages could only be sent through a chain of MMSEs whose central element, the MMSC, has been manufactured by the same company. Since in practice all MMSCs cannot be made by the same manufacturer (currently no company has a monopoly on MMSCs), mutually non-interoperable MMSCs could only hinder the overall utilization and adoption of MMS.

4.3.2 MMS User Agent

An MMS User Agent is software residing in an MMS-capable handset. The MMS User Agent is capable of retrieving and handling MMS messages, and is also responsible for negotiating terminal capabilities. The MMSC needs to know the capabilities of the MMS User Agent (i.e. terminal capabilities) in order to do content transcoding, such as scaling too large images, reducing colours of an image or converting a stereophonic sound sample to a monophonic one. An MMS User Agent can also offer support for things like encryption or modification of received MMS messages, but such functionality is not required by the MMS functional description. The only mandatory functionalities are the retrieval of MMS messages and terminal capability negotiation. [ETS2002]

The MMS functional description also defines a minimal set of media formats an MMS User Agent must support [ETS2002, ETS2002a]. These formats include e.g. textual formats, such as short messages and plain text, and various sound and image formats. In addition to this, the MMS conformance document [CMG2002] defines the "lowest common denominator" for terminal capabilities. Despite the existence of such a document and the apparent strong commitment to ensuring interoperability of message content between terminals, there have been problems with the SMIL parts when sending messages between Nokia and Ericsson MMS phones [Nok2002a]. It is likely that in the early phases of MMS adoption there will be many such trouble spots, but as time goes on they will gradually be eradicated.

4.3.3 MMS Databases

An MMSE contains, from an architectural point of view, two database elements. These databases are the collection of MMS User Databases and the HLR (Home Location Register). The MM-interface between the MMSC and the HLR is MM5, and the interface between the MMSC and the MMS User Databases is MM6. The MMS functional description does not define the MM5 interface in any more detail than specifying standard MAP (Mobile Application Part) operations to be used when accessing the HLR. There is no discussion about using the MM5 interface in non-GSM networks. [ETS2002]

The MMS User Databases contain information about the users of the MMS Center. According to the MMS functional description, the MMS User Databases will hold user subscription information, access control information, terminal capabilities, rules how to handle incoming messages (e.g. reject messages from ”Bob”) and information about the server (such as the amount of available server storage space) [ETS2002]. Although the MMS User Databases can consist of more than one database, it is seen as one architectural element which is accessed via the MM6 interface.

The implementation details of the MMS User Databases and the associated MM6 interface are not defined in the MMS functional description. The only mandatory information needed with the MMS User Databases is the previously mentioned information. Since neither the MMS User Databases nor the MM6 interface are specified in detail [ETS2002], it can be expected that the MMS User Databases of different MMSC manufacturers are unable to interoperate. In practice, one of the possibilities for an MM6 realization might be LDAP (Lightweight Directory Access Protocol), which is an open protocol for accessing X.500 directory services. [Yeo1995]

4.3.4 MMS External Servers

MMS External Servers are machines which are connected to the MMSC using the MM3 interface. These may include ”legacy servers” such as e-mail servers, SMSCs, fax gateways and so on. The purpose of supporting external servers in the MMSC is to provide the services of the external servers in some form to the MMS User Agent, thus enabling convergence of legacy services with the MMSE. [ETS2002]

4.3.5 MMS VAS Applications

MMS VAS (Value Added Service) Applications provide value added services to the MMSE users. [ETS2002] These value added services can be, for example, weather and news services. An MMS VAS Application behaves much the same way as a fixed MMS

User Agent, although an MMS User Agent may be unable to provide all the features that an MMS VAS Application can [ETS2002].

MMS VAS Applications communicate with the MMSC using the MM7 interface [ETS2002]. The MMS VAS Applications can be a source or destination (or both) of MMS messages. When the VAS application is functioning as a source and destination simultaneously, it receives content, extracts information from or possibly modifies the message, and then sends it back to the MMSC. Using such a receive-modify-send filtering mechanism, it is possible to implement e.g. a content transcoding server as an VAS application and avoid building such functionality into the MMSC itself.

4.4 Interfaces

There are a total of 8 MM-interfaces, named MM1 to MM8. Table 2 shows an overview of the interfaces specified in the MMS functional description. Not all of the MM-interfaces are well-specified in the specification or interesting within the scope of this thesis. The MM3 and MM7 interfaces are examined in more detail than the others. The interfaces and how they settle into the overall architecture can also be seen in figure 5.

Table 2: MM-interfaces.

MM-interface	Specified	Protocol	Note
MM1	Yes	WAP	Can be external via a WAP GW
MM2	No	SMTP/HTTP	Internal to the MMSC
MM3	Yes	SMTP/HTTP	
MM4	Yes	SMTP	
MM5	No	MAP	
MM6	No	-	Depends on the manufacturer
MM7	Yes	SOAP 1.1	Was not specified until v5.3.0
MM8	No	-	Depends on the manufacturer

As the MM1 interface is implemented with WAP, there are two options for manufacturers: they can license or make their own WAP stack and couple it tightly with their MMSC, or they can use an external WAP gateway to implement the MM1 interface. The WAP gateway can be logically within the MMSC, but still architecturally (also possibly physically) an element external to the MMS Relay/Server part. At least Nokia uses such an external WAP gateway with their Artuse MMSC product. [Nok2002b] An external WAP gateway is an easier solution, as implementing a WAP stack from scratch is a non-trivial task. Luckily there are easier ways to interface with the MMSC for message mediation purposes, namely, the MM3 or MM7 interfaces.

The MM2 interface is internal to the MMSC and depends on the implementation of the MMSC. The MM2 interface is not defined in the MMS functional description. In practice, MM2 may not exist at all. The MM2 interface is uninteresting within the scope of this thesis, as there is most likely no possibility to connect to and utilize the MM2 interface for mediation purposes.

The MM4 interface is used for interworking between different MMSCs and thus connecting two MMSEs together. This enables messages to be sent from one MMSE to another. Early versions of the MMS functional descriptions did not specify MM4 in abundant detail [ETS2001b]. It may be for this reason why for example the Nokia Artuse MMSC release MC1 did not initially support the SMTP-based protocol at the MM4 interface, but instead had implemented a proprietary protocol based on HTTP [Nok2002b]. According to Nokia, specifications-compliant implementation for MM4 interface would be supported in later versions. This was in year 2001, and things have likely changed by now, although available information on the capabilities of the current Artuse MMSC is very scarce. In retrospect, when thinking of ways to interface with the MMSC, concentrating on other interfaces than the MM4 proved to be the right choice.

The MM5 and MM6 interfaces are used to access the databases of the MMS architecture. The MM5 interface is used by the MMSC to connect to the HLR. [ETS2002] Although outside the scope of the MMS functional description document, interfacing with the HLR can offer interesting new possibilities, as it gives access to e.g. location data. The MM6 interface connects the MMSC to the MMS User Databases. The implementation of the MM6 interface depends on the solution the MMSC manufacturer has chosen to use as the realization of the MMS User Databases. The MMS functional description does not specify the MM6 interface in detail.

The MM8 interface is used to access the MMS Billing System element. As with the MM6 interface, the MMS functional description has nothing to say about the implementation of the MM8 interface [ETS2002]. However, it is of interest to note that there *is* a separate interface specified as the MM8 interface, even if this is only so that the billing system can be explicitly mentioned as a logical element of the MMSE. It shows that there is demand for such an external billing solution among the companies that participate in the 3GPP group which creates the MMS functional descriptions.

The MM8 interface depends completely on the MMSC implementation. If an operator has multiple MMSCs which are from different manufacturers, how does the operator guarantee uniformity, with regard to e.g. management and billing? This is a case where message mediation can add value to the operator in the form of billing mediation, i.e. the main purpose being not the mediation of messages but the provision of uniform billing and management interfaces. To receive the messages from the MMSC, the message mediation system acting as a billing mediator can use the MM3 or the MM7 interfaces, if the specifications for MM8 are not available. The mediator can act as a message filter (or sink, if copies of the messages are given to the mediator) and examine the received messages to get relevant data for use with billing.

Of special interest within the scope of this thesis are the aforementioned MM3 and MM7 interfaces, which are used to interface with external application and value added services, respectively. These interfaces will be examined in more detail next.

4.4.1 MM3 - External Applications

The purpose of the MM3 interface is to allow existing servers and applications to interface with the MMSC more or less transparently and possibly without any modifications at all. The MMS functional specification proposes a fax gateway, a voice mailbox, an e-mail server and an SMSC as an example of the external applications which could possibly be connected to the MMSC via the MM3 interface. [ETS2002]

There are some ambiquities in the specification of the MM3 interface. For example, ways to convey message class, priority and subject are very dependent of the implementation of the external application protocol. Naturally, the specification cannot anticipate for all possible protocols that could be used with the MM3 interface, and this fact leads to ambiguity in the specification. Thus, the MMS functional specification gives only a description of the high-level functionality needed by an MM3 interface realization, and not a deeply technical and comprehensive definition.

Also, the scope of support for the various MM3 protocols is up to the MMSC implementation. If the MMSC does not support certain protocols with the MM3 interface, then such external applications depending on that protocol cannot connect to the MMSC via the MM3 interface. This problem can be circumvented by placing a mediator between the MMSC and the external servers, as illustrated in figure 7.

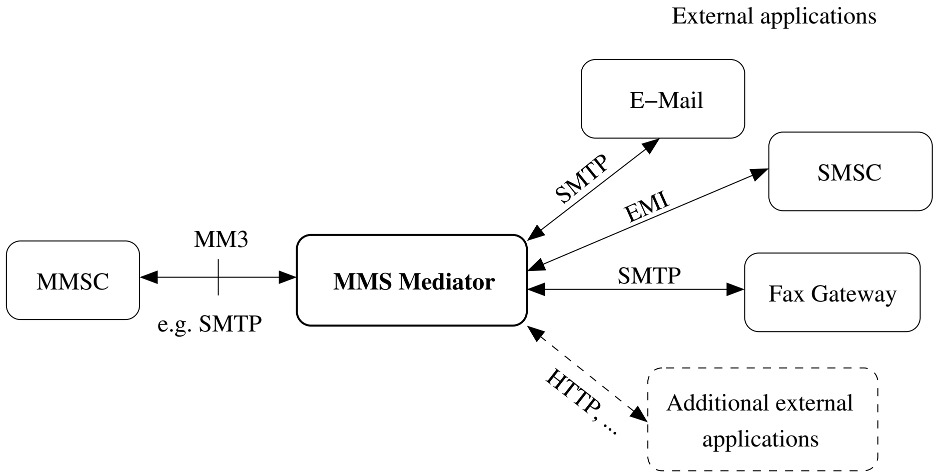


Figure 7: MMS mediator at the MM3 interface.

The mediator at the MM3 interface would have to implement some MM3 protocol(s) known to the MMSC (to connect to the MMSC) and a number of external application protocols (to connect to the external applications). Also, routines for conversions of the message content, and therefore parsing of the message itself, would be needed for some combination of protocols, e.g. from CIMD to an SMTP realization of MM3. However, the more external application protocols an MMSC supports, the less need there is for a message mediator at the MM3 interface.

Any external server that connects to the MMSC and can receive messages from and send messages to the MMSC can act very much like a VAS application. Therefore, if need be, the MM3 interface can be used as functionally equivalent substitute to an MM7 interface. If it is enough to have at least some kind of connection to the MMSC, and the MM7 interface is not available, the MM3 interface can be used to fulfill this task.

4.4.2 MM7 - Value Added Services

The MM7 interface was not well defined in the early versions of the MMS functional description. Versions before and including v5.1.0 did not specify anything concrete about the MM7 interface [ETS2001]. Version 5.2.0 introduced the primitives and message sequences, but did not specify exactly the protocol to use as the transport protocol of the MM7 primitives, but instead had an ambiguous statement ”this reference point shall be based on existing protocols e.g. SMTP or HTTP” [ETS2002b]. It was in version 5.3.0 of the MMS functional description that the MM7 interface was for the first time specified to be based on SOAP 1.1 using HTTP as the transport layer [ETS2002c]. SOAP is an XML-based RPC-implementation (Remote Procedure Call) and a W3C Note [W3C2000].

Due to ambiguities in the definition of the MM7 interface in the early versions of the MMS functional descriptions, the manufacturers of the MMSCs with the biggest market shares have made differing implementations of the MM7 interface. The specifications for some of these implementations are nevertheless available. For example, Nokia has a HTTP-based External Application Interface (EAIF) [Nok2002], whereas Ericsson uses an SMTP-based implementation [Eri2001]. These two interfaces are non-interoperable. Although previously the MMS functional description has been vague about which protocol (HTTP or SMTP), to prefer, there is likely also business tactics at work for not making the two MMSCs use the same protocol for MM7.

Due to non-interoperable implementations in the MM7 interface, there appears to be an immediate message mediation need to allow the VAS applications using the HTTP- or SMTP-based protocols to communicate with each other. However, although enabling interoperability can add value to the mediation, it is not enough from the point of view of the operator. If the MMSCs can work together using the MM4 interface, there is not much sense in using a mediator to connect the MM7 interfaces of two MMSCs together. However, it can be of great use to content providers to be able to communicate with both Nokia and Ericsson (and possibly more) MMSCs using their MM7 interfaces.

If the future versions of the MMSCs currently on market actually implement the MM7 interface using SOAP as specified, there will much less need for mediation products such as Intellitel Messaging Gateway to have their functionality migrated directly to the MMS world. This is because SOAP is in itself a well-specified technology with free and commercial implementations already available [Apa2002, Mic2002]. Thus, basic protocol conversions are not enough. Value adding functionality is required to make a succesful mediator product for the MMS world.

4.5 Addressing

MMS supports different kind of addresses within the multimedia messages. Since the addressing is especially relevant in the elements using the MM1 interface, which was implemented using WAP, the specification for the addressing scheme is done by WAP Forum and can be found in the WAP-209 MMS Encapsulation document [WAP2002].

The addressing scheme allows for ordinary RFC822 mail addresses [Cro1982], IPv4 and IPv6 addresses, phone numbers and a an extendable address type. [WAP2002] The RFC822 mail address is used as-is according to the RFC. The other addresses are separated with a /TYPE=<TYPE> postfix, where <TYPE> indicates the address type used. For IPv4 and IPv6 addresses the type is ”IPV4” and ”IPV6”, respectively. For phone numbers the non-mandatory [ETS2002] type is ”PLMN ” (Public Land Mobile Network). Extendable addresses can have a freeform address with anything[6] in the type field.

Table 3 shows some example MMS addresses. Also, concrete examples of the usage of the /TYPE parameter can be seen. The first two addresses are phone numbers. The third one is an RFC822 mail address, and the last two are IPv6 and IPv4 addresses, respectively.

Table 3: Example MMS addresses. [WAP2002]

To: 0401234567/TYPE=PLMN
To: +358501234567/TYPE=PLMN
To: Joe User <joe@user.org>
To: FEDC:BA98:7654:3210:FEDC:BA98:7654:3210/TYPE=IPv6
To: 195.153.199.30/TYPE=IPv4

Addressing also affects the MM4 interface. The addressing on the MM4 interface can be a mapping of the MMS Relay/Server domain name to an IP address using normal DNS operations. In the case the recipient number is in MSISDN format the mapping can be an DNS-ENUM lookup from the name server. The DNS-ENUM mechanism is not mandatory, and if it is not available, hardcoded look-up tables or other such mechanisms can be used to map the addresses. [ETS2002] In any case, the addresses should resolve to an IP address for the MMSC, so that the sender MMSC can relay the message to the correct recipient MMSC. In the case of inter-MMSC traffic, the sender and recipient addresses are reformatted to include the FQDNs (Fully Qualified Domain Name) of the appropriate server.

On the MM7 interface the addressing schemes to support are E.164 (MSISDN) addresses and RFC2822 [Res2001] e-mail addresses. Support for short codes is not mandatory, but desired. [ETS2002] Short codes enable the VAS-terminating MMS message to be sent to e.g. number 400, or other short code which is easy for the users to remember.

[6] Limited to alphanumeric characters.

4.6 MMS Centers

Like in SMS, MMS has dedicated centers which handle the message traffic. Some of the interfaces which the MMSC offers or connects to have been specified in much technical detail. It is important for the manufacturer of the MMSC to implement these interfaces in a correct manner so that software and hardware interoperability is guaranteed and that there are no trouble spots within the MMSNA. The most important interfaces in this regard are the MM1 and MM4 interfaces.

Since the implementation of interfaces MM2, MM6 and MM8 are not specified in detail but left instead to the manufacturers, there is more freedom and less pressure for interoperability in the implementation of these interfaces, thus giving room for maneuvre if time or other constraints become limited when designing and implementing the MMSC.

Also, support for the interfaces MM3, MM5 and MM7 is optional. Thus, a minimal-implementation MMSC must implement at least MM1 and MM4 interfaces according to the specifications. Because implementing interfaces for external servers or value added services is relatively easy and adds value to the MMSC implementation in the form of enhanced functionality, it can be expected that most MMSCs provide also support for MM3 or MM7 interfaces, or both.

The three manufacturers having the biggest marketshare as of 25th of October 2002 are Nokia, Ericsson and CMG [Mob2002], as shown in figure 8. As Ericsson and Nokia are the two highest ranking manufacturers, it is worthwhile to examine them in slightly more detail than the rest. Overall, it is difficult to find useful information about the MMSCs. This is understandable, as certain secrecy provides competitive advantage. However, trying to be too secretive about external server and application interfaces is not productive, as there can be no products that add value to the MMSC if such products cannot connect to it. Without content and service providers there is less content and fewer services, and without content and services there will be less customers, and without customers no business can work in the long run.

4.6.1 Nokia Artuse MMSC

The product description document for the Nokia Artuse MMS Center Release MC1 contains, among other things, a description of the architecture of the Artuse MMSC as

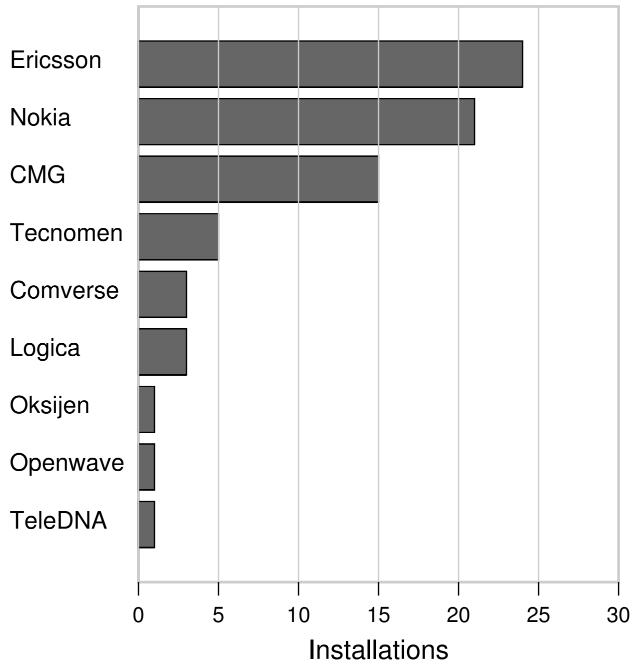


Figure 8: Number of MMS installations per manufacturer, including trial installations as of 25th of October 2002. [Mob2002]

well as descriptions of functionality provided by the MMSC. In the document there was also the mention of an HTTP-based Nokia proprietary protocol for the MM4 interface. [Nok2002b] It is unknown whether the issues described in the product description document are still valid, and whether an MM4 interface implementation compliant with the MMS functional description (i.e. based on SMTP) has already been adopted.

For the MM7 interface Nokia has specified a protocol with the name EAIF, an acronym for ”External Application Interface”. EAIF is based on HTTP 1.1 with persistent connections. The message payload is an octet stream encoded with the mechanism explained in the WAP-209 MMS Encapsulation document [WAP2002]. The payload is encapsulated inside a HTTP request in both directions (i.e. messages originating from and terminating to the external application). In addition to normal HTTP operation, there are a couple of Nokia MMSC specific headers, which deal with recipient and sender addresses, message type and identifier, charging information and MMSC status code and version. [Nok2002] This thesis and the implementation focuses on a Nokia product because Intellitel was a subsidiary of Sonera, and Sonera has bought its MMS solution (including MMSC) from Nokia.

4.6.2 Ericsson MMC

Ericsson MMC uses an SMTP-based MM7 protocol. [Eri2001] The Ericsson SMTP interface can function both as an external application interface as well as a VAS application interface. When sending messages to the Ericsson MMC via the SMTP interface, certain mail headers beginning with X-MMS- are used to, for example, set the expiry time and priority of the message. It was a pleasant surprise to note that the Ericsson SMTP interface was such that it allows the implemented EAIF-SMTP gateway to be made into a unidirectional Nokia-to-Ericsson VAS interface bridge with little extra effort.

Also, it is interesting to note that Ericsson chose to use an SMTP-based protocol for their VAS interface, whereas Nokia chose to use an interface based on HTTP. It is unknown whether this was intentional, due to possibly different intended usage scenarios, or the unfortunate result of having two protocol possibilities recommended in the MMS functional description. This situation will correct itself if (or when) the manufacturers implement interoperable VAS interface protocols.

4.6.3 Other MMS Centers

Other MMS Center manufacturers include CMG, Tecnomen, Comverse and Logica. There are some smaller players too, and also companies with a product or products similar to the Intellitel Messaging Gateway with an added MMS capability. First Hop is an example of such a company. Still, this thesis concentrates mainly on the Nokia MMSC, and does not concern itself with MMS Centers from other manufacturers.

4.7 Current Situation Among MMSC Manufacturers

It is difficult to predict the direction towards which the manufacturers are aiming to develop their MMS Centers. There appears to be a drive towards interoperability, but it is not clear whether this is the actual strategy preferred by the manufacturers. In the first versions of the MMS functional description the MM interfaces were not specified on a detailed enough level [ETS2001b, ETS2001, ETS2002b, ETS2002c]. This resulted in the manufacturers creating their own interfaces, following the specification where it was reasonable and beneficial for them. This situation is especially gaudy with Nokia and Ericsson MMSCs, which have a non-interoperable MM3 protocol implementation.

The biggest problem with the available MMSC material is that it is marketing material and it is not certain if the material contains facts or is a sort of ”augmented reality” view of the product. It is not easy to find factual information for comparing different MMSCs. It is difficult to even tell if an MMSC actually follows the MMS functional description, and more importantly, if it does, which version of the specification! Nevertheless, the biggest manufacturers seem to be leading the show with the smaller ones coming in their heels and trying to get their hands on whatever bits and pieces the big ones seem to miss.

4.7.1 Support for MM3 Interface

Since the MM3 interface is meant to interface with existing ”legacy” servers, it makes sense to implement to the MMSC at least the most commonly used protocols, i.e. HTTP and SMTP. However, support for MM3 interface protocols is not uniform among different manufacturers. HTTP and SMTP seem indeed to be the most abundant, although there exists also some support for certain SMSC protocols. If the MMSC has been developed by evolving an older SMS gateway product, support for an SMS protocol or protocols and at least some ”content protocol” (HTTP, SMTP) is a natural consequence. This is because such protocols are often supported by SMS gateways; SMSC protocols to interface with SMSCs and HTTP or SMTP to interface with content providers.

The operator’s needs and desires dictate the level of support for additional protocols at the MM3 interface. However, it might be the case that the manufacturer of the MMSC is not interested in providing a customized solution for supporting some obscure protocol. This kind of situation can be remedied with a message mediator which understands both the desired protocol and some MM3 protocol understood by the MMSC. A message mediator is likely to be, due to the very nature of the software, more suited to assimilating new kinds of protocols more quickly and easily than a complex MMSC product.

4.7.2 Support for MM4 Interface

The MM4 interface, which is used by two MMSCs to communicate with one another, is well specified in the MMS functional description. [ETS2002] This is because MM4 has to be interoperable among all MMS Centers. Otherwise there would no possibility to relay messages between two different MMSEs, which would make roaming in MMS networks impossible. Thus, support for MM4 is likely to be available in any given MMSC. For this reason, MM4 would be a good interface to implement into a mediator. By using MM4 to connect to the MMSC, the mediator would appear to the MMSC as an external MMSC. However, implementing full support for the MM4 interface is an order of a magnitude larger undertaking than implementing some of the MM3 or MM7 interfaces.

Even if relaying between MMSEs might be endangered by non-interoperable MM4 implementations, it is not certain if the different manufacturers actually even want total interoperability. [Nur2002] If MM4 does not become interoperable among all manufacturers, and the interface specifications are available, a mediation solution can be created to link two MM4 interfaces together. If the interfaces are proprietary, and specifications for them cannot be found, then mediation can be made to work via MM3 and MM7 interfaces. Although such a solution might be a little unorthodox, it should be plausible as long as the MM7 interface specifications are known and/or a common MM3 protocol is supported among the two communicating MMSCs. If MM4 becomes interoperable and everyone starts using it as such, advanced value added functionality (instead of protocol conversions) in an MMS-capable mediator can possibly justify investments by an operator.

4.7.3 Support for MM7 Interface

There is not much distinction between the intended functionality of MM3 and MM7 interfaces. The actual realization of MM3 can be proprietary as long as it is ”based upon existing standards e.g. HTTP, SMTP” [ETS2002] and follows the MMS functional description in other aspects. There is no reason why a HTTP protocol interface meant primarily for MM3 could not be used by content providers for two-way messaging with the MMSC. Although such a trick is technically possible, a more robust solution is to implement support for a proper MM7 interface.

Currently there are differing implementations for the MM7 interface. [Nok2002, Eri2001] The situation is likely to change in the future, as manufacturers gradually catch on to the latest versions of the MMS functional description. According to the MMS functional description, the interface for MM7 shall use SOAP 1.1 as the protocol [ETS2002]. Having SOAP as a mechanism with which to interface with any MMSC is a good thing for content providers and providers of value added services. With a SOAP implementation there is no longer a technological lock-up to a certain operator or MMSC manufacturer. In theory, this sounds like a dream come true for interoperability and ease of use. In reality, things are different, as it is up to MMSC manufacturers to decide whether or not they want to support SOAP at all. Adopting SOAP can enhance interoperability towards constantly developing web services and thus offer a future strong solution.

However, if MMSC manufacturers indeed start to support SOAP, there will be less need for such a message mediation solution that Intellitel Messaging Gateway currently provides. This is because the MM7/SOAP interface is standardized and based on XML, thus making non-interoperable extensions to the protocol itself practically impossible. In this case, there is a strong need to add value to the message mediation, as simple arbitration of messages and protocol conversions will not be enough to justify the investments. Even if MM7/SOAP will not gain widespread adoption, adding value to message mediation is nevertheless wise, as it can not only give a competitive edge, but also enable whole new products to be created as an off-shoot of the value-adding elements.

4.8 Requirements for an MMS-Capable Mediator

Creating an MMS-capable message mediator requires the implementation of one or more of the MM1, MM4, MM3 or MM7 interfaces. If there is desire to communicate with the MMS User Agent directly, then the MM1 interface must be implemented. However, implementing only the MM1 interface is useless, as communication with other MMS elements is also needed. Thus, if the MM1 interface is implemented, the MM4 interface should also be implemented. This is because the MMS message coming via the MM1 interface might be directed to a recipient in another MMSE. This also applies in the other direction, i.e. an inbound message coming from another MMSE.

In practice, implementing an MM1 interface demands much work, as it requires either a built-in WAP stack or an external WAP gateway, both requirements which were impossible to fulfill at Intellitel. Also, during initial research about the best way to make the Intellitel Messaging Gateway suitable for MMS message mediation, the MM4 interface specification of the Nokia Artuse MMSC was proprietary and not available from Nokia. [Nur2002]

Basing a message mediator on the MM4 interface alone is not a viable option. Implementing interfaces such as MM3 and MM7 can add more value to the message mediator even if the MM3 or MM7 interfaces are mutually non-interoperable among MMSCs. Also, ways to handle the MMS messages themselves are beneficial. If the mediator is to handle the MMS messages, by extracting data from the message, adding data to the message, or removing data from the message it needs to implement the WAP WSP-209 and, in practice, WAP-WSP-230 codecs. The MMS messages are encoded using these codecs [ETS2002, WAP2002]. If the mediator is to alter the order of the individual parts of the message, or change the aspects of the presentation in some other way, SMIL decoding and encoding [W3C2001] routines need to be implemented. An XML parser is also needed for SMIL handling.

Other requirements for an MMS-capable mediator are satisfied mostly by the existing message mediator. These include logging, statistics, alarms, high availability mechanisms, integration to billing systems, robustness, management functionality and other such features. If there is no existing message mediator product on which to base the MMS-capable mediator, these functionalities need to be implemented. Security aspects must also be taken into account. For example, the concept of CUGs (Closed User Group) can be useful in some services. A CUG is a whitelist, i.e. it lists the persons who are allowed to use certain services. The opposite of whitelisting is blacklisting, where certain users are banned from using the services. The mediator should provide white- and blacklisting capabilities.

There can also be other requirements, which depend on the nature of the value added functionality of the message mediator implementation. For example, if a transcoding implementation is realized, decoding and encoding capability for the transcoded media format is required. Also, transcoding implementation requires the codecs needed to alter the MMS messages.

The overall level of interoperability between MMSCs in the future is yet unclear. If the MMSCs cannot interoperate, there is a need for a mediation solution which can somehow link these two MMSCs and content providers together. One possibility is to connect two MMSCs together using the MM3 or MM7 interfaces. This scenario is the best for the message mediation product manufacturers as there will be a definite need for a mediation solution, making the situation analogous to what it has been in the SMS world. However, regardless of the actual interfaces which are implemented, the most important functionality of the mediator is the kind of functionality which adds value to the message mediation.

Implementing various value added functionalities can have the side effect of spawning new products and business opportunities. Some of the realizations to add value might become killer applications, if there will be a strong demand for such applications. However, foreseeing the killer applications and the demands for them can be difficult. For example, who could have known that the possibility to send an SMS and see it scroll by on TV after a while would become so successful, especially when using the service costs many times the price of a normal SMS?

5 Value Adding Functionality

This chapter explores and suggests different means to add value to message mediation, so that there would be a marketable case for message mediation in multioperator MMSNA regardless of how the interoperability issues of the MM interfaces develop.

The implementation in this thesis was an EAIF to SMTP gateway. Although acting as a simple gateway does not add much value as such, there is nevertheless a valid business case for the gateway. Operators who have not purchased or gotten an SMTP interface bundled with their MMSC may find it desireable to purchase the EAIF to SMTP functionality cheaper than the original MMSC manufacturer can offer. Also, operator-specific customizations can be implemented with less hassle and also possibly cheaper than when working with the big manufacturers.

The EAIF to SMTP gateway implementation gathers some information about the MM messages which pass through the gateway. The data gathered by the gateway can be easily extended to contain more information. The extracted data could then be used for billing or to provide statistics about the messages. Indeed, an almost trivial idea of adding value is to gather data for billing and statistics purposes. In practice, it might not be known beforehand what kind of data is needed. In such a case, the set of gathered data should include all data that is possible to extract, to anticipate the unforeseen usage.

Another way to add value to the mediator is to implement facilities for some kind of services. This way the mediator product can do some specialized (and possibly widely needed) services ”out of the box”, without the operator having to purchase or develop such services separately at a greater cost, effort and loss of time. This chapter also explores some possible services which could be implemented in order to add value to the message mediator product. Yet a third way to add value is to implement existing or totally new kind of functionality into the mediator, and combine this functionality with the previously presented ways to add value (i.e. billing, statistics and services).

5.1 Billing

For billing purposes it is wise to take as large a set of data as possible, to anticipate even the most specialized future usage. Enabling run-time configuration of the set of logged data allows for greater flexibility when bringing new kind of services to the market. If all the facilities to log data are already in place, it suffices to reconfigure the mediator instead of developing custom applications to gather data necessary to do billing. This lessens the time needed to bring new services to the market. If billing cannot be done, the service cannot make money.

MMS allows different kind of scenarios with regard to billing in messages. The MMS functional description contains dedicated functionality for reply-charged billing, in which the sender of the multimedia message agrees to pay for a reply to the original message, as long as certain criteria, such as the allowed time window for reply, are obeyed. Other scenarios, such as prepaid, postpaid and cost control, can be implemented using other mechanisms. The charging data gathered at MM1 and MM4 interfaces [ETS2002d] can be used as a rough guideline when determining what data to gather from the messages and transactions occuring at the MM3 and MM7 interfaces.

Since connecting to the MMS User Databases or other such databases might be difficult or even impossible, it is best for the billing mediator to have a dedicated database. The dedicated database has many benefits: first, it allows more complex billing logic to be implemented into the mediator. Second, it reduces coupling between the billing mediator and the MMSC, allowing for a more independent billing mediator implementation. For billing purposes, the database has to contain a mapping from some unique user identifier (e.g. telephone number) to the user's account.

The actual billing logic and the decisions how to handle the contents of the account depend on the billing scenarios, which are presented next. The term "generic data" used in the billing scenarios refers to a minimal set of data which is necessary to make the service possible. This generic data can include information like the size of the message, sender and recipient addresses, type of message, and so on. The precise composition of the generic data cannot be determined accurately, as it depends on many factors, such as the services themselves and the possible demands of local legislation. For example, for short messages the Finnish regulations define that the user can request pricing information for a service, but the cost for doing so may not exceed the price of a regular short message [FIC2002].

5.1.1 Postpaid Scenario

Postpaid scenario is the regular pay-as-you-go manner of billing. After the billing period, for example, once a month, billing data from the billed messages is compiled and a bill is sent to the customer. For this scenario, there are no special requirements unlike in the other scenarios. It suffices to gather all the generic data relevant for ordinary billing and store them for at least the duration of the billing period.

5.1.2 Prepaid Scenario

In a prepaid scenario, one or more messages have been paid beforehand. The user has an account which decreases with every billable message. Thus, the sender has to be known[7] so that the sender's identity can be mapped to the prepaid account and the billing logic be performed. If the account has no more credits, the user is denied from sending a message. In this scenario, at least data about the sender (e.g. MSISDN number) is needed. Other information can be used as well, such as size of the message, type of content of the message and type of message. [ETS2002] The most important information, however, is the data about the sender's identity.

Prepaid accounts are well suited for cost-conscious consumers, such as parents who want to avoid nasty surprises when paying for the telephone bills of their children. When using prepaid, the amount of money stored on the account is the ultimate limit to how much money can be used within some given period of time. Also, prepaid accounts are convenient when staying in a foreign country for a few months, as signing a ”real” contract with an operator might be impossible.

5.1.3 Cost Control Scenario

The cost control scenario is similar to the prepaid scenario. In cost control an upper limit for the bill is specified. The bill accumulates during use and is sent to the customer after the billing period. Unlike prepaid, which is paid in advance, the cost controlled bill is paid like a regular bill. Cost control ensures that the bill will never exceed some pre-defined value.

[7] Even a unique numeric identifier will suffice, using a real name is not necessary.

Like prepaid, cost control can be used by people such as parents of a teenager, who want to be sure that a certain limit is not exceeded during the billing period. In cost control, data about the sender is implicitly needed. This is because the customer's account needs to be verified so that the cost of the new transaction will not cause the accumulated cost to go over the pre-set limit. If the limit would be exceeded, the customer is not allowed to send the message. An appropriate barring message would be displayed in this case. The contents of the barring message should be configurable.

5.1.4 Reply-Charging Scenario

The reply-charging scenario is explicitly specified in the MMS functional description. In reply-charging the sender of the MM message is charged for the cost of the reply MM message. In other words, the recipient can reply at no cost. The sender may indicate special restrictions for the reply MM message, such as latest time of delivery and size of message. The reply MM message must be to the original MM message. This is controlled with the MESSAGE-ID header. The original MESSAGE-ID is indicated in the reply message. The MMSC will only accept one reply-charged reply to the original message. The reply must be text. Further restrictions, which may be alleviated in the future, is that the sender and the recipient must belong to the same MMSE. Forwarded MM messages are not to support reply-charging at all. [ETS2002]

Reply-charging can be used in scenarios where it is necessary to ask something from the customer. This can be, for example, a confirmation for a purchase whose value exceeds some limit set by legislation, or a query for some secret passcode to enable some service. For example, the customer could use some service which requires a confirmation in the form of a 4-digit passcode. In this case, the customer initiates the service by sending a service request using the usual methods. When the service replies, the reply-charged message instructs the user to reply with the passcode. The user can then reply to the message with the passcode, without having to pay for the necessary extra message. Thus, reply-charging can be used with a service which requires interactivity from the user, without the user having to separately pay for each intermediate message.

As reply-charging is part of the MMS functional description, it needs no additional data to be gathered for the billing. However, services using the reply-charging functionality may need additional data to charge for the original message. Because of this, it is prudent to gather as much data as possible.

5.1.5 Special Billing Scenarios

Some billing scenarios might require special logic. For example, a company could offer subsidized message prices during an advertising campaign. In this case, all messages sent during some time period would be free. In this case, the date and time of the message are needed for the billing. There can also be scenarios where messages sent to a certain recipient address will be free. Naturally, in this case, the sender and recipient addresses are needed. An extension of this scenario is a ”friend-ring”, in which messages sent to the people within the ring are subsidized during a certain period of time. Gathering data about date and time is relevant also here. Also, an operator could give the first 50 messages free for someone who signs up as a customer. This case is somewhat analogous to the prepaid scenario. However, unlike in the prepaid scenario, the user can continue to send messages even after the number of free messages have been exhausted. The additional messages would be billed normally in the regular bill.

5.2 Management

Using a mediator can under some circumstances be beneficial from the point of view of system management. When using multiple MMSCs from different manufacturers, their management interfaces might not be able to interwork. Even if they can interwork, management information and capability will not be available from a single point. An SNMP-capable mediator can offer management from a single point and integrate with existing legacy systems as well as new MMSCs. The downside for this approach is that it requires much work, as a number of possibly non-standard management interfaces would need to be implemented. Also, the approach might not be feasible for those MMSCs or legacy systems where the management functionality is nonexistent or closely tied to the insides of the products themselves. Although convenient from a system architectural point of view, a single point for management also effectively creates a single point of failure, which may be undesireable.

A better management-related approach is a mediator which provides a user-friendly configuration and management interface aimed at service providers. An example of this is the Service Provider Portal (SPP) component, which is a part of the Intellitel Messaging Gateway. [Int2002c] Although SPP works with SMS messages and a non-MMS-capable IMG, it is worthwhile to examine the architecture and functionality on a high level and think of how such an approach could work in MMS world.

The SPP allows the service provider to manage various parameters of the SMS services: keywords used (e.g. SMS messages beginning with the keyword WEATHER would send a weather report to the user), service type, status, access lists and price. The actual message sent by the user is delivered using HTTP or SMTP to a network entity indicated by a user-defined URL (Uniform Resource Locator) address. By using the SPP, the service provider does not need to know the technical details of the underlying Short Message Service. [Int2002c]

Figure 9 illustrates how a service provider could benefit from an MMS mediator which provides an MMS-capable SPP-like portal for service management and configuration. The portal is used with a web browser. The MMS mediator uses the information given by the user (the service provider administrator) when determining how to handle an incoming or outgoing message. The actual service data is located in the content service element.

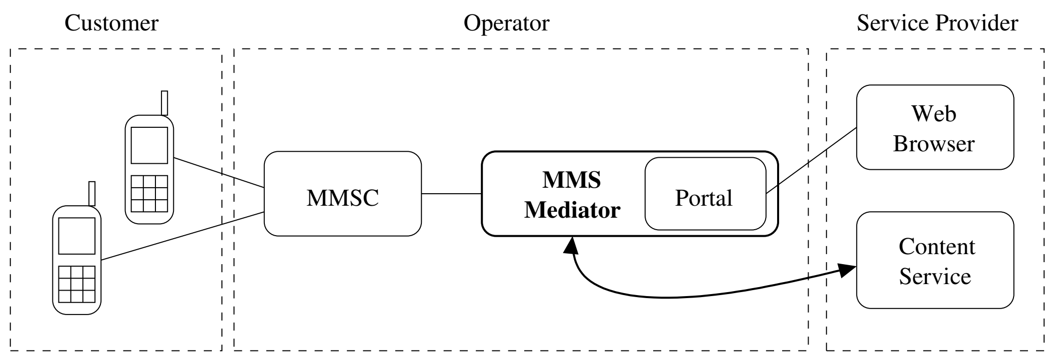


Figure 9: Managing MMS services with an MMS-capable SPP-like portal.

Such portal architectures, which are aimed at service providers, can extend the functionality of the MMSC by enabling faster and easier service creation. Also, as the service provider can manage their own limited, but sufficient, set of configuration parameters with a user-friendly interface, it removes the need for the operator to act as a ”proxy” between the configuration requests from the service providers and the configuration facilities of the mediator software. Thus service provider self-provisioning functionality in the mediator takes extra work away from the operator. Offering configuration and management functionalities for the service providers add value to the mediator by steering away from basic protocol conversions into more higher level functionalities, namely, the domain of services.

5.3 Statistics

For statistics purposes, it is necessary to first find out what kind of statistics are wanted and what are relevant. After that, the data related to the chosen statistics needs to be logged. Different kind of statistical applications require different sets of data to log. For example, from a network management point of view it might be relevant to know the amount of traffic in some time period, whereas from a marketing point of view the amount of users for some service is more relevant.

If it is not known beforehand what kind of statistics are needed and logging of *all* data is done as a precaution, the amount of logged data can be overwhelming. Thus, possibilities to trim down the amount of logged data is needed. The most flexible way to achieve this is during run-time operation, e.g. by using dynamic configuration directives which affect the behaviour of the statistic logging component. In demanding environments where high availability is essential, the changing of the configuration directives must not cause downtime in the entire system. Also, the inactivity time for single components should be as little as possible.

The mediator can also render aggregate statistics. These include, for example, the total amount of message traffic in bytes, the total number of messages with the type of ”Advertisement” and other accumulating attributes. If the mediator is unable to process aggregate statistics, the usefulness is somewhat diminished. This is not too critical an issue, as the gathered data probably needs to be post-processed with some external program anyway. The external program can be a sophisticated data mining application or a simple script which parses the logs for the statistics which the mediator is unable to provide.

The 3GPP specification describing charging data for application services defines that parameters such as message size, duration of transmission, sender visibility and message submission time are to be used when gathering data for CDRs. [ETS2002d] The document defines charging only in MM1 and MM4 interfaces, but can nevertheless be used as an inspirational help when defining what statistical data to gather. Some of the defined parameters are well suited to be gathered also for statistical purposes, even if they are not collected from the MM1 and MM4 interfaces.

5.3.1 Usage Statistics

For usage statistics, such as ”16-20 year olds send the most images on Friday at 19-21 o’clock”, data about each sender needs to be cross-referenced with an existing user database. It is assumed that the user database contains some personal information about the user, such as date of birth. This information is not to be used to track individual users but compute statistics and usage patterns and possibly do service barring. For example, an underage person is denied access to adult-oriented services. Aside from personal information, the time when the message was sent is also a crucial datum.

This kind of information is valuable for marketing purposes for many reasons. First, it gives accurate and up-to-date information about the market behaviour and conditions. Second, it can be used to better target advertisements to the target groups, with the aim of pumping up the usage even more. Third, special campaigns can be used to bring in non-subscribers who belong to the same demographic group(s) as the stereotypical heavy users. Other kind of usage statistics are also possible. From a technical point of view, having up-to-date usage statistics can be of help in improving network utilization. In general, usage statistics answer the questions ”who” and ”when”.

5.3.2 Traffic Statistics

Traffic statistics are used to answer questions such as ”what part of the traffic is advertisements” and ”how big is an average MM message”. Traffic statistics can also show how big a part of the messages are person to person messages, requests for some service and replies from a service. Also, the content and nature of the messages can be discovered. These data includes, for example, the number of images in the network, size of an average image, the number of sent sound files and the percentage of users utilizing the MMS streaming capability (if supported by the network).

Like usage statistics, traffic statistics can give insight to and aid marketing-oriented usage. From a technical point of view, traffic statistics, like usage statistics, can discover potential bottlenecks and thus help improve network utilization, e.g. via message prioritization. Message prioritization works by preferring some messages over others, allowing the preferred messages to be delivered and processed before than the non-preferred ones.

5.4 Content Transcoding

Mobile terminals are small and lightweight so that they can easily be carried around. A small form factor and a limited power supply places restrictions on the capabilities available in the terminal. For example, the small memory size limits the size of the messages the terminal can handle and the processor in the terminal is not meant for heavy all-purpose calculations, limiting the complexity of the messages. The size of the screen in the mobile terminal is often small and limited in the number of colours it can display.

Due to these limitations, it might not be possible to scale a huge image to a smaller version as the original image might not fit completely into the memory at all. Also, scaling a video clip at the terminal is a computationally intensive and memory consuming task. To spare the mobile terminal from having to do these gargantuan computational tasks, the content can be modified to be suitable before it is sent to the terminal. This is what content transcoding aims to do. The purpose is to adapt the content to fit the hardware profile of the terminal. The burden of adaptation is then taken from the mobile terminal and given to another device, which is able to process data much faster and has many times more memory.

Content transcoding is concerned with tasks such as reducing the number of colours in a picture, thus making the image smaller (at least when an indexed format such as GIF is used). Also, the resolution in a video clip and the sampling rate of an audio clip can be reduced so that the files become smaller. The device doing the content transcoding has to know the parameters of the target terminal. For this, the content transcoder should have access to the profile of the mobile terminal. The terminal capability profile is given to the MMSC by the MMS User Agent and is stored inside the MMS User Databases [ETS2002]. Thus, the content transcoder needs access to the MMS User Databases or has to implement some other method of receiving the terminal profile.

A realization of a content transcoder could utilize the MM3 or MM7 interfaces. In both cases, the content transcoder would act as a filter, receiving a message (and the terminal profile), transcoding the content, and then sending the transcoded message back to the MMSC. An MMS-capable mediator can function as a content transcoder as long as certain requirements are fulfilled. For example, the content transcoder has to be fast and have a large memory and sufficient mass storage. Otherwise, many users sending simultaneously a large amount of messages requiring content transcoding would create an accidental denial of service attack against the content transcoder.

Even if a fully functional content transcoder is not built, implementing even some basic content transcoding functions can add value to the mediator product. If the mediator is capable of receiving e-mail messages from the Internet, for example when acting as a bidirectional EAIF to SMTP gateway, the possibility to transcode images would be especially helpful. The user could use a digital camera to send an image directly as a mail message, without thinking about the size of the image. The mediator, upon receiving the message, could either use the terminal profile information or it could use the ”lowest common denominator” for the images [CMG2002], making the image small enough to be accessed with any mobile terminal. In any case, the extent of the transcoding functionality should be configurable, because the MMSC might have a dedicated content transcoder, in which case the MMSC will take care of the transcoding and the original message should be passed intact to the MMSC.

5.5 Positioning

The HLR, an element of the GSM network architecture, contains information about the current location area of the mobile terminal. This information is needed in order to localize the user in the worldwide GSM networks [Sch2000], and enable roaming i.e. using the terminal in networks other than the home network. If the message mediator can interface with the HLR, it is possible to add location-aware functionality to the mediator. In practice, interfacing with the HLR requires knowledge of MAP. Companies which already possess SS7 knowledge have likely already developed software for MAP connectivity. Other companies will need to use more effort, but this pays off in the end, as positioning-based services have huge business potential and as such, supporting them in the mediator can add much value to the mediation process and to the product itself.

Although HLR positioning is not as accurate as that provided by GPS (Global Positioning System) receivers, HLR location area data has the advantage of being an intrinsic property of the GSM network. Thus, HLR location area data is always available regardless of the handset capabilities. On the other hand, GPS receivers are often implemented in the handsets only. In order to get more accurate GPS positioning data from a GPS-enabled handset, the handset would have to inform the network of the location upon receiving a request for location data. This kind of functionality would require changes to the handsets and the network. As time goes by, methods to pinpoint the location of the handset using GPS-capable handsets and other schemes will surely be developed and standardized. In the meantime, HLR positioning can be used, even if it lacks high accuracy.

Support for positioning functionality in a mediator will bring new dimensions to the services and the possibilities available. With positioning concepts such as time/location-based billing would become a possibility. This would make it possible to charge differently the messages sent during events with much people, such as concerts, festivals and sports events. These events are often concentrated on one well-specified geographical area with well-known starting and ending times. From the point of view of the user, the changes in billing would be transparent. There would be no need for the users to register beforehand, aside from possible legislation-defined procedures to grant the operator access to the user's location data.

Also, receiving a colour map image of the current location, or multiple images showing the path to the chosen location, can be done using positioning data and access to a cartographical service. Such a service would be especially useful to people who do not know the surroundings well and need navigation assistance: e.g. tourists, truckers and people from out of town. Combining the navigation assistance service to a database about local businesses would allow queries such as ”where is the nearest Chinese restaurant” or ”where is the nearest pharmacy”. Such a service, assuming it is simple and easy enough to use, would aid local businesses, big and small, as long as data about their line of business and location is up-to-date and readily available.

5.6 Advertising Campaign Gateway

It is possible to modify or add new parts to the message as it goes through the mediator, if the mediator is able to extract data from or modify the MMS messages. To accomplish this, the mediator needs to implement suitable codec routines. The codec consists of routines for the WAP WSP-209 MMS encapsulation [WAP2002] and also in practice the WAP WSP-230 WSP codec [WAP2001] routines, which are used by the WSP-209 encapsulation. Also, the SMIL part of the message may need to be modified in order to implement proper delay and other presentation parameters for the added content.

If the mediator acts as an unidirectional gateway, it suffices to implement either encoding or decoding functionality, depending on the direction of the gateway. Decoding functionality is enough for a gateway which receives MMS messages and sends messages using some other protocol. Likewise, for a gateway which only sends MMS messages, and receives messages using some other protocol, encoding functionality is sufficient.

The EAIF to SMTP gateway implementation done in this thesis has routines to add an extra part to the outgoing multipart SMTP message. The extra part is a simple text ”This MMS message brought to you by Intellitel Messaging Gateway!”. This part is the first thing the user sees upon opening the mail message. Generally, adding an image, text or sound to the multimedia message could be used to implement an ”advertising campaign gateway”. Such a gateway would add a sponsored logo (still image), slogan (text) or jingle (sound) to the message and mark the modified messages to be subsidized in the billing data. This allows for cheaper messages to the user and visibility to the company paying for the advertising.

From the point of view of the user, the usage of an advertising campaign gateway has to be completely voluntary. The user has to understand that the content of the messages will be slightly modified. However, since the advertising party pays part of the message, the cost of the message is reduced, which is advantageous to the user. Based on this information the user should then make a decision whether to start using the service or not. Making the system voluntary instead of involuntary is essential. Involuntary message content modification, even with subsidized message prices, is likely to make customers unhappy. Also, whatever the added content is, it has to be relatively subtle so that it is definitely noticed but not too much or it will disrupt the messaging experience.

Combining the advertising campaign gateway concept with other functionality can be beneficial. For example, implementing a location-aware advertising campaign gateway can have interesting applications. The advertisement chosen to display could be the nearest advertising party at the location of the receiver. If there are more than one companies available, as is the case when the receiver is e.g. at a shopping mall, the company could be chosen so that all companies get equal share of visibility, or so that the highest paying company could get the visibility with a higher probability than the others. The highest paying company must not get the advertisement shown all the time, as the others would be paying for nothing.

Figure 10 illustrates on a conceptual level how a location-aware advertising campaign gateway works. The HLR is informed of the recipient’s location. The original message, containing a greeting in this example, is modified en route to contain also an advertisement for a company at the recipient’s location. There are multiple companies at the recipient’s location. Although company B had a smaller probability of being selected due to cheaper pricing of the service, it was selected this time.

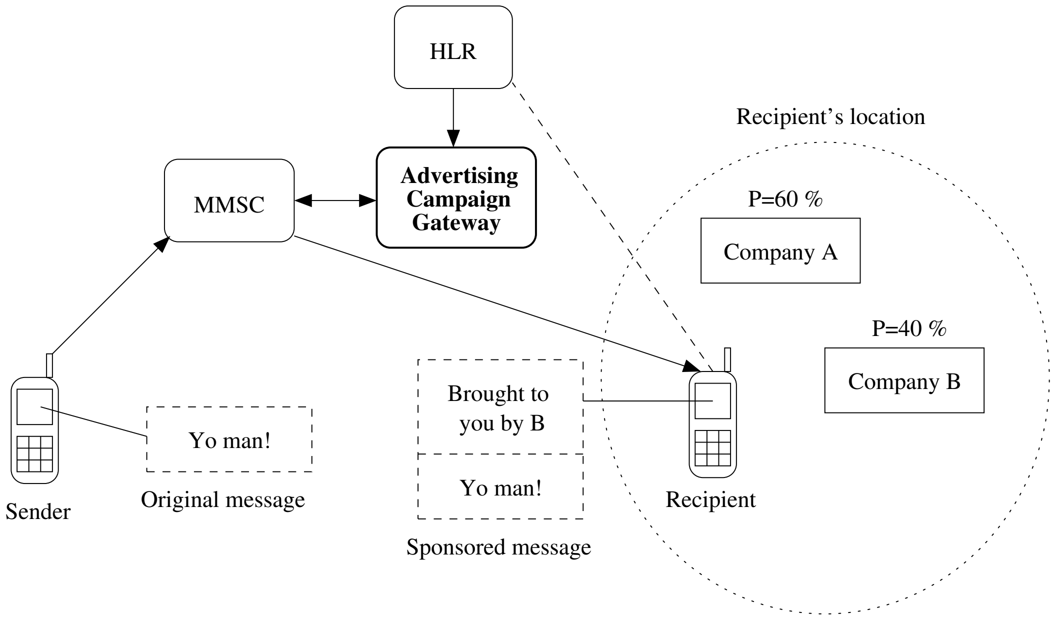


Figure 10: Location-aware MMS advertising.

If the advertising campaign gateway functionality is implemented, additional marketing research is needed to find out the proper pricing structure for the messages sent via the gateway. It is unlikely that companies would be willing to pay many times the price of a regular MMS message just to get visibility. Sane pricing and user acceptance is the key to success.

The advertising campaign gateway has a certain analogy to the "free e-mail" services such as Hotmail. In those services, the user is given a free e-mail address which can be accessed anywhere using a web browser. Small commercials are placed to the end of the messages. Also, upon signing up, the user has the option of receiving advertisements from the partners of the service. Thus, with the help of paid advertisements, an e-mail service is provided to the user at no cost.

5.7 MMS Voting

Reply-charging is a billing scenario where a reply to a message is free of cost, assuming certain constraints[8] are honored. Reply-charging is ideal for situations in which feedback from the user is required. Since the reply is free of charge, the only thing a reply requires from the user is some of her time. This makes reply-charging well suited for voting

[8] E.g. the reply message is not sent too late and is a reply to the original message.

and gallup[9] purposes, as the user might find it easier and more convenient to respond. Reply-charging is not an MMS-only specialty. It could just as well be implemented with SMS messages. However, reply-charging is defined in MMS functional description. There are also other advantages with MMS messages.

The difference between SMS voting and gallup applications is that MMS offers diverse media formats and larger messages. Whereas SMS is enough for a reply consisting of a simple answer, perhaps only one character in length, SMS is ill-suited for the presentation of the actual question. For example, if a person were to vote in a song contest, it would be necessary to hear the song beforehand on e.g. television. With MMS messages the parts of each contestant song (or a video of the performance) could be short clips in the message itself. This would make it possible to vote for the artist anywhere without having to sit by a television. Other kind of contests, e.g. beauty contests which place heavy emphasis on visual representation, can also benefit from voting implemented with multimedia messages. The user could actually see the contestants as images embedded into the message.

Implementing voting logic to an MMS-capable mediator is relatively simple. The mediator has to be capable of originating an MMS message, possibly do reply-charging, associate the replies (identified with some keyword) as votes and store the results for later use. Additional functionality might be the offering of graphical representations of the results as they are received. Such graphical representations could be shown during the voting to add to the excitement of the voters.

For gallup purposes, extra logic is needed. If there are many questions it is pointless to send a huge list to the user. The gallup logic could break a long list of questions into subsets containing two or three questions, and handle the aggregation of the answers. The gallup logic needs to ensure that the question subsets are distributed suitably. The distribution could be an uniform distribution so that all questions get an equal share of answers from users. Alternatively, the questions could be distributed in suitable ”tournaments”, where each pair or triplet (or a larger subset) chosen among the set of all questions is compared against each other. The tournament approach is motivated by the tournament selection mechanisms used in evolutionary algorithms [Ban1998]. The distribution method used in the gallup logic depends on the nature of the questions. For example, if the point of the gallup is to find the top three companies from a set of 100 companies, based on some subjective criteria (e.g. media visibility, or the question ”where

[9]”Gallup” is preferred over ”polling”, as the latter has certain connotations.

would you rather work”), then the tournament approach could work. For other kinds of questions which require the user to answer many individual questions in order to give meaningful results, it might not work. It is up to the gallup logic implementation of the mediator to anticipate various kinds of scenarios.

Whatever the approach might be, voting or gallup functionality can be implemented to add value to the mediator. Even in their rudimentary form such functionalities could prove to be useful. Voting and gallup questions rendered with multimedia messages offer much more choice for presenting the questions, as visual and aural media can also be sent to the user.

5.8 Digital Photos to Mailbox

There are high expectations that digital photos taken with a camera-equipped MMS terminal will become a killer application for MMS. [Mob2000] MMS to mail conversion is relatively basic functionality. Creating a mediator which is capable of sending digital photos to an e-mail box requires that the mediator is capable of sending mail messages, i.e. SMTP capability is required. Also, since there is a conversion from WAP WSP-209 encapsulated MMS message [WAP2002] to a mail message, the mediator has to be able to decode the MMS message and then encode it as a MIME multipart mail message. As the decoding and encoding can work with any supported media type, the mail message can also contain media other than still images. Also, the MIME content types are the same in both the WSP-209 encapsulated messages and mail messages, which eases the mapping from/to MMS messages to/from e-mail messages, regardless of the media formats used.

An SMTP interface can often be bought along with an MMSC. However, it is not mandatory to buy such functionality. Even if the operator already has an existing SMTP-capable MMSC, investing in an MMS-capable mediator is worthwhile if it can offer more functionality. Also, the mediator could be offered cheaper than the SMTP interface from the MMSC manufacturer.

Being able to send MMS messages into a mail box is convenient. Ordinary textual MMS messages can then be seen as ”mobile e-mail”, which can be sent at any time from any place the mobile terminal can connect to the network. The pictures and sound contained within MMS messages offer even more power of expression than ordinary textual messages. If there is also capability to receive e-mail as MMS messages, instead of just sending them, a transparent (from the user's point of view) bidirectional e-mailing functionality is achieved. Even if the e-mailing functionality is unidirectional[10], the system is useful, as it allows basic e-mail connectivity with an MMS-capable handset. The implementation done for this thesis was a component which implements the MMS to e-mail conversion functionality.

[10]That is, e-mail messages can only be sent or received (but not both) as MMS messages.

6 Implementation

The MM7 interface was not specified in a detailed enough level in the early MMS functional description documents. Although the specification was lacking, Nokia provided a well-specified protocol for the MM7 interface of the Nokia Artuse MMSC. The choice for using Nokia EAIF protocol was mostly because the EAIF protocol was seen as the fastest way to get usable MMS capability. The implementation done for this thesis was a unidirectional EAIF to SMTP gateway, implemented as an Intellitel Messaging Gateway message mediator module. Unidirectionality in this case means that only those messages originating from the mobile terminal are supported. In SMS terminology, this means MO messages. In EAIF terminology, the gateway functions in application-terminated mode. From the point of view of the MMSC, the gateway is a VAS application where the message flow stops, hence ”application-terminated”.

The implementation was done with CVOPS (C Virtual Operating System). The existing HTTP protocol interface module was used to implement the transport part for the EAIF implementation. The existing SMTP protocol interface module was used for the SMTP interfacing functionality. Due to the modular design of the Intellitel Messaging Gateway, the existing protocol interface modules could be re-used quite easily. Most of the troubles encountered with the implementation were ultimately related to scarce and difficult to find documentation.

6.1 CVOPS

CVOPS is a library, tool and run-time environment for implementation of communications protocols using the C and C++ languages. CVOPS provides, among other things, timers, routines for internal communication between CVOPS entities, utility routines and an interpreter for a macro language which all aid in the creation of protocols. In CVOPS, a protocol is implemented using so-called virtual tasks, or ”vtasks”, which are ”objects” consisting of a state machine and the data and functions used by the state machine. The virtual tasks communicate using message passing. CVOPS takes care of delivering these messages to the intended recipient or recipients. A protocol stack consists of one or more virtual tasks in each protocol layer. [Cel2001]

CVOPS is displayed on a conceptual level in figure 11. The virtual tasks are displayed in the upper part of the figure. CVOPS itself and the services provided by CVOPS are visible

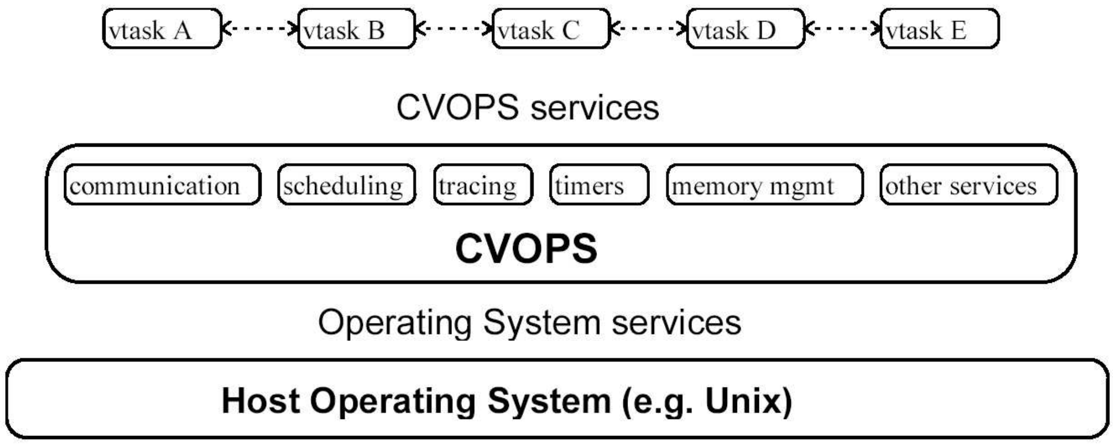


Figure 11: CVOPS conceptual architecture. [Cel2001]

in the middle part of the picture. CVOPS runs over the underlying operating system, and uses the services provided by the operating system. The figure shows only a very high level view of CVOPS. For example, the different virtual task types (entity and connection) are not shown.

6.2 SMTP Module

The SMTP protocol interface module was partly written by the author for a previous version of the Intellitel Messaging Gateway. Due to internal requirements and extensibility, the protocol interface was implemented in two parts: a ”driver” part which implements the actual SMTP protocol and an ”adapter” part which commands the driver part and interfaces with the Intellitel Messaging Gateway. Despite the architectural duality, the SMTP protocol interface module can be seen as one module which implements SMTP functionality. The situation is analogous with the HTTP protocol interface module, which has a similar internal structure. The SMTP module supports the minimal SMTP implementation as specified in RFC2821, i.e. the commands EHLO, HELO, MAIL, RCPT, DATA, RSET, NOOP, QUIT and VRFY [Kle2001].

The SMTP module supports also base64 encoding when dealing with MIME messages. Other encoding methods, such as quoted-printable, are currently integrated into the SMTP module. However, the integration was not complete at the time of the EAIF to SMTP gateway implementation, and thus there is no support for the quoted-printable and other encoding methods in the gateway.

Multipart MIME messages were not properly supported in the SMTP module during the implementation of the EAIF to SMTP gateway. This is mainly due to the reason that there has not been much need to support such features. The multipart functionality was originally due to be implemented, but was abandoned due to time constraints. Therefore, the implemented SMTP to EAIF gateway has to do a little maneuver to get around this restriction. The outgoing message is not given to the SMTP module in multiple parts internally. Instead, a proper multipart mail message is formed at the EAIF part of the implementation. This way the multipart message can be transferred as a pair containing the headers and the message body. In this case, the headers already include the appropriate MIME headers[11], and the message body contains the multipart message ”flattened” into one part. Relevant boundaries and headers are written into the single part message (which was transformed from the original multipart message) for each multipart element.

All in all, the SMTP module provides transparent RFC2822 functionality. The EAIF module need not concern itself with the intricacies of SMTP protocol or message encoding, except for the workaround for the restrictions in handling multipart MIME messages.

6.3 EAIF Message Mediator

To interface with a Nokia MMSC, an EAIF protocol interface was implemented. EAIF is the MM7 (VAS application) interface for Nokia MMS Centers. EAIF uses HTTP 1.1 with persistent connections as the transfer protocol. The MMS messages are embedded inside the HTTP request.

From the point of view of the MMSC, the VAS application connected via EAIF can work in three application types: terminating, originating or filtering. The last one is actually a combination of the first two. A terminating application is such that a multimedia message is delivered to the VAS application and the message will go no further, i.e. the VAS application is an end point for the multimedia message. An originating application is such that a multimedia message comes from the VAS application to the MMSC. A filtering application receives a multimedia message from the MMSC, thus acting as a terminating application, after which the VAS application modifies the message or extracts data from it, and then sends the message back to the MMSC. [Nok2002] Figure 12 illustrates the

[11] Such as CONTENT-TRANSFER-ENCODING, MIME-VERSION, CONTENT-TYPE with proper boundary, etc.

application types used in EAIF. The numbers illustrate the sequence of the messages in the case of filtering applications.

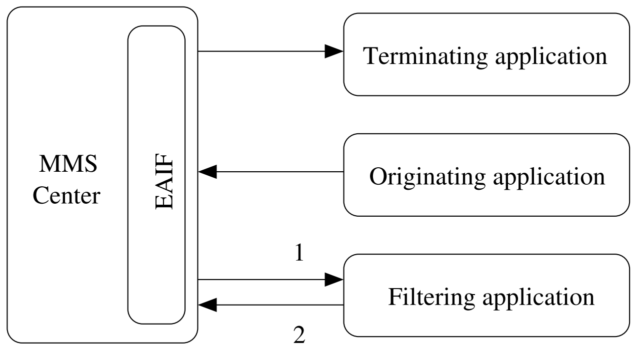


Figure 12: EAIF application types. [Nok2002] Augmented.

Terminating and filtering applications can work in two modes: synchronous and asynchronous. In synchronous mode, a response follows immediately the original request. Thus, synchronous applications can only handle one request at a time, per connection. An asynchronous mode allows a response to come at a later time. In this case, the original message is identified using a message identifier inside the X-NOKIA-MMSC-MESSAGE-ID message header. Many asynchronous messages can be sent sequentially, as there is no need to wait for the response after sending. [Nok2002] The implemented EAIF protocol interface is capable of working in synchronous terminating application mode only. Such a mode is sufficient for a unidirectional gateway.

EAIF messages are HTTP POST requests, with the multimedia message as the payload of the message. The mandatory HTTP headers are HOST, CONTENT-TYPE and CONTENT-LENGTH. Also, because the EAIF protocol is closely coupled with the MMSC, and the MMSC needs to interact with the VAS application, there are certain extension headers used in the HTTP POST messages. All of these headers start with the stem X-NOKIA-MMSC-. Table 4 summarizes the headers and the meaning of each header.

The message body is encoded using the WAP WSP-209 MMS encapsulation [WAP2002], which specifies MMS-specific extensions to the WAP WSP-230 encoding [WAP2001]. This binary encoding of the actual message contents is the same as used in the MM1 interface. EAIF supports two WAP WSP-209 protocol data units (PDU): M-SEND-REQ and M-DELIVERY-IND. These are encoded with values ”MultiMediaMessage” and ”DeliveryReport”, respectively. The type of PDU is indicated as the value of the X-NOKIA-MMSC-MESSAGE-TYPE header. [Nok2002] In a unidirectional EAIF

Table 4: HTTP extension headers for EAIF. [Nok2002]

X-NOKIA-MMSC-	Meaning
MESSAGE-ID	Unique identifier for the message
STATUS	Processing status for messages from asynchronous applications to the MMSC
CHARGING	Tariff class number for billing
CHARGED-PARTY	Defines whether the sender or recipient is charged
TO	Recipient
FROM	Sender
MESSAGE-TYPE	EAIF protocol message type
VERSION	EAIF version

to SMTP gateway it was sufficient to implement handling for the M-SEND-REQ (or ”MultiMediaMessage”) only, as the implementation acts as a terminating application, and no delivery report indications are sent to the MMSC.

6.3.1 Functional Description

The structure of the EAIF to SMTP gateway is described in figure 13. Note, that the word ”IF” in the picture is a mnemonic for ”interface”. A concise description of the process is that the MMSC sends an EAIF message, which goes through the HTTP protocol interface module to the EAIF protocol interface module. The message is decoded and an SMTP message is formed in the EAIF module. The SMTP message is then passed forward in the system, and eventually it ends up in the SMTP protocol interface module, which transmits the mail message to a mail server, where it will end up in the recipient's mail box.

A slightly longer, but more detailed description begins as follows: first, the MMSC sends an EAIF protocol message over HTTP to the VAS application configured in the MMSC. This VAS application is the EAIF to SMTP gateway. The TCP/IP Module takes care of socket-level TCP/IP communication. It passes the received data, i.e. a HTTP request, to the HTTP Driver module using the TCP/IP interface. The HTTP Driver takes care of the HTTP protocol and interprets the received HTTP request. Then, the HTTP Driver passes the decoded request in an internal format to the HTTP Adapter module.

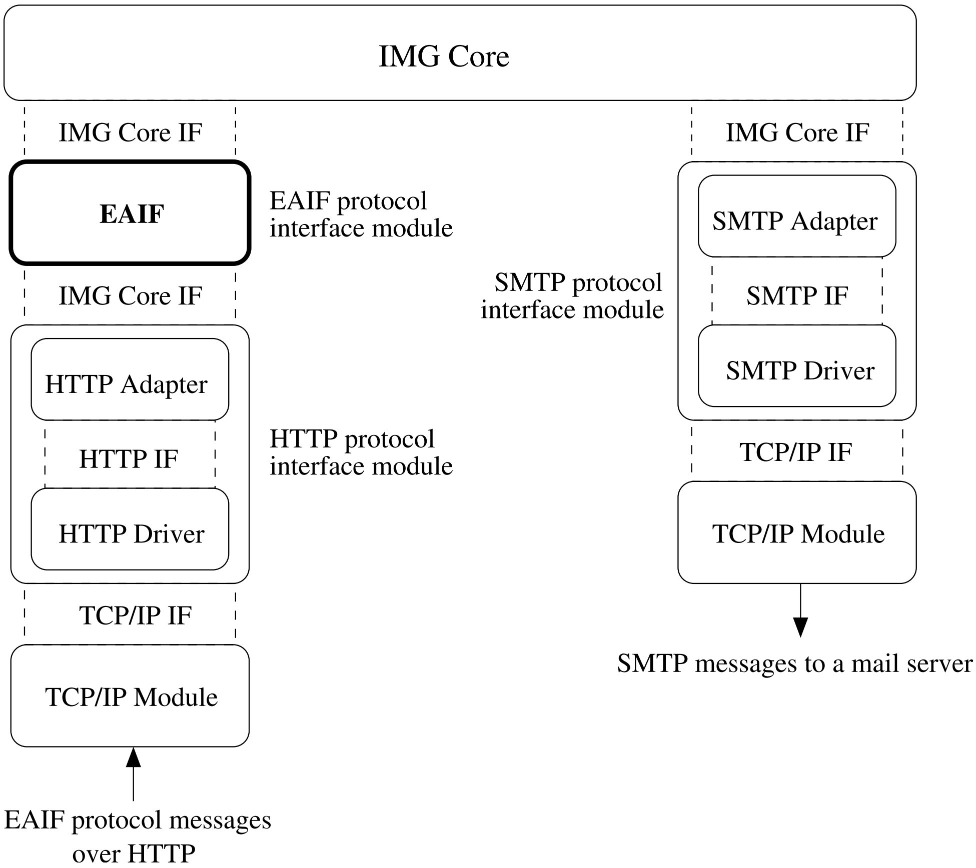


Figure 13: EAIF to SMTP gateway architectural structure.

The HTTP Adapter module normally passes messages to the IMG Core using the IMG Core interface. The HTTP Adapter also transforms message acknowledgements to successful HTTP return codes, which are passed ”downward” back to the HTTP Driver. In order to not have to use the HTTP Driver directly using the HTTP interface, the EAIF protocol interface module was implemented as such that it is placed between the HTTP Adapter and the IMG Core. This was the ”path of least resistance”, and offered the simplest approach of making the EAIF module capable of using the available HTTP functionality. Also, this way the EAIF module will not have to deal with HTTP return codes, as normal HTTP failure/success return codes are compatible with EAIF.

The EAIF module decodes the received WAP WSP-230 encoded message and transforms it into a multipart (or single part, if such a message is received) MIME message. The MIME message is then given to the IMG Core in a proper format. Using internal routing mechanisms, the IMG Core passes the MIME message to the SMTP Adapter, which is similar to the HTTP Adapter in the sense that it takes care of interfacing with the IMG Core and offers a uniform interface for the processes who wish to use the SMTP protocol interface module functionality. The SMTP Adapter then passes the message, already in proper format, to the SMTP Driver, which takes care of the SMTP protocol level intricacies. The SMTP Driver connects to the proper MTA (Mail Transport Agent, i.e. mail server), and conducts an SMTP message transaction. After this, the message goes through proper SMTP mechanisms (local delivery to the user or possible relaying forward to other mail servers) to arrive at the mail box of the intended recipient.

The recipient may then view the multimedia message in her MUA (Mail User Agent, i.e. mail program). There is a caveat here: unless the MUA can show SMIL files, the multimedia message will not display in the same fashion as it does in an MMS terminal. However, all the parts of the multimedia message, such as sounds, images and so on, will be displayed as normal attachments and can be viewed separately. Appendix 1 shows a multimedia message as a mail message inside a MUA.

The EAIF to SMTP gateway also takes the liberty of adding a short text to the beginning of mail message. This message reads ”This message brought to you by Intellitel Messaging Gateway!”. The message need not be textual. It might be a user-configurable picture or a small banner commercial. Such functionality is easy to implement in the EAIF to SMTP direction. However, in the reverse direction (SMTP to EAIF) the situation would be slightly different. If the image or text is to be a part of the multimedia message then routines to modify the SMIL part of the multimedia message are required. If the SMIL part is missing in the original message, i.e. the message is a single part message, and a presentation differing from the handset defaults is required, there must be routines to add a valid SMIL part. Note, in this case adding an image to an originally single part message will result in a total of three parts: the SMIL part, the new image and the original single part. Thus, the original single part message becomes a multipart message. This only applies to the bidirectional case and the case when SMTP messages are transformed into MMS messages. As the SMIL part is optional, it can be left out when adding content to an originally single part message. In this case the handset will present the multiple parts as it sees fit.

With very little or no changes the EAIF to SMTP gateway could be made into a (unidirectional) mediator between the external interfaces of Nokia and Ericsson MMS Centers. If the functionalities required by bidirectional operation, i.e. the encoding parts of the codec and changes to the protocol automaton, are implemented then the gateway can act as a properly two-directional gateway between the two external interfaces with small changes to the EAIF protocol part and the overall program logic.

6.3.2 WAP WSP-209 Codec Implementation

The WAP WSP-209 codec [WAP2002] defines an encoding for MMS encapsulation. In practice, it uses the WAP WSP-230 encoding [WAP2001] extensively. Thus, the implementation of the WAP WSP-209 codec required also to implement the codec routines defined in the WAP WSP-230 document. Note that when talking about the WAP WSP-209 codec, the meaning is collectively the implementation of both the WAP WSP-209 and the WAP WSP-230 codecs. The implementation of the codec was a bit octet stream scanning state machine with one octet lookahead. The lookahead is needed for some types, such as text strings. For each possible encoded type, a routine to decode the given type was written. The decoding routine for each type thus became a function. The octet stream and various other internal data were passed in a context area (a struct).

The high-level functionality of the state machine is as follows: the state machine scans the octet stream representing the encoded message. Depending on the type at the current position, e.g. DATE-header, the state machine decodes the type and maps the octet(s) into an RFC2822 compliant mail header name. The state machine then calls a decoding routine for the type. The decoding routine extracts an appropriate number of bit octets according to their type, and maps or converts the value represented by the octet(s) into an RFC2822 compliant mail header value. Integers and other non-textual values are converted into text. When data parts are encountered, they are base64-encoded and added to the aggregated mail message body.

Due to current limitations in the SMTP protocol module in handling multipart data passed to it internally, the EAIF protocol interface module had to form a proper mail message by itself. As the state machine worked its way through the octet stream representing the message, the decoded body data was placed into a CVOPS FRAME [Cel2001] data structure. The decoded headers were placed into an internal header structure. After handling the entire stream, the contents of the two data structures were a proper RFC2822 multipart mail message, with headers and body stored separately.

6.3.3 Implementation of MMS Addressing

To deal with MMS addresses, a simple address rewriting functionality was implemented to the EAIF to SMTP gateway. The address rewriting maps MMS addresses to SMTP mail addresses. It is possible to map the sender and recipient in both MO and MT directions.

Since the EAIF to SMTP gateway is unidirectional in application-terminated mode, which corresponds to MO direction in SMSC protocol parlance, the MT direction mapping is currently unused. No messages are sent in the MT direction. The rewriting functionality was already implemented in the SMSC protocol interface modules, and had been made into library routines for modularity purposes. The implementation of the address rewriting using the library was therefore relatively straightforward.

The address rewriting can be done separately for each of the MMS address types: mail addresses, PLMN numbers and IPv4 and IPv6 addresses. The free-form type can also be used. The type is based on the value of the /TYPE= parameter, which is given after the MMS address. [WAP2002] If the parameter is missing, the address is assumed to be a mail address. The rewriting functionality is controlled via configuration parameters in the system configuration file. The purpose of address rewriting is not to do fine-grained mapping of e.g. phone numbers to e-mail addresses on an individual basis, but instead to do a more coarse, or high-level, mapping. For example, all phone numbers beginning with 42 could be made to map into the domain ALPHA.OPERATOR.COM, and the phone numbers beginning with 69 could be made to map into the domain BETA.OPERATOR.COM.

The rewriting works using regular expressions. There are two configuration parameters per address to rewrite: the matching parameter and the rewrite parameter. If the regular expression in the matching parameter matches successfully, the address is replaced with the rewrite parameter. Submatches can be used. For example, a match for ^040-(.+) and rewrite to \1@GAMMA.OPERATOR.COM would result in mapping the PLMN number 040-12345678 to 12345678@GAMMA.OPERATOR.COM. Here the \1 denotes the first grouped match, i.e. the first part in parentheses in the original regular expression. If there were two or more parts in parentheses, the first part would be \1, the second \2 and so on.

The address rewriting functionality is not capable of mapping individual MMS addresses to individual mail addresses. This could be remedied by implementing suitable database lookup routines to the EAIF to SMTP gateway. Before a more elaborate mechanism of mapping the addresses exists, it is recommended that an external MTA be used in conjunction with the gateway. The external MTA likely has more capable mechanisms, such as database lookups, for mapping e.g. a phone number to an e-mail address on an individual basis. There are also other reasons, for example, if there is a need to add mail routing headers to the message, it cannot be easily done in the gateway, as such functionality has not yet been implemented into the SMTP protocol interface module.

6.3.4 Limitations

The EAIF to SMTP gateway has certain functional limitations, the biggest being the unidirectionality. The gateway can act only as a terminating application, processing messages which originate from the MMSC. The gateway cannot receive mail messages, encode them into an MMS message and send it to the MMSC as an EAIF message. The gateway only works one way. However, the other limitations are not as major.

Support for bidirectional messaging in the EAIF to SMTP gateway would require the encoding functionality in the WAP-209 codec part of the gateway to be written. The current unidirectional functionality required only decoding functionality in the codec. Also, for bidirectionality, modifications to the part of code interfacing with the SMTP protocol interface module needs changes with regard to the handling of multipart messages. The routines which encode and decode multipart messages in a way coherent with the overall design of the Intellitel Messaging Gateway exist but have not yet been integrated into the EAIF to SMTP gateway. The previously mentioned issues require the most work. The communication part requires much less work, as the state automaton of the gateway implementation is simple. The other limitations are not as major.

Because the gateway is unidirectional, there is no coding functionality in the WAP WSP-209 codec. The decoding functionality of course is implemented, as are some hooks for the coding parts, but the actual coding routines have not been implemented. However, it should be relatively easy to extend the functionality of the codec, as the coding routines are quite symmetrical to the implemented decoding routines.

The codec itself is not very robust, but this is mostly due to the structure of the WAP WSP-209 encoding itself. If the codec loses synchronization with the stream, even for one bit octet, the rest of the stream becomes garbage and the automaton scanning the stream is likely to decode trash or crash altogether. This was noticed with some totally broken files generated by the Nokia Mobile Internet Toolkit version 3.1 [Nok2002a]. The reason why the files broke was that the toolkit had default values for some headers, which were always displayed in the editor, but were never actually written into the file correctly. Opening the saved file caused the default header values to be seen in the editor again, giving a false impression that everything had been saved correctly. A workaround for these kind of broken files was not implemented, as it appeared too complex to fix. However, a workaround was made for another bug, which was caused by the toolkit putting a zero-length header name with an empty value into the stream.

The address rewriting functionality is not powerful enough to do mappings on an individual basis. This limitation can possibly be circumvented, or at least diminished, by using an external mail server with the gateway and doing more detailed address mapping there. For further development, implementing a dedicated database for the address mapping functionality is a more elegant solution.

The X-NOKIA-MMSC-CHARGED-PARTY header is not yet supported. This header was introduced in a later version of the EAIF specification [Nok2002], whereas the original EAIF to SMTP gateway implementation used the previous version of the specifications, which lacked this header. Adding support for this header is trivial.

The main reasons for these limitations were time constraints and other workload. The bugs in the MMS example messages from Nokia also contributed to the confusion during implementation. Also, originally the implementation was meant to be an experimental implementation. However, the prototype eventually became an actual gateway implementation.

6.3.5 Testing

Unit testing [Pre2000] of the implementation was done in two phases. The first phase was mainly focused on the WAP WSP-209 codec, before any networking code had been written. The test cases were various example messages from the Nokia Mobile Internet Toolkit 3.1. All test cases were valid messages, although some of the messages contained unintentional minor errors. For this version no deliberate attempts were made to test totally invalid messages or messages with values near or over the acceptable value ranges. After the codec was found to be working with the test messages, implementation of the networking code was started.

Another set of unit testing was conducted while the networking code was developed. This testing focused on getting the networking code state automaton to work correctly with the internal messages it has to handle. Additional bugs in the WAP WSP-209 codec were found and fixed at this phase. After the automaton was discovered to be working, an integration testing [Pre2000] phase was begun. During the integration testing phase a testing environment for the EAIF to SMTP gateway was built. This environment is described in figure 14.

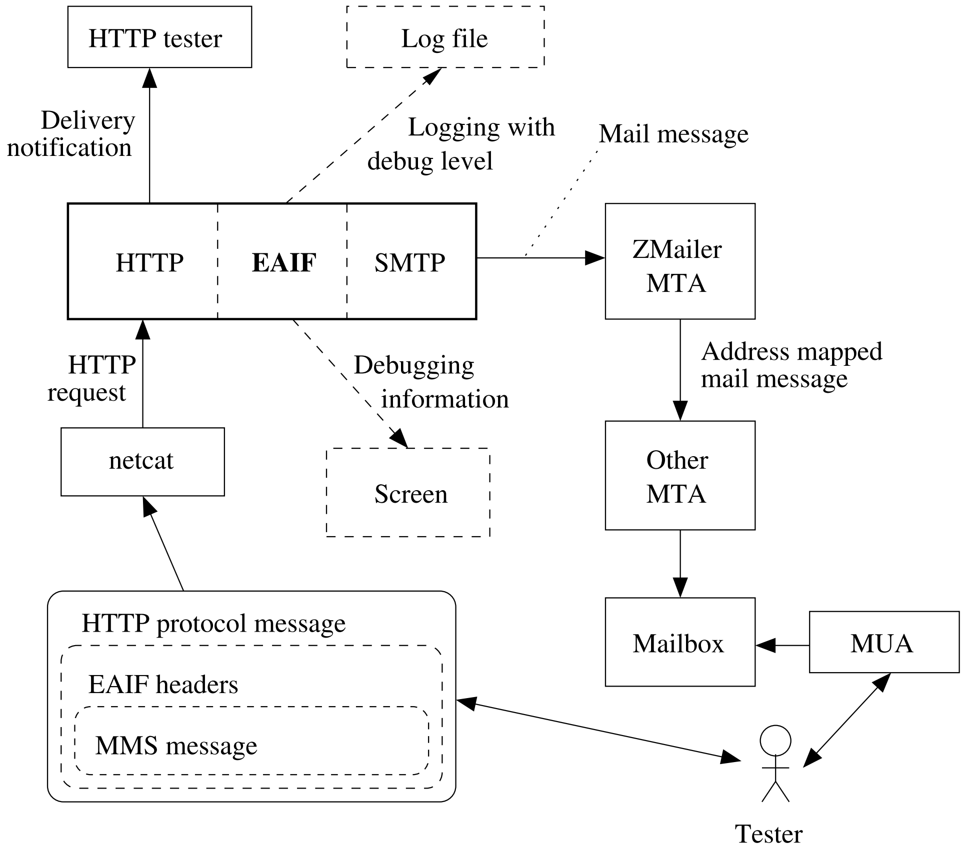


Figure 14: EAIF to SMTP gateway testing environment.

The MMS messages used during testing were the example messages from the Nokia Mobile Internet Toolkit 3.1 [Nok2002a]. Attempts to create new MMS messages with the toolkit were unsuccessful. After many debugging sessions, the created files were found to be broken. Despite many attempts, a correctly created file could not be created with the Nokia Mobile Internet Toolkit, and thus no more time was wasted in futile attempts. It was decided at this point to work with the example files only, as they were correctly created. Also, additional MMS message files from Nokia EAIF Emulator [Nok2002a] were used, as an inspection showed these files to be correctly created.

An EAIF protocol message was created by embedding an MMS message inside a file containing the EAIF headers. This file was in turn embedded inside an HTTP protocol message. The result was an EAIF protocol message; a file containing a valid HTTP request, with EAIF-specific extension headers and an MMS message inside. This file was sent using the NETCAT utility[12] to the HTTP server part of the EAIF to SMTP gateway. The request was handled by the HTTP part, and given to the EAIF part. The whole

[12] Available from @stake Research Labs, http://www.atstake.com/research/tools/.

gateway process was run with debugging logging level. The log file was written to a file which was constantly viewed. Also, the EAIF part wrote various debugging information to the screen, as it was run under the interactive ”development” mode of CVOPS.

After receiving and decoding the EAIF message the mail message was sent by the SMTP part to an external mail server, which was the ZMailer MTA [Aar2002]. The MTA did suitable address mappings from test addresses to real addresses so that the message could be delivered to an actual mailbox via the company-wide mail server. The message could then be viewed using a MUA, and each part of the message could be more closely inspected for correctness.

After integration testing, when the the EAIF to SMTP gateway was found to be working, the testing department assimilated the EAIF to SMTP gateway into their smoke testing [Pre2000] procedures. Smoke testing means that the software to be tested is integrated into the daily ”build” (a complete packaged product) and then tested as a part of that build. The testing environment described in figure 14 was also used during the smoke testing phase. While smoke testing, the gateway was run as one of the processes under the control of the management functions of the Intellitel Messaging Gateway. In this phase, some bugs and issues were still discovered, some of which were fixed but some which were left as outstanding issues to be fixed in future versions.

6.3.6 Outstanding Issues

Testing uncovered some issues, which could not be resolved. One of these was a memory leak, although it was not certain whether the memory leaking would stabilize at some point. An attempt to analyze the memory leak was conducted using Insure [Par2002] and Valgrind [Sew2002] but without success. This leaking of memory was especially visible when running a mass test with large, over 150 kilobyte messages. Thus it appears to be related to the message itself, or its handling, but the ultimate cause was not found, despite comprehensive examination.

Also, with extremely slow links and large messages some problems were noticed. These could be remedied somewhat by increasing the timeouts of various timers in the components along the flow of the message. For example, with over 170 kb messages and full debugging information both to log files and to the screen it takes minutes for the message to ooze through the system. This results in the SMTP part timeouting and killing the connection to the mail server uncleanly, which can have adverse effects. Without debugging information to the screen the message delivery is nearly instantaneous. However, this does not remove the problematic behaviour if the network connection is extremely slow and the endpoints are slow to answer.

A Nokia EAIF Emulator exists, but it was not used during testing for various reasons. Testing with the Nokia EAIF Emulator should be conducted at some point in the future. Also, a possibility to run tests in a live environment might be especially beneficial. This would require access to a Nokia MMSC and an MMS-capable mobile phone from which to send the message.

7 Conclusions

The MMS functional specification emphasized the usage of open standards. For example, the media formats used in multimedia messages have MIME-compatible media types. This property stems from the use of WAP in the MM1 interface. The HTTP and SMTP protocols already widely used in the Internet were also to be used within the MMS network architecture, especially at the MM3 interface. The requirements for the MM7 protocol interface were not specified conclusively, which resulted in the creation of non-interoperable MM7 protocol implementations among the largest manufacturers.

In the early versions of the MMS functional specification, the MM4 protocol was also ambiguously specified. This resulted in at least Nokia specifying and creating a proprietary MM4 implementation for their MMSC. Thus, also the MM4 interfaces became non-interoperable among the major manufacturers. This situation and the problems with interoperability suggested that a mediation approach similar to that used in the SMS networks would be the solution. The emphasis in such a mediator is in conversions from one protocol to another. Aside from protocol conversions, an SMS mediator also often provides added functionality such as billing ticket generation, management functionality, statistics, security functionality, load balancing etc. However, it is not enough to bring a mediation solution motivated by the problems in SMSCs directly to the MMS networks.

Although the first signs were different, now the manufacturers seem to be committed to interoperability. This means that eventually the MMSCs can communicate with each other using the MM4 interface. An interoperable MM4 interface is essential in order to do MMS roaming, and this fact adds pressure to make interoperable MM4 implementations. For VAS applications using the MM7 interface, the situation is still somewhat unclear. If the manufacturers want to bring interoperable MM7 interfaces to the market and succeed in it, there will be less need for a mediator solution doing MM7 protocol conversions from one interface to another. Moreover, if the manufacturers adopt SOAP as the MM7 interface implementation, there will be a single ”universal protocol” which can be used by any party to access an MMSC. This is a non-desireable scenario for mediation products focusing only on protocol conversions.

Even if the MM7 interfaces remain non-interoperable, a mediator solution relying on protocol conversion between the different MM7 interfaces will not necessarily be commercially attractive. This is because there are already many existing SMS gateways which can be made MMS-capable with little effort. Thus, many companies can make such MMS-capable mediators, which are in this case more or less clones of each other. More differentiation is needed for commercial attractiveness.

Straight protocol conversions do have a case at the MM3 interface. They can be utilized to artificially enlarge the set of external application protocols the MMSC can communicate with. However, the number of MM3 interface protocols is in practice limited, as very exotic or unpopular protocols tend to die away as time goes by. The less there are different and often needed MM3 protocols, the more there will be products which offer all of these protocols, eventually leading to a situation where the market is filled with practically similar products. Again, a mediator product has to offer something more than the others to stand out from the masses and bring justification to MMS mediation. In other words, the mediator has to add value to the mediation process itself. The basic idea of adding value to message mediation is to break free from the level of protocol conversions and go to a ”higher state of mediation”, where the message content and the domain of services is dealt with. This kind of value-adding functionality should be an integral part of the mediator.

Various ways to add value to MMS message mediation in multioperator environments were proposed. These ways included billing mediation with various billing scenarios, content transcoder functionality, management and service provider self-provisioning functionality, gathering of statistical data, MMS voting applications, positioning functionality, adding advertisements to a message, integration with HLR (for location services), integration with existing Internet servers and a location-based advertising scheme. Note that these are not MMS-specific as such, but could be implemented at least partly with SMS messages. However, with MMS the usability and practicality is greatly enhanced. For some services, such as voting applications, the message can be self-contained. The user can get all relevant information from the MMS message itself without having to rely on television or other media to obtain information which the message format is unable to present. For instance, an SMS message cannot properly present audiovisual media. Simple small monochromatic images and monophonic ring tones are useless for such a purpose. Thus, although possible up to some degree, some services are not very practical with SMS.

MMS messages can be larger than SMS messages and they integrate well to messaging formats used in the Internet and WWW. Programmability in the handsets allows parts of the service logic to be distributed to the 3G handsets, creating new kinds of applications. However, the information that the handsets process and display has to be obtained from somewhere. To fetch this data, the 3G handset can use WAP, GPRS (General Packet Radio Service), or some other method. Depending on the nature of the service, it can also be implemented entirely as an MMS VAS service.

Despite the programmable handsets, MMS VAS services have their place. This is because by installing a custom application to the handset, the user is committing to a long-term use of the service. Each application takes memory space from the handset, and memory space is a premium in a mobile appliance. Thus, the user is likely to install only such applications that she truly finds useful. For a user who only occasionally uses some service, using the service as an MMS VAS is easier, if there is such a choice available.

The implementation done for this thesis was a unidirectional EAIF to SMTP gateway. The gateway allows an MMS message to be sent into an existing e-mail address. The gateway can also act as a Nokia to Ericsson MM7 interface bridge with very little or no changes. Implementing a bidirectional gateway will require encoding functionality for the WAP WSP-209 codec and some changes to the protocol state automaton. The implementation does not only do protocol conversions. To add value to the message mediation, the implementation also deals with messages. The received binary encoded MMS message is transformed into a single part or multipart MIME mail message format, and the resulting message is modified to contain a small textual advertisement. Although being in text format in the implementation, the added advertisement could very well have been an image or a sound clip. Nevertheless, some aspects from the advertising campaign gateway scenario were utilized in the implementation. Although the implementation has limitations, it can be considered successful.

The implementation shows that IMG offers a solid base from which to build an MMS-capable product. However, there are recommendations for further IMG development. In the short run, the SMTP protocol interface of IMG should be developed further to eradicate any bugs still present and add missing functionality. In order to make the interface more robust, testing should focus on conformance with various real-life MUAs and MTAs. The EAIF to SMTP gateway should be developed further to add functionality required by bidirectional operation and the Nokia-Ericsson MM7 protocol conversion. Also, in the longer run, IMG needs a compliant SOAP interface. This will allow IMG to interface with future versions of major MMSCs. The biggest manufacturers are likely to start supporting SOAP on the MM7 interface now that the MM7 interface has been specified thoroughly enough. It will take some time for the implementations to catch up to the functional specification. This time can be used to implement the SOAP interface.

Also, routines to modify the MMS messages (i.e. the WAP WSP codecs and a SMIL parser/compiler) are definitely needed. A simple scripting language would bring more flexibility and more powerful ways to configure IMG in scenarios where message modification is done. For example, IMG could be instructed to add an image as the first part of the message, as the last part of the message, do any other conversions, etc. Configuring such functionality is cumbersome with the current line-oriented configuration file format.

IMG also needs a simple service-oriented API (Application Programming Interface) so that its capabilities can be used as a messaging ”black box” by invoking the API calls from a programming language. For example, simple request-response access to an HLR and sending or receiving of SMS or MMS messages would be good candidates for the API functionality. If the SOAP interface is implemented for MM7, the API can be realized using SOAP. The advantages of SOAP are that it is simple, easy to use and also architecture and language neutral.

If a mediator product is to succeed in the MMS world, it has to add value to the mediation process. This way there are justifications for the investments by the operator. Also, value-adding functionality in the mediator may enable new services to be created more easily. The value-adding functionality can also be a new service in itself. Whatever the MMS functionality of IMG will eventually be, some method of value-adding message mediation scheme should be implemented. The methods proposed in this thesis can be used as-is or as a starting point for innovation. At the very least, investing time and effort to build functionality to help in real-time message content modification is suggested. In conclusion, implementing even the most rudimentary value-adding functionality to the mediator can be beneficial, as it may provide the groundwork for a new kind of MMS killer application.

References

[Aar2002] Aarnio, Matti: ZMailer [WWW page], 2002 [Referenced 10.12.2002]. Available: `http://www.zmailer.org/`

[Apa2002] The Apache Software Foundation: Apache SOAP [WWW page], The Apache Software Consortium, 2002 [Referenced 19.11.2002]. Available: `http://xml.apache.org/soap/`

[Ban1998] Banzhaf, Wolfgang et al.: Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications, Morgan-Kaufmann, 1998. ISBN 1-55860-510-X.

[CMG2002] CMG, Comverse, Telefonieaktiebolaget LM Ericsson AB, Logica, Motorola, Nokia Corporation, Siemens Aktiengesellschaft, Sony Ericsson Mobile Communications AB: Conformance Document Regarding Interoperability of Multimedia Messaging Service, MMS Conformance Document, Version 2.0.0. [WWW document], CMG et al., 2002 [Referenced 28.10.2002]. Available: `http://forum.nokia.com/`

[Cel2001] Celtius Ltd., Users Guide for CVOPS 7.0, Celtius Ltd., 2001.

[Cro1982] Crocker, David H.: Standard for the Format of ARPA Internet Text Messages, Request for Comments: 822, RFC822, Department of Electrical Engineering, University of Delaware, 1982 [Referenced 14.11.2002]. Available: `ftp://ftp.rfc-editor.org/in-notes/rfc822.txt` [Note: obsoleted by RFC2822]

[ETS2000] European Telecommunications Standards Institute (ETSI): Technical Specification; Digital Cellular Telecommunications System (Phase 2+); Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM-ME) Interface; GSM 11.14 Version 8.3.0; Release 1999; ETSI TS 101 267 v8.3.0 (2000-08), ETSI, 2000. Available: `http://www.etsi.org/`

[ETS2001] European Telecommunications Standards Institute (ETSI): 3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional Description; Stage 2 (Release 5), 3GPP TS 23.140 V5.1.0 (2001-12), ETSI, 2001. Available: `http://www.3gpp.org/`

[ETS2001b] European Telecommunications Standards Institute (ETSI): Universal Mobile Telecommunications System (UMTS); Multimedia Messaging Service (MMS); Functional Description; Stage 2 (Release 4), 3GPP TS 23.140 version 4.2.0 Release 4 (2001-03), ETSI, 2001. Available: http://www.3gpp.org/

[ETS2002] European Telecommunications Standards Institute (ETSI): 3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional Description; Stage 2 (Release 5), 3GPP TS 23.140 V5.4.0 (2002-09), ETSI, 2002. Available: http://www.3gpp.org/

[ETS2002a] European Telecommunications Standards Institute (ETSI): 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Messaging Service (MMS); Media Formats and Codecs; Release 5, 3GPP TS 26.140 V5.1.0 (2002-06), ETSI, 2002. Available: http://www.3gpp.org/

[ETS2002b] European Telecommunications Standards Institute (ETSI): 3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional Description; Stage 2 (Release 5), 3GPP TS 23.140 V5.2.0 (2002-03), ETSI, 2002. Available: http://www.3gpp.org/

[ETS2002c] European Telecommunications Standards Institute (ETSI): 3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional Description; Stage 2 (Release 5), 3GPP TS 23.140 V5.3.0 (2002-06), ETSI, 2002. Available: http://www.3gpp.org/

[ETS2002d] European Telecommunications Standards Institute (ETSI): 3rd Generation Partnership Project; Technical Specification Group; Telecommunication Management; Charging Management; Charging Data Description for Application Services, Release 5, 3GPP TS 32.235 V5.0.0 (2002-09), ETSI, 2002. Available: http://www.3gpp.org/

[Eri2001] Ericsson: SMTP Interface Description for Content Providers, Ericsson, 2001. Nr. 1/155 19-1/HDB 104 02 Uen.

[FIC2002] Finnish Communications Regulations Authority: Regulation on Barring Categories in Telecommunications, FICORA 35 G/2002 M, Unofficial

Translation [WWW document], FICORA, 2002 [Referenced 29.12.2002]. Available: http://www.ficora.fi/englanti/document/FICORA35G2002M.pdf

[Gar1998] Garrard, Garry A.: Cellular Communications: Worldwide Market Development, Artech House, 1998. ISBN 0-89006-923-9.

[Int2002] Intellitel Communications Ltd: Product Description, Intellitel Messaging Gateway 2.2, Intellitel Communications Ltd, 2002.

[Int2002b] Intellitel Communications Ltd: User Manual, Intellitel Messaging Gateway 2.2, Intellitel Communications Ltd, 2002.

[Int2002c] Intellitel Communications Ltd: Service Provider Portal User Manual, Intellitel Messaging Gateway, Intellitel Communications Ltd, 2002.

[Kem2001] Kemppainen, Pasi: Network Intelligence Middleware: Enabling Network Convergence, White Paper, Intellitel Communications Ltd, 2001.

[Kle2001] Klensin, J. (editor): Simple Mail Transfer Protocol, Request for Comments: 2821, RFC2821, The Internet Society, 2001 [Referenced 14.11.2002]. Available: ftp://ftp.rfc-editor.org/in-notes/rfc2821.txt

[Mer2002] Merriam-Webster Inc., Merriam-Webster OnLine, Merriam-Webster's Collegiate Dictionary [WWW site], Merriam-Webster Inc., 2002 [Referenced 10.11.2002]. Available: http://www.m-w.com/

[Mic2002] Microsoft Corporation: Microsoft Software Developer Network (MSDN) Online, SOAP [WWW page], Microsoft Corporation, 2002 [Referenced 19.11.2002]. Available: http://msdn.microsoft.com/soap/

[Mob2000] Mobile Lifestreams Ltd.: Next Messaging: From SMS to EMS to MMS, Issue 1, Mobile Lifestreams Ltd., 2000, 276 p. ISBN 1-929105-30-4.

[Mob2002] Mobile Streams Ltd.: MMS Contracts Awarded [WWW page], Mobile Streams Ltd., 2002 [Referenced 25.10.2002]. Available: http://www.mobilemms.com/contract.asp

[Mot2000] Motorola Personal Networks Group: Wireless Application Development, Issues and Guidelines, Paging Systems Emphasis, Motorola Personal Networks Group, 2000.

[Nok2002] Nokia Networks Oy: Nokia MMSC, External Applications Developer's Guide, Reference Document [WWW document], Issue 3 en, DN00148759, Nokia Networks Oy, 2002 [Referenced 19.11.2002]. Available: http://forum.nokia.com/

[Nok2002a] Nokia: Nokia Knowledge Network, Nokia Forum [WWW page], Nokia Corporation, 2002 [Referenced 28.10.2002]. Available: http://forum.nokia.com/

[Nok2002b] Nokia Networks Oy: Nokia Artuse MMS Center Release MC1, Product Description, Introductory Document, Issue 1-0, Nokia Networks Oy, 2001. NB: This document was available from Nokia Forum, but seems now to have been removed.

[Nok2002c] Nokia: Nokia 6650 [WWW page], Nokia, 2002 [Referenced 02.11.2002]. Available: http://www.nokia.com/phones/6650/index.html

[Nur2002] Nurminen, Jani: Multimedia Messaging Service (MMS), Lappeenranta University of Technology, 2002. Final report for the Information Technology Project course. Available on request.

[Par2002] Parasoft: Insure++, Automatic Runtime Error Detection [WWW page], Parasoft, 2002 [Referenced 19.11.2002]. Available: http://www.parasoft.com/jsp/products/home.jsp?product=Insure

[Pre2000] Pressman, Roger S.; Ince, Darrel: Software Engineering, A Practitioner's Approach, European Adaptation, Fifth Edition, McGraw-Hill Publishing Company, 2000. ISBN 0-07-709677-0.

[Res2001] Resnick, P. (editor): Internet Message Format, Request for Comments: 2822, RFC2822, The Internet Society, 2001 [Referenced 14.11.2002]. Available: ftp://ftp.rfc-editor.org/in-notes/rfc2822.txt [Note: obsoletes RFC822]

[Sch2000] Schiller, Jochen: Mobile Communications, Addison-Wesley, 2000. ISBN 0-201-39836-2.

[Sew2002] Seward, Julian; Nethercote, Nick: Valgrind, An Open Source Memory Debugger for x86 Linux [WWW page], 2002 [Referenced 19.11.2002]. Available: http://developer.kde.org/~sewardj/

[Sir2001] Sirén, Pasi: Global Deployment of a Short Message Gateway, Master's Thesis, Lappeenranta University of Technology, Department of Information Technology, 2001.

[W3C2000] W3C: Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000 [WWW page], World Wide Web Consortium (W3C), 2000 [Referenced 18.11.2002]. Available: `http://www.w3.org/TR/SOAP/`

[W3C2001] W3C: Synchronized Multimedia Integration Language (SMIL 2.0), W3C Recommendation 07 August 2001 [WWW page], World Wide Web Consortium (W3C), 2001 [Referenced 22.12.2002]. Available: `http://www.w3.org/TR/smil20/`

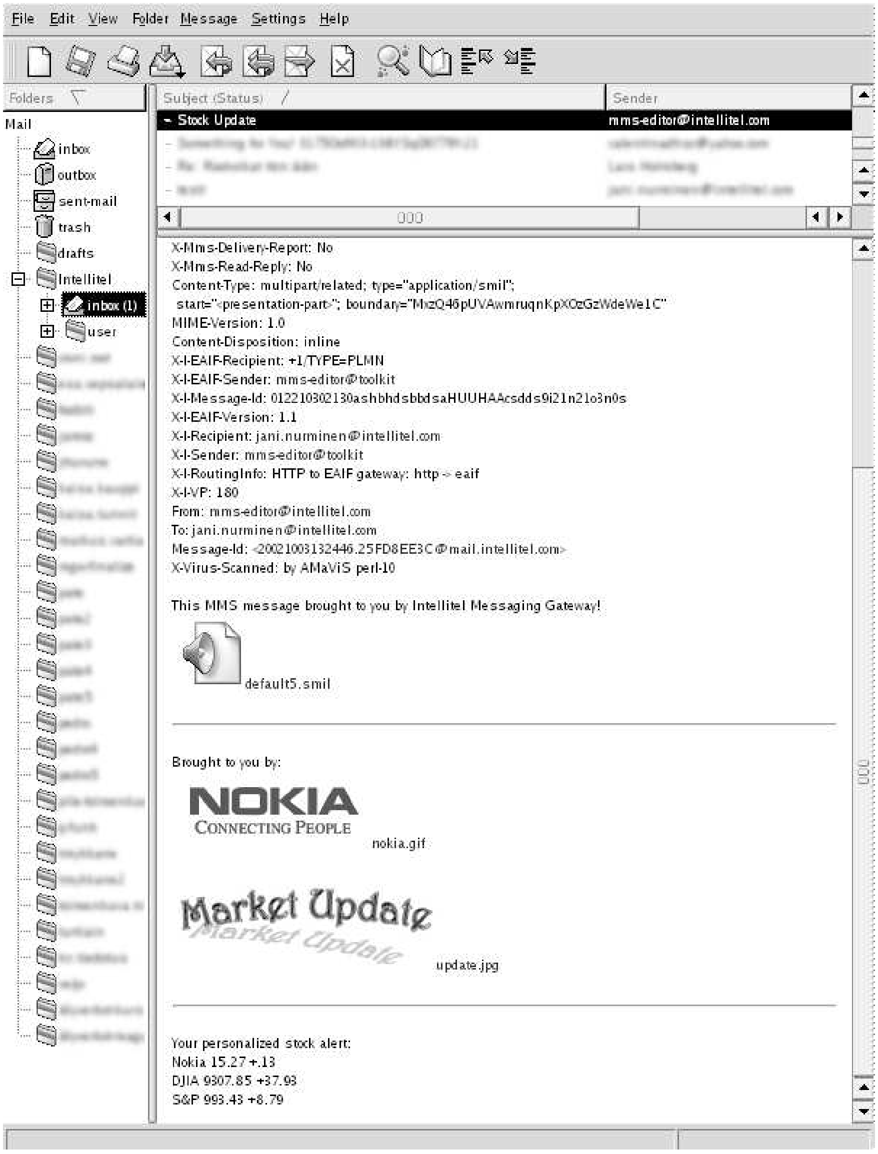
[WAP2001] Wireless Application Protocol Forum, Ltd.: Wireless Application Protocol, Wireless Session Protocol Specification, Approved Version 5-July-2001, WAP-230-WSP-20010705-a [WWW document], Wireless Application Protocol, Ltd., 2001 [Referenced 02.11.2002]. Available: `http://www.wapforum.org/`

[WAP2002] Wireless Application Protocol Forum, Ltd.: Wireless Application Protocol, MMS Encapsulation Protocol, Version 05-Jan-2002, WAP-209-MMSEncapsulation-20020105-a [WWW document], Wireless Application Protocol Ltd., 2002 [Referenced 02.11.2002]. Available: `http://www.wapforum.org/`

[Yeo1995] Yeong, Wengyik; Howes, Tim; Kille, Steve: Lightweight Directory Access Protocol, Request for Comments: 1777, RFC1777, The Internet Society, 1995 [Referenced 22.12.2002]. Available: `ftp://ftp.rfc-editor.org/in-notes/rfc1777.txt`

Appendix 1. Example MMS Message in a MUA

The Market Update MMS example message shown as a mail message inside a MUA. The SMIL part cannot be displayed in the MUA, hence the icon-only representation. The images, on the other hand, are shown as inline attachments. The short blurb added by the EAIF to SMTP gateway is shown as the first part of the message. Two images, a Nokia logo and the Market Update logo, are followed by the stock data. The parts are displayed in the order they appear in the original MMS message.



Appendix 2. Example MMS Message as Mail Message

The Market Update MMS example message as a mail message. The mail message is a multipart MIME message with the binary parts base64-encoded. The message is the same as in appendix 1.

```
From mms-editor@intellitel.com Fri Jun 28 23:20:44 2002
Return-Path: <mms-editor@intellitel.com>
X-Sieve: cmu-sieve 2.0
Received: from localhost (localhost [127.0.0.1])
        by mail.intellitel.com (Postfix) with ESMTP id B847BEE3D
        for <jani.nurminen@intellitel.com>; Thu,  3 Oct 2002 16:24:48 +0300 (EEST)
Received: from toolkit (xxxxx.intellitel.com [10.0.1.119])
        by mail.intellitel.com (Postfix) with SMTP id 25FD8EE3C
        for <jani.nurminen@intellitel.com>; Thu,  3 Oct 2002 16:24:46 +0300 (EEST)
X-Mms-Message-Type: m-retrieve-conf
X-Mms-Transaction-ID: 1234
X-Mms-MMS-Version: 1.0
Date: Fri, 28 Jun 2002 20:20:44 -0000
Subject: Stock Update
X-Mms-Message-Class: Personal
X-Mms-Priority: Low
X-Mms-Delivery-Report: No
X-Mms-Read-Reply: No
Content-Type: multipart/related;
  type="application/smil";
  start="<presentation-part>";
  boundary="MxzQ46pUVAwmruqnKpXOzGzWdeWe1C"
MIME-Version: 1.0
Content-Disposition: inline
X-I-EAIF-Recipient: +1/TYPE=PLMN
X-I-EAIF-Sender: mms-editor@toolkit
X-I-Message-Id: 012210302130ashbhdsbbdsaHUUHAAcsdds9i21n21o3n0s
X-I-EAIF-Version: 1.1
X-I-Recipient: jani.nurminen@intellitel.com
X-I-Sender: mms-editor@toolkit
X-I-RoutingInfo: HTTP to EAIF gateway: http -> eaif
X-I-VP: 180
From: mms-editor@intellitel.com
To: jani.nurminen@intellitel.com
Message-Id: <20021003132446.25FD8EE3C@mail.intellitel.com>
X-Virus-Scanned: by AMaViS perl-10
X-UID: 2630

--MxzQ46pUVAwmruqnKpXOzGzWdeWe1C
Content-Type: text/plain; charset=iso-8859-1
Content-Disposition: inline
Content-Transfer-Encoding: 8bit
Subject:

This MMS message brought to you by Intellitel Messaging Gateway!

--MxzQ46pUVAwmruqnKpXOzGzWdeWe1C
```

Content-Type: application/smil
Content-ID: <presentation-part>
X-Wap-Content-URI: http://orig.host/default5.smil
Content-Location: default5.smil
Date: Fri, 28 Jun 2002 20:19:35 -0000
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="default5.smil"
Subject:

DQo8c21pbCB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9TTUlMMjAvTGFuZ3VhZ2UiPg0K
IDxoZWFkPg0KICA8bGF5b3V0Pg0KICAgPHJvb3QtbGF5b3V0IHdpZHRoPSIzNTIiIGhlaWdodD0i
MTQ0IiAvPg0KICAgPHJlZ2lvbiBpZD0iSW1hZ2UiIHdpZHRoPSIxNzYiIGhlaWdodD0iMTQ0IiBs
ZWZ0PSIwIiB0b3A9IjEiIC8+DQogICA8cmVnaW9uIGlkPSJUZXh0IiB3aWR0aD0iMTc2IiBoZWln
aHQ9IjE0NCIgLz4NCiAgPC9sYXlvdXQ+DQogPC9oZWFkPg0KICA8Ym9keT4NCiAgIDxwYXI+DQog
ICAgIDxpbWcgc3JjPSJ1cGRhdGUuanBnIiByZWdpb249IkltYWdlIiAvPg0KICAgPC9wYXI+DQog
ICA8cGFyPg0KICAgICA8dGV4dCBzcmM9InF1b3Rlcy50eHQiIHJlZ2lvbj0iVGV4dCIgLz4NCiAg
IDwvcGFyPg0KICAgPHBhcj4NCiAgICAgPHRleHQgc3JjPSJhZC50eHQiIHJlZ2lvbj0iVGV4dCIg
Lz4NCiAgICAgPGltZyBzcmM9Im5va2lhLmdpZiIgcmVnaW9uPSJJbWFnZSIgLz4NCiAgIDwvcGFy
Pg0KIDwvYm9keT4NCjwvc21pbD4NCg==

--MxzQ46pUVAwmruqnKpXOzGzWdeWe1C
Content-Type: text/plain
Content-Location: ad.txt
Content-ID: CID1
X-Wap-Content-URI: http://orig.host/ad.txt
Date: Fri, 28 Jun 2002 20:19:35 -0000
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="ad.txt"
Subject:

QnJvdWdodCB0byB5b3UgYnk6

--MxzQ46pUVAwmruqnKpXOzGzWdeWe1C
Content-Type: image/gif
Content-Location: nokia.gif
Content-ID: CID2
X-Wap-Content-URI: http://orig.host/nokia.gif
Date: Fri, 28 Jun 2002 20:19:36 -0000
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="nokia.gif"
Subject:

R0lGODlhhwAyALP/AP///zBnzgEvy1GE1Jy/5Hih3c7g8e3x+7rS7d/q9vX4/RpIyQAAAAAAAAAA
AAAAACH/C0FET0JFOklSMS4wAt7tACH5BAAAAAAALAAAAACHADIAAAT/EMhJq7046827/2AojmRp
nmiqrmzrvnAsz3Rt33iu73zvZ4GgcHjIKIaBAgUxWDgF0KjUuRggKAUkAXNAChWA7PCaESOLK6la
EMgc1G1JYE2nxwFz6Z2SWNC3AH5SShgJdXsodVEDXHByipCLj1KMFYZrC2h4aoQXeXQGaZFkFW96
AJeRkQmblBUIdmATn1Cdr5ALoqsWplFtBKqqW7QClRLAa4itUbYUgoqkJ8GZpY7IiwfZ2tvZBWrD
asbea82TzBfXUc9QuSlqC+nq1afxoRu9UOCuAAN00RTEyuGDMiDVuURq+NExBmDgLzX/LhgUoE9S
v3esMAS0cFEKK2IC/2SZWGNu0ASHAOJFtDCxIsGO8jRspDCwmISJAsqJIClhXZSMKFVyaLlMUTuZ
nCqA1LRUWsKbdKgFhTj0W9FiIAOI9JR0ggFyfNYwHMFTwldMABRYo7qBKDFGMKOsLJkT4BoCBAro
JQAyI4myEsaFA7BWytywUlzaVAiWq0mzweyMfGqXXOEo9jS4HRxYsoWZgSKDKgFYggKfkuMt8OLF
p2KG8dhpoksotmgBR3dSRgzp4W1/V8faFpD5KiHUvwHploIBVu+Gv+mAeWvhbGPjYe5y2+5zwdYP
pSkMhxJnfDAy1HkhD1DEZ4Gayiw4fxwiPIW4pyQgQB5pAan06kXVxf83+BWXAWojbHfPdtl815AB
EBqAwIQUUmjAbKZx4yBN24WiADcNafiBgj+UaOKJKJqowIQYpljKAQok0KKL97ERgB86ubgfGzjS
yBFFUBkzowYbeuCgLB/+mM0cOQJw4Q9f5UYAIwSsBo9CoeynlwBX7GeFEzfGMcSUN64WQJUBGLBa
YP7hYSVD/RRhyABNBLFAKFU2wch+QnTRZlph3niYCP2UAwwBCgCDgClFpMloIFuwcYABjDghYxAF
JDBHAgWxoRAjagaSiQID7BHnEVx60w2EQL7BSDGG2ImFADIWMGgI/Sg3gRO7ZtJELo6WiccVbCTw
HjB+ATLHBDbxldP/Af7ROoFfCt1YlzcJIKDEjbMIQJgSYOaGrQFNjlCoBZ7KsQCnrhZggKauDrCF
H34YG9Jn3kpgk625JnFvH3dO0A8CfnlDhZsCe4vbjZqKixuvK0RZwWlHOdFFSrgd0AfGAXt6oar4

MsuIXqEFAEtGyyY8mzfHKDRLLrD25PAxRZLQzwBgaGorcU5SdLFCH7bTTyhsKEAqLHEgoGy+hI2s
hFrerjYd0wqtLACMfBmixFdKLCaq0XtdveJYKBjsRCUFVLGAEk2gd0DaykoYbi5RBnGFmm0Ck2YQ
ZrUBbQBNdFKlpN3aqQSfVQCAdBHQmsllP4b7KPnklFdu+eWYZ675D+acd+7556CHLvropJMeAQA7

--MxzQ46pUVAwmruqnKpXOzGzWdeWe1C
Content-Type: image/jpeg
Content-Location: update.jpg
Content-ID: CID3
X-Wap-Content-URI: http://orig.host/update.jpg
Date: Fri, 28 Jun 2002 20:19:36 -0000
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="update.jpg"
Subject:

/9j/4AAQSkZJRgABAQEAlgCWAAD/2wBDAAMCAgICAgMCAgIDAwMDBAYEBAQEBAgGBgUGCQgKCgkI
CQkKDA8MCgsOCwkJDRENDg8QEBEQCgwSExIQEw8QEBD/2wBDAQMDAwQDBAgEBAgQCwkLEBAQEBAQ
EBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBD/wAARCAA+ALMDASIA
AhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQA
AAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3
ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWm
p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEA
AwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSEx
BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElK
U1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3
uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD9U6K4
T4lfHD4Y/CN7aHx74mj06e8jMlvCYndpF3hOqgquWZVBYqCTgVH4h+Mnh7Sfgxc/G7TLeXUNFi0o
avEjsLZpICAckv8AdABz0OQPlDZALs9yPaRu431W539FeBfssftVWn7Ty+Ir3SfDMOmWOhtCiyC+
aV5Gd5lwUeKNgMQ7g2MHdjtmvc7vU9NsGRL/AFC2tmk+4JpVQt9MnmhpxdmFOpGrFThsy1RVLWNZ
0rw/pN3rut6hBZafYQtcXNzM4WOKNRlmJ9AK8h8Y/tk/s5eCYtObUviXpl1PqkTz29tZyCSYRKSG
eRSR5QBVgQ+05B44OBJvYcpxgrydj1bU/FXhnRdV03Q9X8QadZajrLvHp9pcXKRy3bqMsIkJy5AI
zj1HrWrX5t+PvHF9rn7c3wh8baf44TxD4d8VSi50hrfdHFFZiYotu0bcB42R8tjJaRie2PZvgF+1
38TfiD+0b6+pCQASSABySa+T/H37e9lY/x4ORm8S6rZyGHT9MZc70Mgjk
LsugeA
MqcFsGuO+If7XN98ev2fddl+G3hvxB4cRtttrOtks9rp0OJN6+ZGhkO9oxEPkTiVWJC5yowcti62
Kp0E+bdJuyT1t27n2vper6TrlkmpaJqdpqFpIWFpIWVLi1mWWNipKsAykg4IIPoQay/Gvjvwr8r8PdH/tz
xbqqWVq0ghj+RnklkIJ2IigsxwrE4HAUk4AJHzn/AME6PF+v+IfgPPosxbJNLVu3j+cAMi0AzTWU7hgggga37evi/wn4G+DsdsgD
JO0ArncGoLn5Xt5GcsRJ4V14JJ2v710r+el7fLU7HQf2v/2cvEnilfB2k/E7TJL9gu1NZtzwzyRYcrzhSvidp76rqMM1xBbNHKpa
OJGd23FdmAqMchucHFdb4J+NXw4vnj9hv9kXwsxXc8YaeNY8e+JbLz/td9+9fT7F1LJbRbshCynKndt6Dnw/8A4J7lLD4caV8eviN
eaFc6q2iHTdsVnCDM+Z7yOLFrmPkcv6P2ZfHWn/8FNvh
DqWhWd3Z+E9fu9YaEzahptqoYWDFR+8cK8XKXyVOBsdQQCMV758D758D/ANoD4aj/dfH/wAJSeLfLfA2ot5dq3
l3trc4Sa0bn/AFgBIAO1sHOMqw4Kk\sCXCUVdo2p4mlVlywkm9z0mmq6Mz\IrqWThgDyPrXyp8Y/wDg
oF8CvDdnr3hHwT8SLGTxZZ\wMlp01pNLZCYfwCVUYM5AIQgMpYrnIzXP/APBODxrnrirwr8SfiN4x
lu5urm68RSWU1zey8+XA0sqs7AbvLuU3N7AcAKActrdg68fakKmtb3/A+zaKwvDvkmi8Ke
MNE1p7c4mXT9QiuDH/vBGOPxEcUcs5wcAehPQE1Js
2krs6OiuU+F3xF0n4r+r+B9O8eaFZZZXdrY6hjt6m8jJuyjMpmU7cggkEEGuro2CLUldBRRRQM+
JP8Agq1p1hL8EvD+NvqqXp1hhL8EvD+qXVpHPJa+ILVCvzvzGN94AMB6mHVkpDK7nF9N5Y94BBBxhckA9s9gR8f/EDx58vM/g
aa50zVrw6e/2m4tLCc3JKJDAjc32v/wVKg839jdRMsnDWKv17rnpNLBNOL10r+UFfE9lgkA4Ijcde3y3Y/Gv
OJ/2GdA+NGleDvi74WOLT+MGmWXjDwZoMou2V7GxJaG1QLNGUVtzbSflJXn3Py3YPy7waUVzbHk4qnd
aXsviST9U7qz2XT1Mv8AZim0n9n79Wfhho9+83hHx7tsxrdzoU8kpdZY3hSa3bceWPliMnu4HLE
14j8d/hf408R/Rtb+Jeu3Gs3V3ZRabrDj+zbAxsbaCKIKVjKB4tx6kN7r8xHR/7ZWga
N8NvH3wXb4WTXAWXv4D3gp03r/wAP7e00eUnAM1vaGMRO0hOCytIWzQD/Dwa1qXxO+EP7Sdx+DzeId
K+IluZzRsLlt5Jru2MrZeJoYsuDvHyEkKyhXBYErTSu1uvxJqJq3s+n/DHT/sh
fELxP4P4p1zxF4VuR3Lk2MJVMd7yi4bUGIDk1ZNVhHQ2uYBDCrRzrNZH1Aeb+HzuNaJFl7G+MDJCEb4i6Fd
eA/GVpdNd6LPHa6bMrgPZ2kNww2IMYCuAcEjAwNu3bgV6V4W+Df7X3+xA/ab0n9qXxH4a2hg2hg
0uedbWKLT1URyhomLSpKUZ+WySrMDivYfgT+zH8Vvhn+0J4J4r+Mepa14fTT/ABBhczHUrGF5JHEWH
MYh+QAHeITlm6Bxgkg0m7X1/4cunTcTcnHcFcraTa1WvK1fW+uj+Z8wftz/Cz6Jc2

ds+maPfSENJFdyzysJN3Cq25856L6eva/sJ6n4d+FnxC8c/Bvxl4bv7H4uaoty0niPWb03FzqDhW
kEBdgMDgyAqBv25Ix5bP9KftHfspN+0H4t8N+JLrxwNKh8MqJLO2+wNKVucvmUOs0fVXA2kH7oII
yc5Xx6/Y98BfFXW/DPjvxL8RtR8O+INGtUsNQ1SDyIf7YjCAEzKQAHyMgrgAHaQVwKnmi4pM1VGr
CrKVPRaaaWatbTqrW8j4B8CfFPxqvw51H9jv4e6e2k+P/iV4ukttf1id1jWxsAiLcGR87tzEMSx4
IaRRknFfaPxBuPg/8Iv2ctc/Zp8BXct/dWPhmaee9t4PMhlliVZJ3nmTIEpRS5HO1dgyMoD554S/
4J2+JfFfxl8feNPivr19Z21/evLompaffKZxGJP3Spt5C+Xlm3nJdgfvbq+q/hn+zB8MPhn4f1PQ
ra1vNZOs2k1jfXOq3DTySW8w/exn7iuQC2OWIXJO1cNuHxdexNOGIVqSS5VpzPe3pbe9tW/PU81
/wCCad3Hcfsm+GbYXCvJaySJLHvDGF2VJCh9OXJAPYivO/8Agq1o+qN4A8CeLobd59M0LX1kvlAy
qjdFIC3sVhkH0J7V9Q/BH4AfD74AaNf6L4CtrpI9SuPtFxJczb2YjO1eAFAG5ucZJY5J4x2Pi7wj
4d8d+Hr3wr4r0uLUNLv08ueCTIzzkEEcqwIBDAgggEVHMlPmR0ujKeH9lLR2sefeHvjt8K7r4LWn
j2HxrotppseiiV/NukUwOkWDGUzu3BgVxjJxnivjT/gnTqllJ8Kfj9rlpIkdytzcQl2vJGn9o
AEofmAPoema+gtI/4J0/s9aRfRzRW+tyWUcvmixa/IhP+yVACheGwBSOxFZvin9lz4S/s5+CviL8
T/Cb6ok1xot/Ba2Uk5eOOS6XyxGv8blmZUUMTy2TljuqvdtaL3MW6/NGPVbhV9PGpViko369LPXY81/4JWWP
gQeBviLqt2mjnXda1mW7vhKUad9PK43Pu58rzDKP7tfN+h6xr/gjwl+0J448ByBHwQxN0I
WNbaacyxkHp/x72zxtjkeYT1YZ9G/Y6/Yx1b4qfC06vLv4pePPAqXV3PaapZ6evlwalauwchHJwQ
eQykEKQOARX3Qv7LHwntfgTefs/aXpUlt4fvICrzMwluWuOCtyzMPmkDKp5GMADAHFVKSjJmNKhO
tRg77J2fe6sv67niml/DX4R+Hv8Agn5NrEnhLRsrX3hX+2ZNQNGQtmOQfQmc5Gz1zX
gPw+j8N6D/wS61TS/EGuz6PJ4x1swW4ty3mXsqSQ5gJBzskS2ZGPPBxhiQD6dL/wT++PU/ga7+Ck
vx7jPw9dzJHZ+bODMFbesUkYAwjMBuAc4zuJkbOZPFH7CXxTvf2SdL+E48Sade+FfC+qjU9JtlGI
/J+QtA0hwHcyRiTJwMFkzyHBePfqPPONG8KfEf9mLw94Y+Ntn8NdS8Fpc3i
TW3iGzM8lm0DRIsEcqkHPmFGcs+1X85gAV5r7G/4KC+JP+E0/Zk8F+P5LqdT4nigtYEM0ggivLq
An7QdjKHeNt0IDhlPnHgda4745L+3N8Uv2f4Pg3qXwkFJvmT03MJKPsZ
wTlZxW915dfx1d7a+fsX/C8dQ/ZB/Z98J6PPodz4w8QFgsrP8xUs
xbaC20sKL/trfHj4TeMND8OftLfB/wAP6SfEcazW9t4f1M3NxbRsX+aTJI48sjBwCf4x0ryv9u74
pTfCT9ov4T/FzRrO01G2h0hQ/CK40jSrt/JIfczzRmSMPGuJ/x3f7OOi/Dvwfbv+1Z+0l8
P8X/ABM8UZu1itHN9HoduAG8qKBASjRKPmfaAn3n5nTi6MZcvZAXfbe/m
fcuiarFrujWOtLh39vHcpaMOoYLIh5VhnBB6GirqsHUOvRhkcYorE9E8/8AjJ8DvA/x
10ax8x8PePkvZ9Osbk3SQ20EwWOYh4lIa9kmaSOzDsX
kIY8Fup2rwvYZ4HtlJgdcU7u1iPZw5ue2vc+tj541HeJtSWFEsLKw05xJHq8tyA75/wBZLJ1f
cfTIAHGT1qvol74bt/Emp+Pm06M6F4/2iGbcenJ/Ldp+uiMhbG/2iGbcenJPXFfFR7Krae
R4pIHtYWjl4dCgKv9R3pFnznr+t67qVjc+Obv4l29uk6vLZ6NBq0omjBJCLth2gEDuVPTk88dXN4
z8W6V8Ot7nxjrSj7IkW1pkiJ4eQsD274B5HIwa9S/4Rxwv5flf8I3pez+79jjx+RXrl+2a9kmaSOzDsX
Ve674z8f3Nzd6ULXxLNGJNLS/by4bNAoIk27G3vsGRkADrwcV71rPgzw14hvYNQ1nTBdT2xBhZ5X
AQjoQoYDPvisbxP8M7PWdWfxNo+oy6XcrirH9rG6RQgGOIywXOMDnI46d6AOX0b4i3WhWdzpMmv2
2v3dpMfteqX9wtpZwB4TDEbI7qOPtLT5EtMiJKbQo6pVM/nPrqaE/Ep6ARvSCi8JjKIYoZBCLUZ
Ezy2IXIuB3ghvCIDeaNrbvG9AgfsfHmZcbDkcIBnV+3ELw9/mTBs
6j9DiBvGQz6g3/HA9CKKh/sz8BRcL51DRqk8XpTZgDahs32hTiXsG9zTMbUxSQjwSBTmmhkv/9pqH
l0dkmWx3w0rOv4x2a5F4rjSR5sxxUx3mTxwVlyFgmd/cLUoTqeJpJHzGQ+ZGtH5+63zAMnpE+e9VrE
mO/8EARwmcYgAE7r9nw+O2z8/4lXHLP4N9t4fUtp0Xdbb6b9xRwhJ6g5DV9lbl5Qn5aKK70HnpQQ
wkY8D17pJQzPyyLpEc6CpMvnhY7tIxPUmX67ysIXjjN25vezTkR3Hx5PyBsV1vN8AeqZeKQp7Fa/
AjqZjr6a+8uNaHQm1YoP1YTgD51eTMsQs99DxovQMYhDhpdd4DbTXrxxWiqHnLaAqYhHfGL3Bc8kSsYk
e4nwPS96vEqzZq8KvTnpBxXBmmfZlDdPE4b7qSM6nEk8D3W/p3OZ9qqj4T+b9cCvbpkqGfyUG5ozD
uw2Jl2ehHQEeU8cj3MzxB5KrGKBG8MRqH7uW+EH6F/K4Rrld6hO47eIVpBNXezG5wkxcZFdVKbJtDvl
VDCsGGPmwB0ZExTbEEDGGWyKNGBCk9QzHydWT1JQvvTR9vIjMJ8Ll+Oig4wCQvJ8N1gI7j7EPdjmS
7T6yYn5Y5ytBxBTYXbBwNGkk7c0SpKjlgKjrZVXQxHUxEDJzYMz9HH3LfJFjmBwGwzdmDZGBzbOE
sBp7nSL6vX04rtDj1Kx4wDTkZMffrPLqNkhGxG0Pv0MHkwgs9hAqI47JMnP6FYR+iXyQjBJTNeUX
6o5ttMKVd5KQljqBrwLEg3Pxa4FrGzaK8VlHhOoE2jVN6Hl1KbeTDxIXtCfVbhLjdUd/bZSjAbFa
wu8Qrbt7rw4aTeVOiVhC4o5TPr0fNjj7YqAAAk6GwX2+UvyRh4XjgXt0nyUJLLwrK3PnFwRhiJ10HC
1nXzDFQ6UlftWxy9Gn5QAiZbC8e/Yd+kNZE59+3lqo/y08Ug6tr2h2P1PXGvF9g3/S0pjO5hySYm
cmRPLKWPO5RBlxU3fwmNTE+NKWCDf2mGkS0gm6wpxqhqyyRQXnXmK5h0RlKeATwTBxJuR76CgYbG
Ni7ZXcqPNyvk7hGr3DtZYVnXyUg2k1kfzKjHSxMaB+SRcO5hzr2hzRBu7eI7BDjkI4ZKaiPvDJ0lY
3rB71Ce5xvEz0SJx3vXVdbmvhTa7rjvq0dqCRbXkFsMGfOc20gEwEP8zYHBAS0bK0cRaJVbPh7kq1N
IJQjMYBRs0Tie4nxkRlOiKyxuPZFNmTSJ6BIrxB+mhAHvkPNxHa6gDrWFd7VMaTLP9Wf3szyQ0g2sP
XrsR2NZGMqtq+zG9CP5f9Ntw

```
--MxzQ46pUVAwmruqnKpXOzGzWdeWe1C
Content-Type: text/plain
Content-Location: quotes.txt
Content-ID: CID4
X-Wap-Content-URI: http://orig.host/quotes.txt
Date: Fri, 28 Jun 2002 20:19:37 -0000
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="quotes.txt"
Subject:
```

WW91ciBwZXJzb25hbGl6ZWQgc3RvY2sgYWxlcnQ6DQpOb2tpYSAxNS4yNyArLjEzDQpESklBIDkz
MDcuODUgKzM3LjkzDQpTJlAgOTkzLjQzICs4Ljc5IA==

```
--MxzQ46pUVAwmruqnKpXOzGzWdeWe1C--
```

Appendix 3. SMIL Presentation Part of MMS Message

The Market Update message contains a SMIL presentation part. The part is shown below. In an MMS handset, the parts are shown in the order defined by the SMIL part. Notice, that this order differs from how the parts are settled sequentially in the original MMS message. The picture in Appendix 1 illustrates this property. For example, the stock quote part (quote.txt) appears last in the original message, whereas the SMIL defines that it must be displayed as the second part, immediately after the Market Update logo image.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
 <head>
  <layout>
   <root-layout width="352" height="144" />
   <region id="Image" width="176" height="144" left="0" top="1" />
   <region id="Text" width="176" height="144" />
  </layout>
 </head>
  <body>
   <par>
     <img src="update.jpg" region="Image" />
   </par>
   <par>
     <text src="quotes.txt" region="Text" />
   </par>
   <par>
     <text src="ad.txt" region="Text" />
     <img src="nokia.gif" region="Image" />
   </par>
 </body>
</smil>
```