Examensarbete BEE 2003-29

Design and implementation of the MMS portal

Henrik Adolfsson
Peter Svensson

This thesis is presented as a part of the Degree of Bachelor of Science in Electrical Engineering with emphasis on Telecommunications

Blekinge Institute of Technology

April 2006

Bachelor of Science in Electrical Engineering
Blekinge Institute of Technology
Department of Telecommunications and Signal Processing
Examiner: Dragos Ilie
Supervisor: Eduardo Ibacache and Oskar Bengtsson, Densitet

Abstract

MMS-enabled terminals on the market today are very complicated to use. It takes several steps to create a multi-slide MMS-message with images and text. This discourages users from using it. To increase usage of MMS, several companies provide web-based or stand-alone programs that allow users to create and send MMS-messages from a regular computer. However these editors have many limitations and are not user-friendly.

This thesis describes the design and implementation of a user-friendly web-based MMS-portal where users can create, edit and send MMS-messages. The portal is integrated into Densitet's system for development of mobile services.

Conclusions that can be draw from this work are that problems with MMS interoperability have mostly the poor standardization to blame. Different terminals support different types of images and sound formats, and to make the MMS-portal user-friendly, format conversions of uploaded content had to be implemented. Also the MMS-portal only supports basic MMS-functionality.

If the MMS-specification includes more audio and image formats and if the MMS-terminals are upgraded to handle these formats, sending MMS-messages will be easier and mobile messaging will continue to grow.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

Multimedia Messaging Services (MMS) is a technology for sending and receiving messages containing one or several images combined with text and sound. The primary use of these messages is to enhance messaging on mobile devices.

When MMS was introduced, both operators and content providers thought that the MMS market would explode just like the SMS market did some years ago. However due to difficulties with interoperability between different operators MMS-centers, MMSC, which is a server that handles the distribution of the MMS between operators, applications and end users, initially users could only send MMS to other users with the same operator. These issues have been or are being solved by the operators, but it has slowed down the overall growth of MMS-usage.

Another problem, which has made development of MMS-services difficult, is the fact that different mobile devices support different picture and sound formats in a MMS. This means that a MMS isn’t displayed the same way on different MMS-enabled mobile devices.
There are requirements that are set by the Open Mobile Alliance (OMA) for the rendering of MMS messages, but they are only the minimum requirements for the entire spectrum of features that an MMS-device must handle [1].

To stimulate the increase of MMS-usage, providers have made both web based and standalone programs for creating and sending MMS. However these MMS-editors are complex and not user friendly.

1.2 Background

Messaging on mobile devices started in 1991 with Short Message Service (SMS) in the GSM-standard. SMS uses free timeslots for sending a maximum of 160 characters of user data in a single message. This standard was easy and cheap for the operators to implement and after a few years SMS functionality was found on every cell phone on the market.

When the operators presented a way for third party content providers to make some money through Premium SMS, the number of commercial SMS services increased rapidly. The user who sends a Premium SMS is billed by the operator according to standard Premium SMS fares, set by the content provider. The content provider receives the SMS and can provide the user requested service or content.
Logos, ring tones, interactive television and voting are the most common services using Premium SMS.

When SMS was introduced, no one could imagine the enormous impact this service should have on the world. Today operators and content providers are making big money on SMS and they are hoping that MMS will continue to strengthen their positions on the market.

1.3 Concept

The MMS portal will be developed together with Densitet, a company founded at Blekinge Institute of Technology. Densitet develops mobile system applications and services. These products give companies and users access to information regardless of their location. Densitet is developing a tool, MObile SErvice System (MOSES), for development of mobile services.

The main goal of this thesis is to create prototypes for two components which will operate independently of each other: the MMS-editor for creating and editing MMS-messages and the Moses MMS-module which will handle all future MMS traffic generated to and from Moses. These two systems will enable an end-user to easily create, edit and send MMS-messages. Both systems should be cheap and simple to implement and run.

1.3.1 MMS-editor

The MMS-editor should enable a user to create, edit and send an MMS-message.

To make the editor easy-to-use, we have chosen to implement it as a Java Applet, simply because Densitet uses other Java Applets in MOSES. It runs in a web browser (such as Internet Explorer or Netscape Navigator) and users shouldn’t have to download or install any programs. If necessary (depending on browser version) the latest Java Run Time Environment for that specific browser needs to be downloaded and installed locally. From a developer’s point of view, Java is free to download and use and it does not require a special development tool. Densitet use Java as their standard development language.

Java applets have poor security and they are, by default, not allowed to perform advanced tasks such as connecting to other computers or reading and writing files on the local computer. To increase security, all web pages surrounding the applet will be written in PHP due to its well-tested session security and its compatibility with MySQL.

1.3.2 Moses MMS-module

The MOSES MMS-module will follow the MOSES code standard and therefore it will be implemented in Java. The MMS-module should also be fully scaleable to be able to communicate with different vendors MMSC. In this thesis we implemented compatibility with Ericsson’s MMSC used by Vodafone.

1.3.3 Database

MySQL will be used for storing user account info, pictures, sounds and MMS because it is a low-cost and well-known database and also because it is already implemented in MOSES.

1.4 Thesis methodology

This thesis was conducted during the spring of 2003 and was divided into three stages.

During the first stage, existing MMS-editors, both web-based and stand-alone, where briefly tested and evaluated. All advantages and disadvantages were noted and taken into consideration during the second stage when the MMS-editor was designed and implemented. During the third stage the Moses MMS-module, which communicated with Vodafone's MMSC, was designed and implemented. During this stage the MMS-Module were able to send MMS-messages to a simulated MMSC.

1.5 Technical description MMS

To understand all the parts of this thesis it is of great importance to have some knowledge of the basics of MMS-message structure, and how it is communicated between end-users.

1.5.1 Basic MMS structure

An MMS-message is more advanced than an SMS-message. It can contain text, multiple images and multiple audios and it shows the content in a specific order and time. It is built up by slides where each slide can contain image, audio and text. These slides resemble in a way a presentation program, for example Microsoft PowerPoint, but are much simpler in their layout.

1.5.2 Synchronized Multimedia Integration Language, SMIL

Synchronized Multimedia Integration Language, SMIL is used to describe the MMS-message in a standardized way in order to display it correctly on any MMS-enabled terminal.
It is an XML-based protocol specified by W3C to enable simple authoring of multimedia presentations including streaming audio and video together with text and images.
A SMIL file is created using HTML-like tags that defines the order and duration of each slide. It also controls the presentation of each multimedia object with respect to position, duration and other information.

1.5.3 Multipart Internet Mail Extension, MIME

The contents of the MMS and the SMIL file are wrapped together using Multipart Internet Mail Extension, MIME to create a package that can easily be sent trough networks. Ordinary e-mail uses this protocol to attach a file or an object to e-mail. By using MIME, messages can easily be sent between MMSC and mail servers using the Internet.

1.5.4 Simple Object Access Protocol, SOAP

The MMS in MIME format is wrapped into a SOAP message and sent to the MMSC using standard Hyper Text Transfer Protocol, HTTP. SOAP uses standard XML and is a protocol for sending messages in decentralized environments.

1.5.5 MM7

The MM7 protocol is specified by 3GPP and is the interface between the MMSC and the Value Added Service Provider, VASP. It shall be based on SOAP 1.1 and use HTTP transport layer [3].

1.6 Existing MMS editors

The MMS-editors evaluated before developing the MMS-editor did all have their drawbacks. None of them had all the functionality we wanted to have in our MMS-editor. The two tested MMS-editors where from Spray and Ericsson.

1.6.1 Spray

Spray has an online portal with a lot of sections, there among a “Message center” where visitors can create and send a MMS. The MMS-editor is a Flash application that looks good and it has a nice flow and is easy to use. The big drawback is that you can only use the material that Spray provides. You have no possibility to upload and use your own images. When this thesis was finished, Spray’s MMS-editor was no longer available on the Internet.

1.6.2 Ericsson

Ericsson has a desktop application that you have to install on your computer. The application is hard to use and you have to choose to which Ericsson-phone you want to send the message to, you can not make one general that fits all recipients. Another drawback is that the Ericsson MMS-editor is not an online tool and you have to send the MMS to your phone through a cable, Bluetooth or IR and then from the phone send the message to your recipient. On the positive side you can use your own material, or you have to since there is no or a few multimedia files provided.

2 System Description

2.1 Design overview

The MMS portal consists of four main components. These are:

- MMS-editor
- Main portal pages
- MMS-module in MOSES
- Database

The MMS-editor runs as a standalone Java applet on a client machine and it is accessible trough the portal pages. These pages are written in PHP and they handle authentication and authorization of users.
The MOSES MMS-module is a standalone Java application that communicates with the operators' MMS-Central, MMSC.
All information regarding users, content and MMS-send-history are stored in a database. Both the MMS-editor and the MMS-module communicate with the database and are therefore located on the same physical server machine as the applet source classes are located on to minimize the security issues that would otherwise occur. It is however possible to handle secure traffic between a Java applet and a database running on a different machine using standard Java Security classes, but that was not within the scope of this thesis. These security issues are further described in chapter 2.6.2.

The communications between the different components are described in figure 2-1 below.

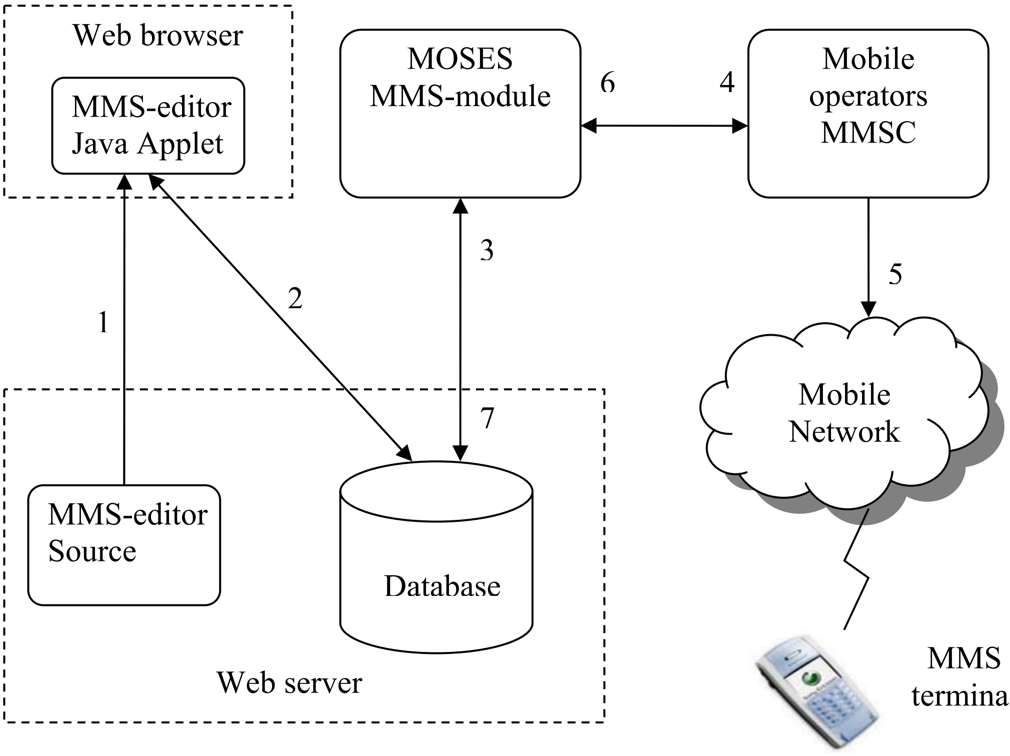
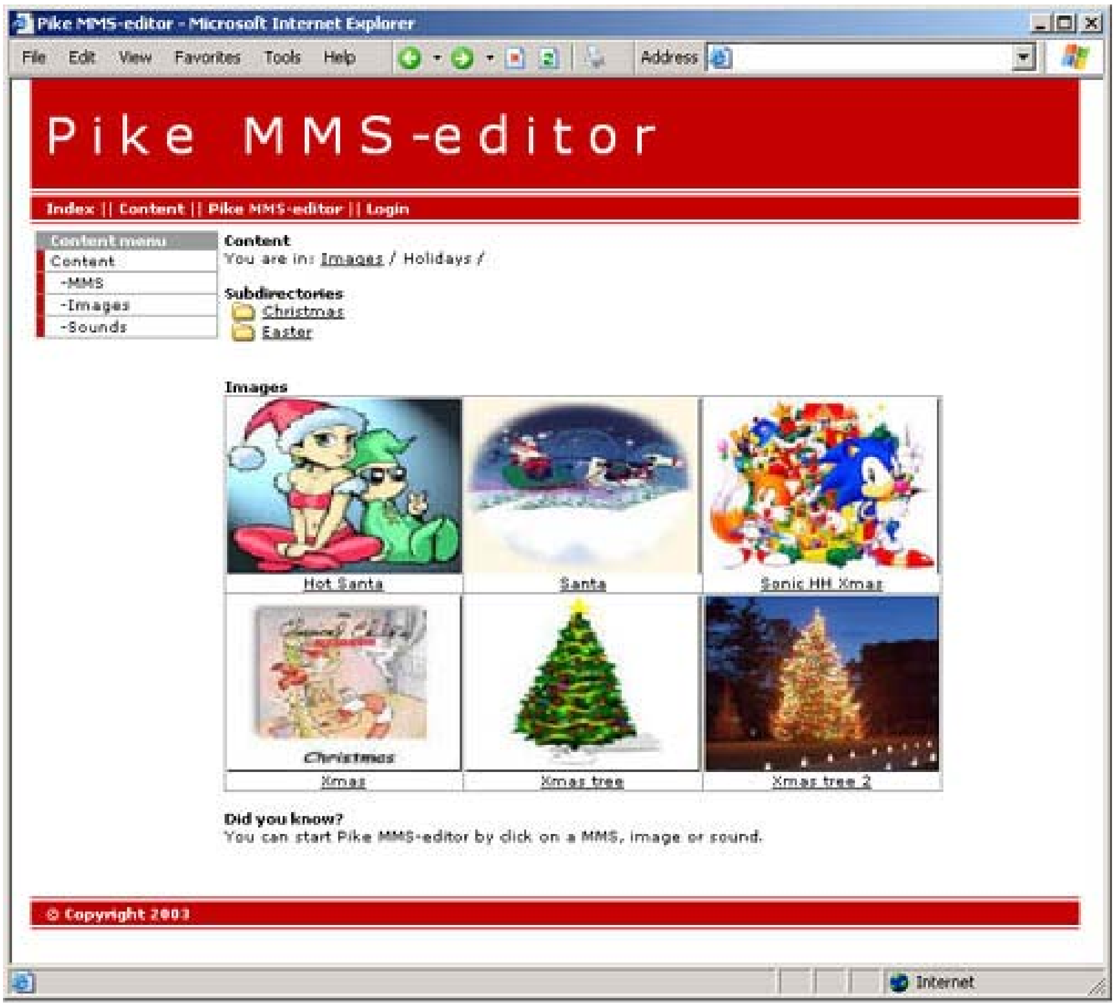


Figure 2-1: System and communication overview

1. The MMS-editor source is loaded from the web server to the client's web browser and is executed locally on the client.
2. The MMS-editor loads images and sounds from the database and MMS's ready to send are stored it the database.
3. When there is a new MMS to send in the database, the MMS-module load the MMS content, i.e. all the images, sounds and the written text, and generates a Synchronized Multimedia Integration Language (SMIL) file. The SMIL file is the structure of the MMS and it controls what is shown and for how long.
4. The SMIL file and all content are wrapped into a Simple Object Access Protocol (SOAP) message and sent to the operators MMSC.
5. The MMSC is now responsible for delivering the MMS to the subscriber.
6. For confirmation and tracing ability, the MMSC return a MMS ID to the MMS-module.
7. The MMS ID is stored in the database.

2.2 Portal

To show how the MMS-editor can be used in reality we have built a simple portal. The main purposes for the portal were to carry out login/logout functions and to show content to be used with the MMS-editor. Via the portal it is also possible to start the MMS-editor with a selected MMS, image or sound.


Pike MMS-editor - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Internet

Pike MMS-editor

Index || Content || Pike MMS-editor || Login

Content menu
- Content
- -MMS
- -Images
- -Sounds

Content

You are in: Images / Holidays /

Subdirectories

Christmas

Easter

Images

Hot Santa	Santa	Sonic HH Xmas
Xmas	Xmas tree	Xmas tree 2

Did you know?

You can start Pike MMS-editor by click on a MMS, image or sound.

© Copyright 2003

Figure 2-2: The portal displaying content in folder Images/Holidays.

2.2.1 Pages

Index – This is the first page that a user meets when visiting the portal. The user gets a short description of what this portal contains and is able to read the most recent news.
Content – This page allows the user to browse public content containing MMS-messages, images and sounds to use when creating own MMS. In logged-in mode the user can browse their private content.
MMS-editor – The page where the MMS-editor is displayed.
Version history – Displays the development of the MMS-editor.
Create user – On this page new users can sign up to get an account with possibilities to store MMS-messages, images and sounds.
Login – The page where registered users can login to use their private content.
Profile – Logged-in users can change their user-settings.
Logout – Logs out the user.

2.2.2 User rights

Users have different rights depending on if they have registered with the portal or not. Non-registered users will be able to create and send an MMS-message using standard images and audio, but they can’t upload own material to the editor. Registered users will be able to do all that the non-registered user can do and they will also have their own account where they can upload images and audio to use in their MMS-messages. They will also be able to delete their uploaded files through the portal.

2.3 MMS-editor

The MMS-editor is a Java Applet running in a single window containing four components: the slide editor section, the content viewer section, size and duration information section and send information section. They are further described below.

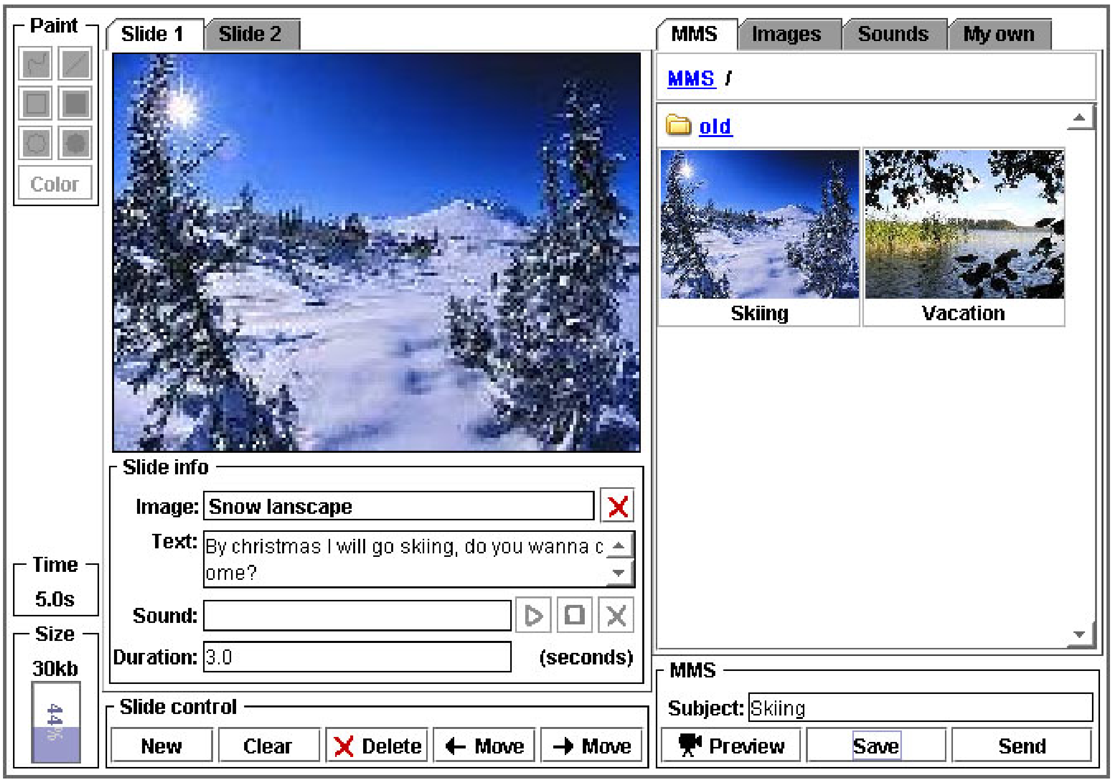


Image: Snow lanscape

Text: By christmas I will go skiing, do you wanna come?

Sound:

Duration: 3.0 (seconds)

Time 5.0s

Size 30kb

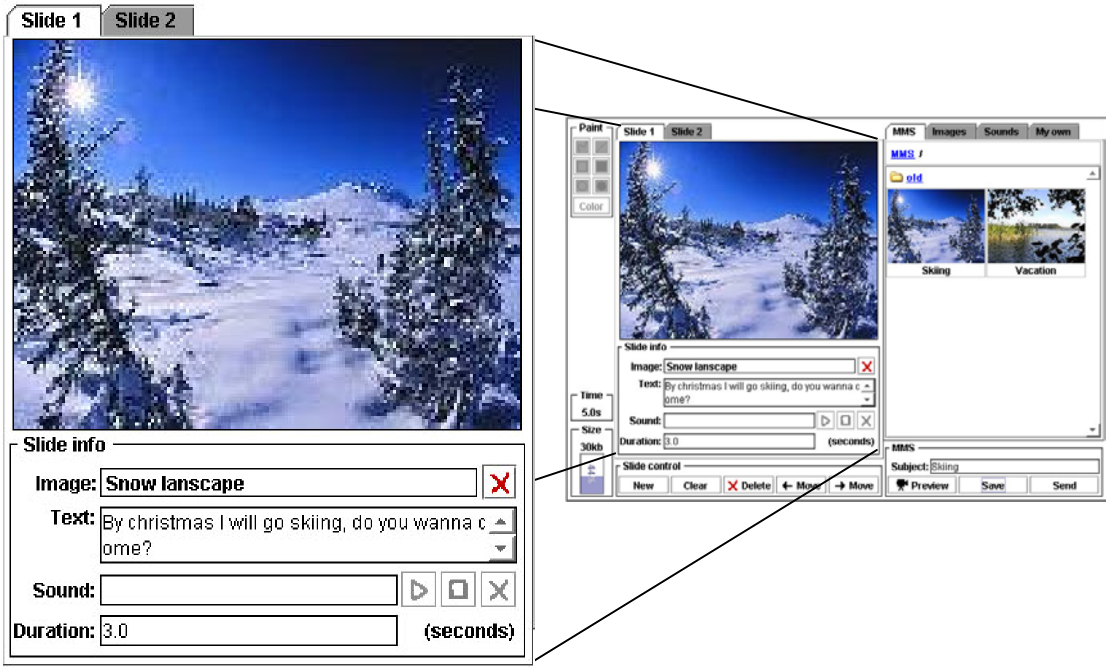
44%

Subject: Skiing

Figure 2-3: The MMS-editor

2.3.1 Slide editor

The slide editor section is the main workspace for the user. It consists of tabs at the top, the slide image in the middle, the slide information below the image and the slide control section at the bottom.



Slide info

Image: Snow lanscape

Text: By christmas I will go skiing, do you wanna c ome?

Sound:

Duration: 3.0 (seconds)

Figure 2-4: The slide editor section

2.3.1.1 Slide tabs

The slide tabs at the top enables the user to select which slide to work with. If a new slide is created it appears last in the tab list. By clicking on one of the tabs the selected slide will appear with the slide image and all slide information.

2.3.1.2 Slide image

The slide image shows the image of the selected slide. If no image has been assigned to the slide, the area is white. An image is added by clicking on a thumbnail image in the content viewer section.

2.3.1.3 Slide information

The information consists of four things: image name, slide text, slide sound and slide duration.

The image name is to identify the selected image. There is a delete button beside the image, which will delete the image from the selected slide.

The slide text is a text-box where the user can type a message that will appear together with the image on the selected slide in the MMS-message. Since the text-box only displays two lines, there are two scroll buttons for scrolling through long messages.

The slide sound shows the name of the selected audio. There are also play and stop buttons and a delete button, which will delete the audio from the selected slide. The play and stop buttons enables the user to preview the selected audio on that specific slide.

The duration window enables the user to determine how long the selected slide will show in the MMS-message. The total MMS duration in the size and duration section is updated instantaneously as the duration of any slide is changed.

2.3.1.4 Slide control

The slide control section enables the user to add, delete and modify the order of the slides. The following buttons are available: new, clear, delete, move left and move right.
New creates a new slide last in the MMS-message.
Clear removes all content on the selected slide, i.e. the image, audio and the text.
Delete removes the selected slide or all slides from the MMS-message via a selection box.
Move left moves the selected slide one step to the left. If pressed repeatedly the slide will continue to move to the left until it is the first slide in the MMS-message.
Move right functions as the move left button but moves the slide to the right instead.

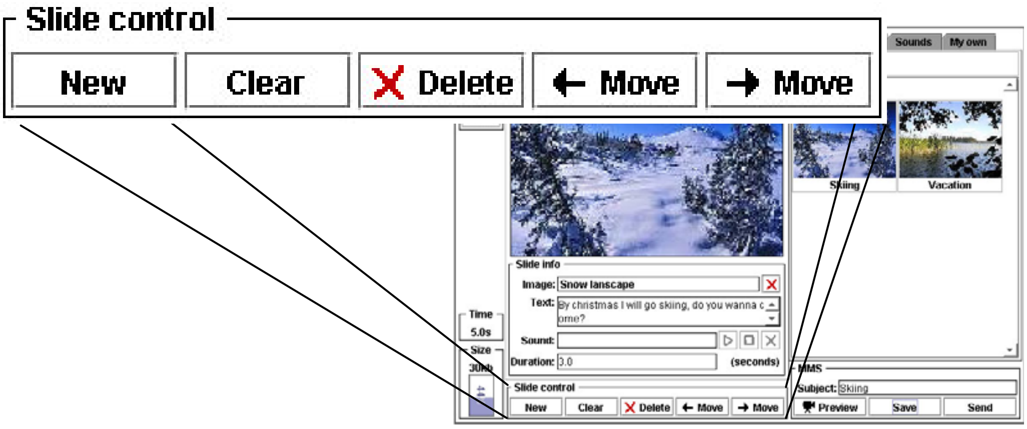


Image: Snow lanscape
Text: By christmas I will go skiing, do you wanna come?
Sound:
Duration: 3.0 (seconds)
Subject: Skiing

Figure 2-5: The slide control section

2.3.2 Content viewer

The content view section is where the user can browse through all content, i.e. images, audio and pre-made MMS-messages. If the user has logged-in, the option to upload own material is enabled.

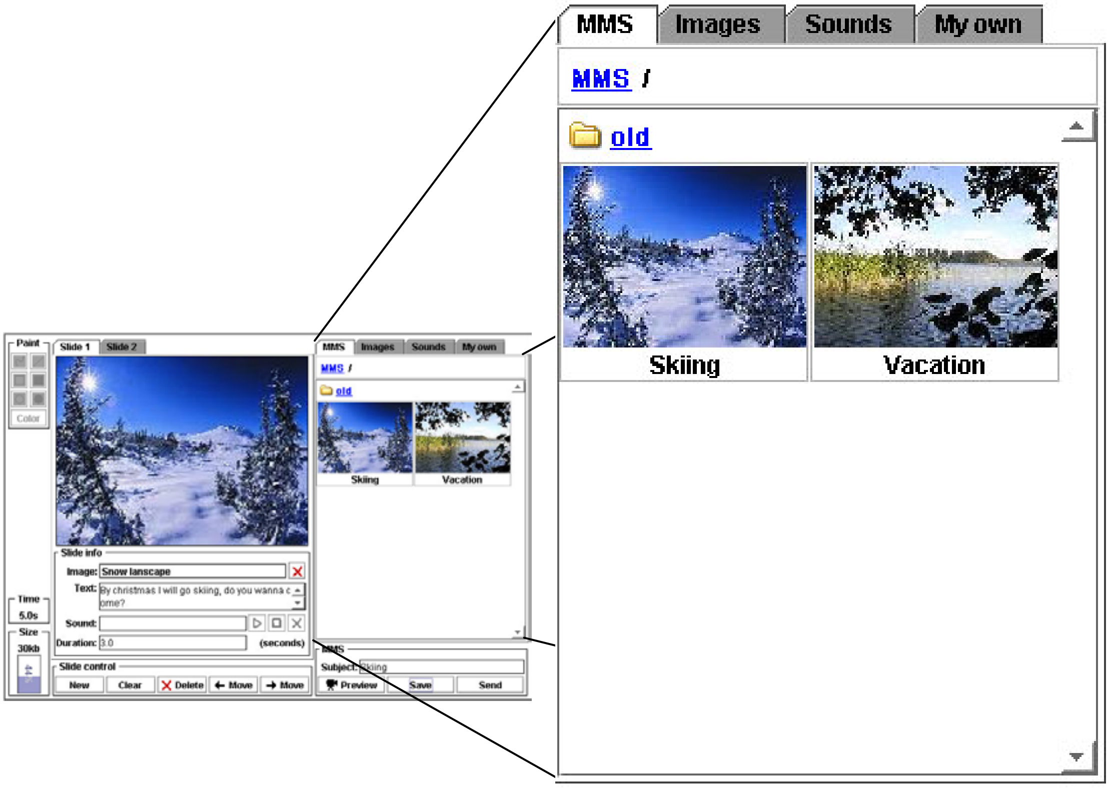


Image: Snow lanscape

Text: By christmas I will go skiing, do you wanna come?

Duration: 3.0 (seconds)

Subject: Skiing

Figure 2-6: The content viewer section showing pre-made MMS

2.3.2.1 Content viewer design

The top of the content viewer section has three tabs (four if the user has logged in). The four tabs are, from left to right; “MMS”, “images”, “audio” and “my own”. The fourth tab is only visible if the user has logged in, otherwise it is invisible.
When clicking on one of the tabs, the selected content will appear in the content window below the tabs.
The content viewer window is designed to resemble a Microsoft Windows file browser environment. Present directory is written at the top and all parent directories are clickable to enable fast browsing.
Subdirectories in the present directory have a small folder icon beside the name, and the name of the subdirectory is clickable. If subdirectories exist together with other content in the same directory, the subdirectories are displayed above the content.
Content types that are displayed are: images, audio and pre-saved MMS-messages.
Images are displayed as small thumbnails with the image name below (see Figure 2-7). The image name is limited to fifteen characters. Only the thumbnail is clickable.
Audio is displayed with a standard audio image with the audio name below (see Figure 2-7). Both images and audio thumbnails are displayed with a width of two thumbnails.
A scroll bar appears to the right if the content is too large to be displayed in one window.

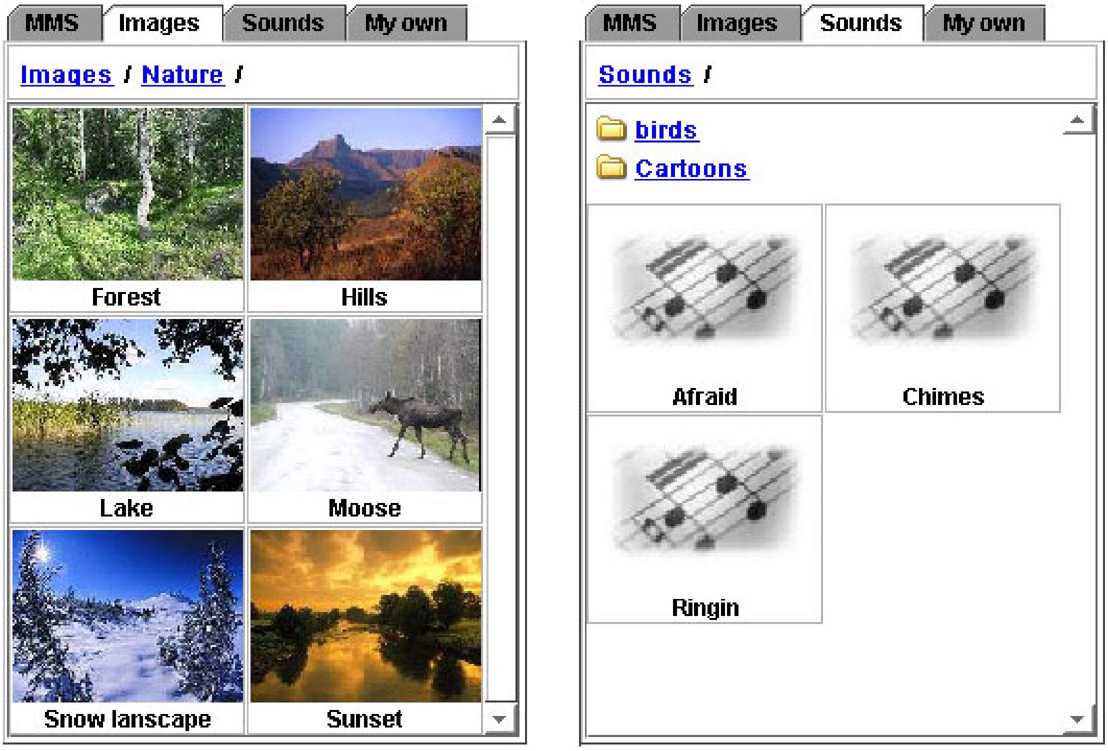


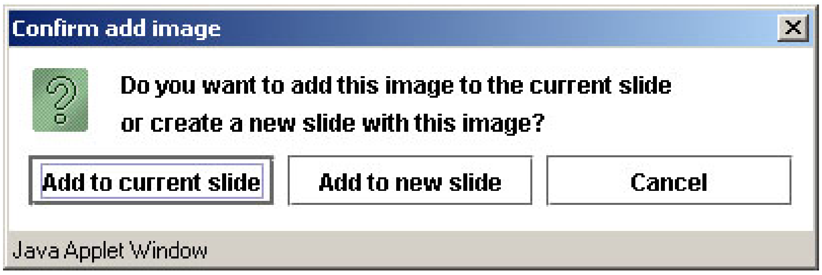
Figure 2-7: Content section showing images to the left and sounds to the right

2.3.2.2 Content viewer function

When the MMS-editor is started, the content viewer connects to the database and collects the active content in the root directory. By default it collects the root content of the MMS-, image- and audio section.

When a user clicks on a subdirectory, content viewer connects again to the database to download the relevant content. Each click by the user in the content viewer section generates a connection to the database and relevant content is downloaded. The advantage of this method is that the user always is displayed with the latest content present in the database.

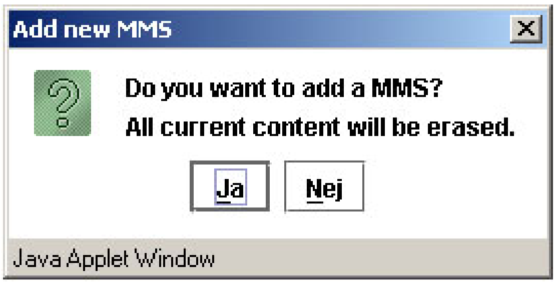
When clicking on a specific content item, such as an image or an audio, the content is loaded to the active slide in the slide editor. If there already is an image or a sound loaded in the active slide, a selection box is displayed (see Figure 2-8). The choices of the selection box are: add to current slide, add to new slide and cancel. Add to current slide replaces the current slide content with the new content. Add to new slide adds a slide after the last slide and adds the content to that slide. The cancel choice closes the selection box without action.


Confirm add image

Do you want to add this image to the current slide or create a new slide with this image?

Figure 2-8: Add image selection box

If a user clicks on a pre-made MMS when the slide section is not empty, a selection box appears and asks if the selected MMS shall be loaded, thus it will erase all other content in the slide section (see Figure 2-9).



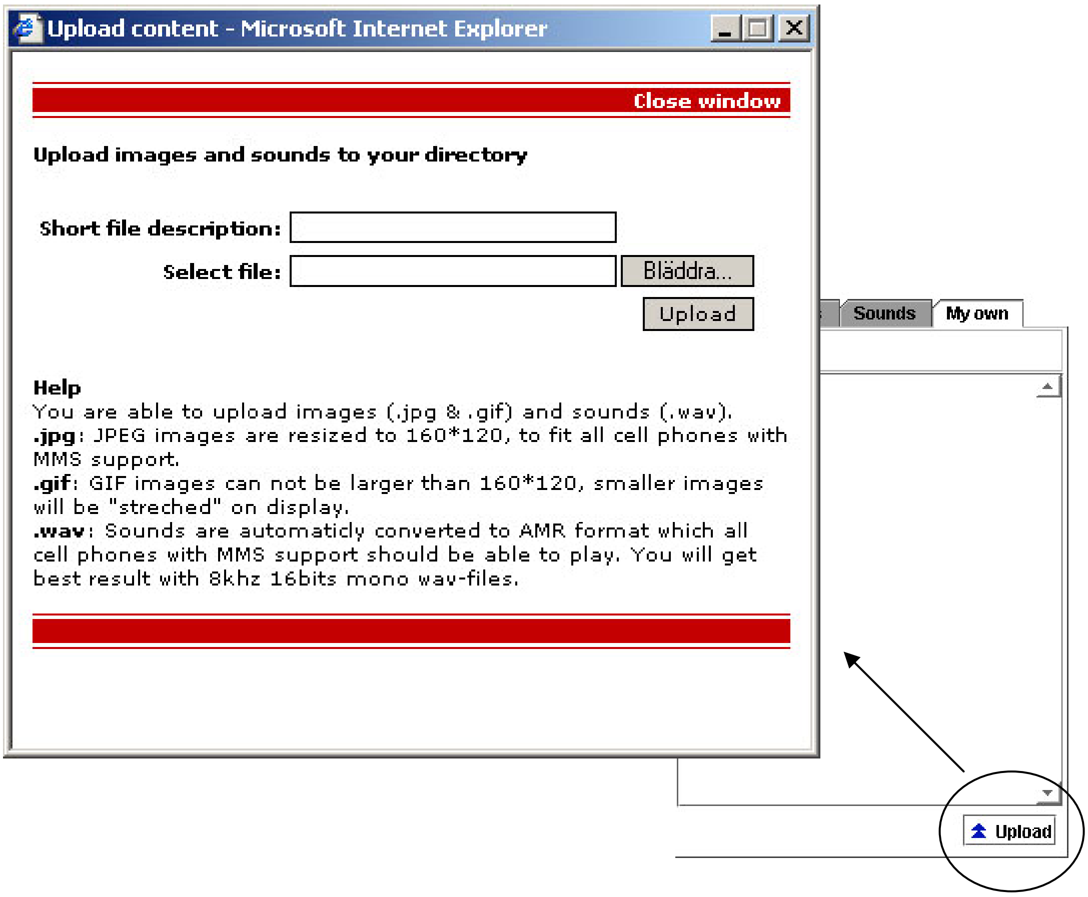
Add new MMS

Do you want to add a MMS?
All current content will be erased.

Figure 2-9: Add new MMS selection box

2.3.2.3 User upload function

The upload function is only enabled to users who are registered with the MMS-portal and also currently logged-in. An upload button appears under “my-own” tab and when a user clicks on it, a new browser window opens. This window contains a file description box, a file selection box and an upload button. To upload a file, the user has to write a short file description (maximum fifteen characters) which will be displayed below the thumbnail, and write the file name including full path in the select file box. The browse button to the right of the select file box opens a standard file browsing and selecting window, which enables the user to browse through the files on the user’s local computer to find and select the correct file to upload.
When the file is selected, a single click on the upload button uploads the file to the database and the boxes clears, which enables the user to upload a new file or close the window. All limitations to the files to be uploaded are displayed in the window.



Upload images and sounds to your directory

Short file description:

Select file:

Help
You are able to upload images (.jpg & .gif) and sounds (.wav).
.jpg: JPEG images are resized to 160*120, to fit all cell phones with MMS support.
.gif: GIF images can not be larger than 160*120, smaller images will be "streched" on display.
.wav: Sounds are automaticly converted to AMR format which all cell phones with MMS support should be able to play. You will get best result with 8khz 16bits mono wav-files.

Figure 2-10: The upload window

2.3.3 Size and duration information

In the bottom left corner is the size and duration section. The size symbol shows the user how much space the MMS will take. A maximum size of 30 kilobyte per MMS is set due to the limitations of MMS-terminals. The duration shown is the total duration of the MMS-message, i.e. all slides duration added together.

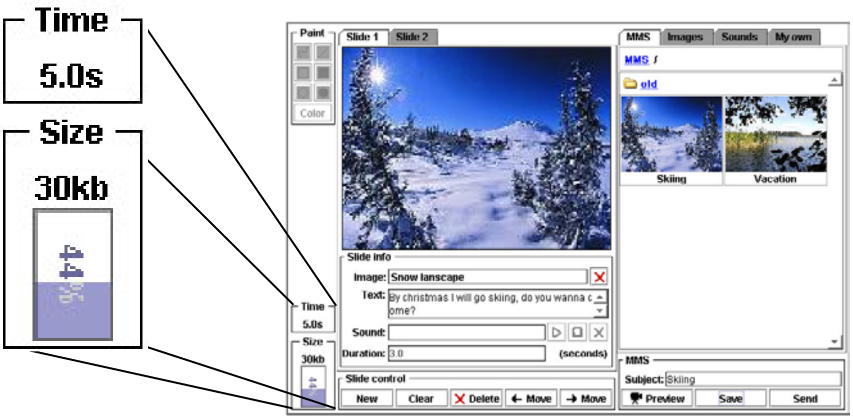


Image: Snow lanscape

Text: By christmas I will go skiing, do you wanna c ome?

Sound:

Duration: 3.0 (seconds)

MMS

Subject: Skiing

Figure 2-11: Size and duration information section

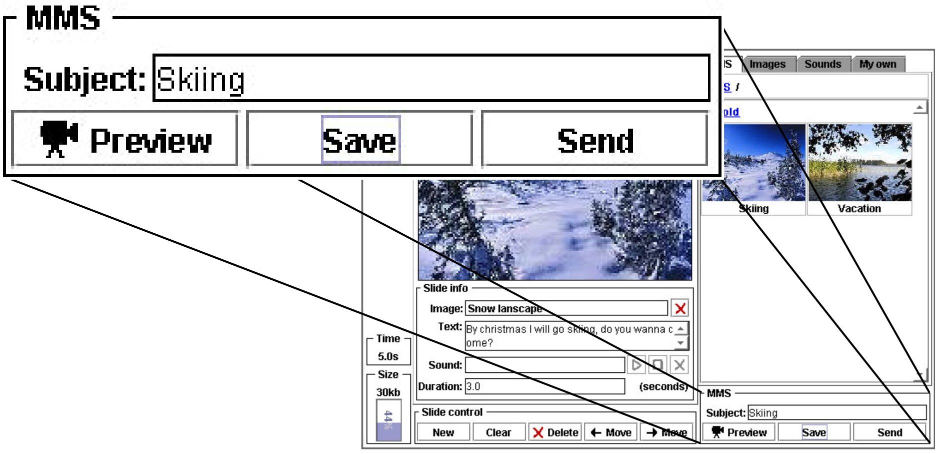
2.3.4 Send information

In the bottom right corner is the send information section. It contains three buttons: preview, save and send. It also has a subject row that enables the user to name the MMS-message. This is required if the MMS-message shall be saved.

Preview button opens a new window and plays the MMS as it would appear in an MMS-terminal, including images, audio, text and the correct duration. This preview can be cancelled at any point by pressing the cancel button. Closing the window is done by pressing the close button (see Figure 2-13).

Save button saves the MMS in the user's home directory, but only if the user has logged-in. Read more about user rights in 2.2.2. The saved MMS will be found under the "my own" tab in the MMS directory in the content viewer section.

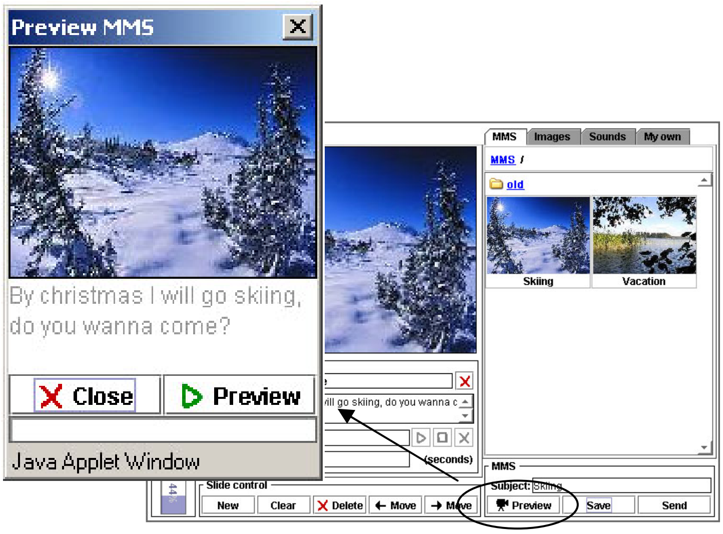
Send button opens a new window which contains two sections: recipient and delivery time. The recipient can be a cellular telephone number or an e-mail address. The delivery time enables the user to choose weather to send the MMS immediately or to schedule it for later delivery. The schedule function is limited to seven days. The send button sends the MMS and the cancel button closes the window without action.



MMS
Subject: Skiing

Slide info
Image: Snow lanscape
Text: By christmas I will go skiing, do you wanna come?
Sound:
Duration: 3.0 (seconds)
Time 5.0s
Size 30kb

Figure 2-12: MMS send information section

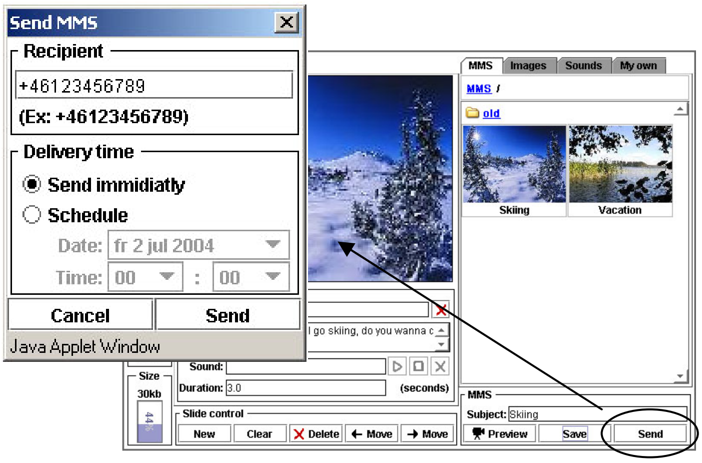


Preview MMS

By christmas I will go skiing, do you wanna come?

Java Applet Window

Figure 2-13: The preview window



Send MMS

Recipient

+46123456789

(Ex: +46123456789)

Delivery time

● Send immidiatly

○ Schedule

Date: fr 2 jul 2004

Time: 00 : 00

I go skiing, do you wanna c

Sound:

Duration: 3.0 (seconds)

Size 30kb 44%

MMS

Subject: Skiing

Figure 2-14: The send window

2.4 MOSES MMS-module

The prime task for the MMS-module is to send MMS-messages to the operators Multimedia Messaging Service Center, MMSC (or Mobile Messaging Center, MMC). It is fully scaleable in order to add interoperability with different MMSC manufacturers. In this thesis we implemented Ericsson's MMC API for communication with Vodafone's MMSC delivered by Ericsson. This API is further described in chapter 2.4.1. Other API's can easily be added afterwards.

Before the MMS-module could be implemented with MOSES its functionality had to be tested, and to do so it was developed as a stand-alone Java application.
The Java application connects to the MySQL database using standard MySQL Java connector libraries and can therefore run either on the same server as the database and Java Applet or on separate server. Due to required interconnection with Vodafone's MMSC and their firewall requirements, the application had to be implemented on the same server as MOSES runs on.

2.4.1 Ericsson MMC MM7 third-party API

The MMSC used by Vodafone is delivered by Ericsson and is referred to as the Ericsson MMC. For interconnection between the MOSES MMS-module and the MMC we used Ericsson MMC MM7 third-party API, supplied by Vodafone.
The Ericsson MMC provides the ability to send and receive multimedia messages from a third-party application, such as MOSES. The API provides a simplified programming interface to the MMC using MM7 functionality in a ready-to-use format. It handles connection and authentication with the MMC, it can send MMS to the MMC and it can receive MMS from the MMC. Another function is that it can compose and decompose a multimedia message in MM7 format. The MM7 protocol uses SOAP, XML and HTTP to interface to the MMC as the table below depicts.

Service provider Java application (MOSES)
Ericsson MMC Java API
MM7
SOAP
HTTP

Table 2-1: Description of Ericsson MMC MM7 third-party API

2.4.2 MMS-Module work flow

The application runs as an infinite loop regularly looking in the database for new MMS's to send. When a new MMS has been generated and queued to send by a user, it is highlighted in the database. The MMS-Module connects to the database and downloads text, images and sound as required and specified in the MMS. The text, images and sound are inputs to the Ericsson MM7 standard library which generate and send the MMS to Vodafone's MMSC. The MMSC returns a tracking ID and also whether the MMS was successfully sent or not. A more detailed description of the workflow is shown below.

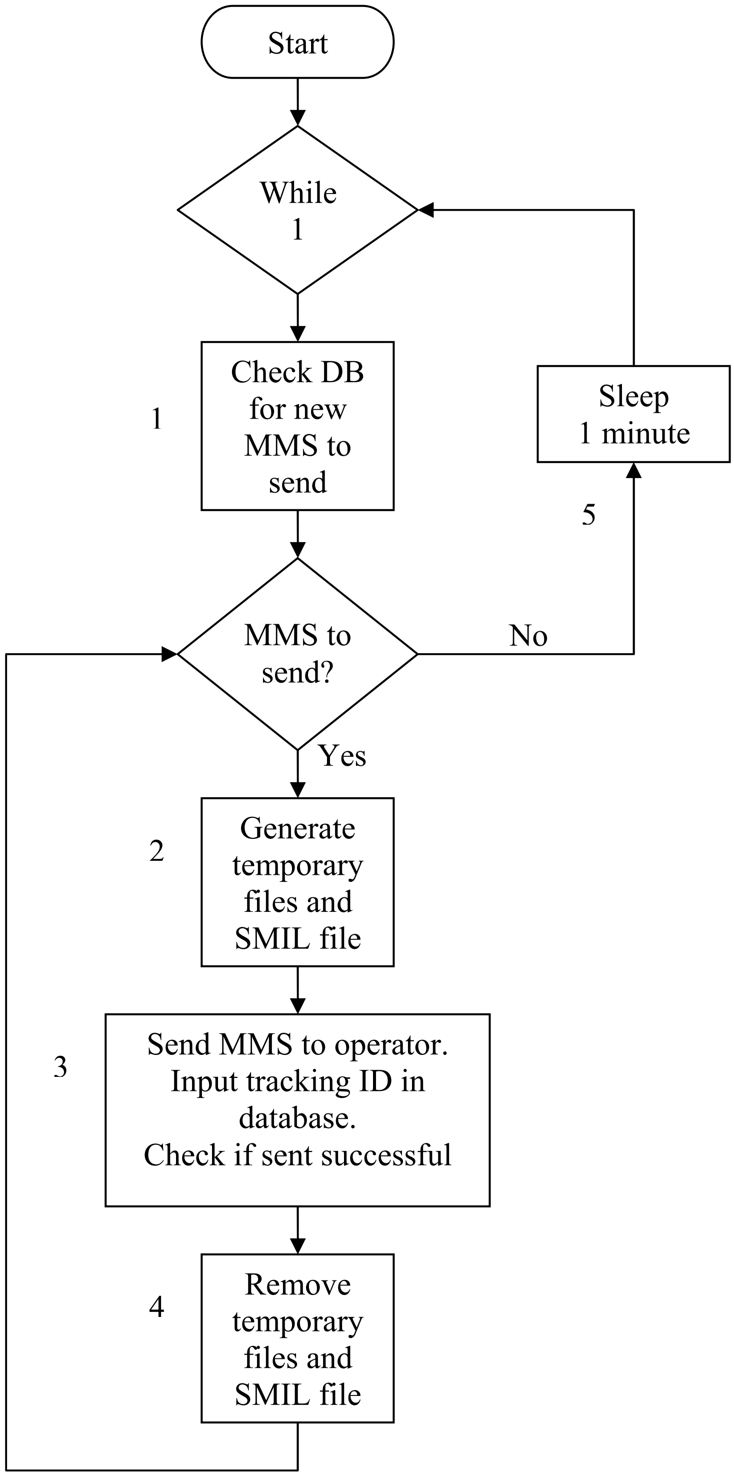

Start
While 1
1
Check DB for new MMS to send
Sleep 1 minute
5
MMS to send?
No
Yes
2
Generate temporary files and SMIL file
3
Send MMS to operator. Input tracking ID in database. Check if sent successful
4
Remove temporary files and SMIL file

Figure 2-15: MOSES MMS-module flow chart

1. The MMS Module connects to the database and checks a specific table (se more in chapter 2.5) for MMS's that are ready to be sent.
2. If there is a MMS to send the MMS Module checks which content it has and download the specified content. It can be written text, images and/or audio. It

also generates the required SMIL file and saves all content in temporary files in the correct format.

3. These temporary content files and the generated SMIL file are wrapped into a SOAP message and sent to Vodafone MMSC by using Ericsson standard MM7 API libraries. A message ID is received from the MMSC and it is stored in the database.
4. All the temporary content files are deleted and a new MMS is ready to be generated.
5. If there are no more MMS's in queue, the module sleeps for one minute before making a new check with the database.

2.5 Database

The MMS-editor and the portal use ten database tables to keep track of all data needed. These tables are described in detail below.

2.5.1 Tables

Table: folderinfo		
ID	unsigned bigint, auto increment, primary key	A unique identification number
ParentID	unsigned bigint	The owner of this folder
Name	varchar(15)	The name of the folder
PathName	varchar(100)	The full search path for the folder

The table *folderinfo* contain information about all folders where content is stored, each image, sound and MMS that is saved in the database has a reference to a folder. If the folder has same ParentID and ID it means it is a root folder.
Standard root folders are MMS (ID=1), Images (ID=2) and Sound (ID=3) where public content are stored, these folders are needed for the editor and the portal to work properly. Every user also has an own root folder that contains three folders, MMS, Images and sounds where private MMS and content are stored.
This table is used by the editor and the portal for deciding what content to show and for displaying of folder structure.

Table: images		
ID	unsigned bigint, auto increment, primary key	A unique identification number
ParentID	unsigned bigint	A reference to the folder that contain the image
Name	varchar(50)	Filename of the uploaded file
Data	longblob	Image data
Size	unsigned bigint	Image size in bytes
Description	varchar(15)	A short description of the image

The table *images* contain information and image data of the images stored in the database. Both the editor and the portal use this table.

Table: sounds		
ID	unsigned bigint, auto increment, primary key	A unique identification number
ParentID	unsigned bigint	A reference to the folder that contain the sound
Name	Varchar(50)	Filename of the uploaded file
SearchPath	Varchar(100)	Where the sound is stored on the server
Size	unsigned bigint	The size of the AMR converted sound file
Length	Float	Length of the sound
Description	Varchar(15)	A short description of the sound

The table *sounds* contain information about the sounds and references where they are located on the server. Both the editor and the portal use this table.
The filed *Length* is not in use due to technical constraints (see section 2.6.1).

Table: mmsinfo		
ID	unsigned bigint, auto increment, primary key	A unique identification number
ParentID	unsigned bigint	A reference to the folder that contain the MMS
Subject	varchar(50)	The subject of the MMS
Size	unsigned bigint	The size of the MMS in bytes

The table *mmsinfo* contain together with table *mmsslideinfo* information about stored MMS. Each MMS consist of one row in *mmsinfo* and one or several rows in *mmsslideinfo* depending on the number of slides.

Table: mmsslideinfo		
ID	unsigned bigint, auto increment, primary key	A unique identification number
MMSID	unsigned bigint	A reference to which MMS this slide belongs to
Image	unsigned bigint	Id of the image
Sound	unsigned bigint	Id of the sound
Text	text	The text of the slide
Duration	float	How long the slide should be shown in the MMS
Slidenbr	unsigned tinyint(3)	The position of the slide in the MMS

The table *mmslideinfo* contain together with table *mmsinfo* information about stored MMS. This table stores information about each slide of an MMS.

Table: sendmms		
ID	unsigned bigint, auto increment, primary key	A unique identification number
SenderID	unsigned bigint	The user id of the sender, 0 if not logged in
Subject	varchar(50)	The subject of the MMS
Recipient	varchar(50)	The recipient of the MMS

DeliveryTime	datetime	When the MMS should be delivered [yyyy-mm-dd hh:mm:ss]
Paid	unsigned tinyint(1)	Indicates if the user has paid the MMS
Sent	unsigned tinyint(1)	Indicates if the MMS is sent or if an error occurred
IP	varchar(15)	The IP of the sender
SubmitTime	datetime	When the MMS was submitted [yyyy-mm-dd hh:mm:ss]
SentMessageID	varchar(50)	The id received from the MMC when the MMS is sent ok.
Size	unsigned bigint	The size of the MMS in bytes

Table *sendmms* contain together with *sendmmsslides* information about MMS that has been or should be sent by the MMSHandler. Each MMS consist by one row in *sendmms* and one or several rows in *sendmmsslide* depending on the number of slides.

Table: sendmmsslides		
ID	unsigned bigint, auto increment, primary key	A unique identification number
MMSID	unsigned bigint	A reference to which MMS this slide belongs to
Image	unsigned bigint	Id of the image
Sound	unsigned bigint	Id of the sound
Text	text	The text of the slide
Duration	float	How long the slide should be shown in the MMS
Slidenbr	unsigned tinyint(3)	The position of the slide in the MMS

Table *sendmmsslides* contain together with *sendmms* information about MMS that has been or should be sent by the MMSHandler.

Table: users		
ID	unsigned bigint, auto increment, primary key	A unique identification number
UserName	varchar(12)	The user login name
Name	varchar(50)	Full name of the user
ShortName	varchar(15)	First name of the user
Password	varchar(16), encrypted	The password in encrypted format
HomeLibraryID	unsigned bigint	Where the user's MMS, images and sound folders are located

Table *users* contain information about users who have an account at the portal. Registered users that are logged in on the portal can upload images and sounds and store MMS in the editor.

Table: pikenews		
ID	unsigned bigint, auto increment, primary key	A unique identification number

TimeStamp	datetime	The date and time the news is published [yyyy-mm-dd hh:mm:ss]
HeadLine	varchar(50)	The heading of the news
PreText	text	The first part of the news that is shown on the first page
BodyText	text	The text that follows the PreText on a read more link

The table *pikenews* belongs to the portal and contain the portal news.

Table: pikehistory		
ID	unsigned bigint, auto increment, primary key	A unique identification number
Ver	float	Version number
Comment	varchar(150)	A comment what change made
Date	date	The date when the version were released, [yyyy-mm-dd]

The table *pikehistory* belongs to the portal and contain version history for the MMS-editor. Same history version can consist of one or several rows in this table.

2.5.2 Database relations

Figure 2-16 describes how the tables in the database are related to each other.

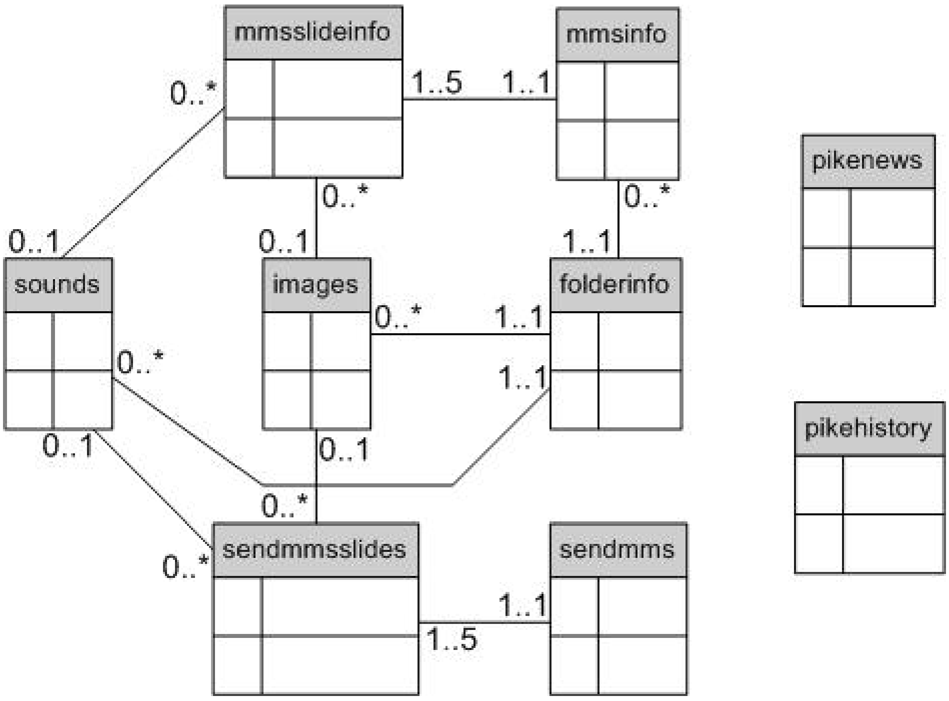


Figure 2-16: Database relations

2.6 Design issues

To limit the scope of this thesis, we have made some constraints to the design and the security of the MMS-editor.

2.6.1 Technical constraints

The MMS-enabled terminals that exist on the market are limited regarding graphical performance and audio capability. Therefore the MMS standard created by OMA includes only basic image and audio formats [1]. The supported formats specified in [1] are:

- Images
  - Base line JPEG with JFIF
  - GIF87a
  - GIF89a
  - WBMP

- Audio
  - Adapted Multi Rate, AMR (defined by 3GPP)

- Personal Information Management, PIM
  - vCard version 2.1
  - vCalender version 1.0

2.6.1.1 Image limitations

The maximum image resolution for which interoperability is guaranteed is 160x120 pixels. Most users are probably not familiar with how to resize image files, thus the need for automatic resizing of image files when uploaded.

Since the uploading script is written in PHP we used standard PHP-functions for resizing images. This works fine with standard JPEG images however GIF images cannot be resized in PHP due to GIF-files are read-only in PHP. GIF-files are read-only because they use a patented compression algorithm, which is not supported in PHP [5].

The MMS-editor drawing tool is disabled since this function was not within the scope of this thesis.

2.6.1.2 Audio limitations

The only supported audio, AMR, is the standard speech coding in GSM and thus all GSM mobile phones support this standard [1].

There are however several manufacturers that support other audio standard, such as MIDI, WAV, and iMelody, but since this MMS-editor should work on all MMS-terminals we have only implemented support for AMR. In the future when the MMS-standard includes other audio formats, they can easily be added to the MMS-editor.

Since AMR is not a standard in normal computer environments, 3GPP has developed a WAV-to-AMR-converter for Microsoft Windows platforms. It requires a standard WAV-file in linear PCM audio file in 16 bits mono and 8 kHz sampling frequency to produce a good quality AMR sound file. In this thesis we use this as a stand-alone program which we call upon from the PHP upload script. When the user uploads a WAV-file, it is automatically converted into an AMR sound file. Both the AMR-file and the WAV-file are saved in order to play the audio file in the Java applet, since the Java applet does not support AMR-standard audio. The WAV-file is played in the Java applet but the AMR-file is sent with the MMS.

2.6.1.3 Size limitations

The recommended maximum size of an MMS-message including all content is 30 kilobyte [1]. Therefore we have set the maximum size of a MMS to be created in the MMS-editor to 30 kilobyte. However with new MMS-enabled terminals this limit will increase and the maximum size variables can easily be changed in the MMS-editor. Due to the size limitation, the MMS-editor is limited to five individual slides.

2.6.2 Security issues

2.6.2.1 Portal security

The portal is written in PHP due to its well-tested session security and its compatibility with MySQL [4]. For the ability to have separate user accounts there must be a function to create accounts and a function to login to an account. When an account is created, the user data is stored in the database and required directories are created on the server. The user data concerning security are the user name and the password. The password is stored encrypted in the database. When a user login the given password is compared with the password stored in the database together with the specified user name. If they match the user is given a session id and granted access. The session expires if the user has been inactive for 60 minutes. It also expires when closing the web browser.

2.6.2.2 Java applet security

The MMS-editor runs as a Java applet in a web browser window on a client machine anywhere in the world. The Java applet communicates with the database located on the service provider server, in our case a Densitet server. By default Java applet does not allow connections to be made to any host, except the host that provided the class file. To enable the Java applet to connect to a database located on any server, we would have to implement Java applet security manager with signing and verification of the source code. Once the browser has verified that the applet is from a trusted source the platform can ease the security restrictions, enabling the Java applet to connect to a specific server, other than the host of the class files.
The Java applet enables signing and verification by using special numbers called public and private keys. Public keys and private keys come in pairs, and they play complementary roles. The private key is the electronic pen which signs a file as trusted. As its name implies, only the user knows the private key. Only the corresponding public key can verify a file signed with the private key. The public key together with its certificate is found in the Java applet source code file (JAR file).

If the database is located on the same host as the Java applet class files, there is no need for verification of the applet. To limit this thesis we implemented the database on the same server as the Java applet class files hence eliminating the need for applet verification.

2.6.2.3 Upload script

The possibility for users to upload their own content to the database requires the Java applet to read files on the local computer. By default Java applet is not allowed to read or write files on the local computer. However this can be enabled by using applet verification described in 2.6.2.2. Since applet verification is not implemented in this

thesis it is not possible to upload files within the Java applet. Furthermore there is also a need to check if the user is logged-in with the portal, which is a necessity for the usage of the upload function.
Using PHP solves both these problems. A PHP page is called from the Java applet together with the session ID which verifies that the user has logged-in. The standard file browsing, selecting and uploading functions in PHP enables easy uploading functionality for the user. It also enables automatic file preparing such as image resizing and audio conversion during upload.

The use of a PHP upload script also enables the possibility to use the upload function outside of the Java applet.

3 Conclusions

The aim of this thesis was to design and implement an MMS-editor within MOSES, in which a user should be able to create and send an MMS-message. The conclusion drawn by this work is that MMS can be a very powerful tool in future communication of information, but it is haltered by the dissimilarities in MMS-enabled terminals and the limitation of the basic MMS-standard specified by OMA.

When investigating MMS functionality, we found out that the MMS-enabled terminals on the market today are very complicated to use. It takes several steps to create a multi-slide MMS-message with images and text. This will in fact impede the usage of MMS and perhaps hider it from being the successor of SMS as it was predicted to be. Until the MMS-terminals become more user friendly, an alternative for MMS usage are computer based MMS-editors.

We briefly evaluated several MMS-editors on the market, both web-based and stand-alone. They were easier to use than the MMS-terminals but there were still some difficulties. They had several limitations, such as poor image type support, limited audio support and some did not have a capability for the user to upload own content.

When studying the MMS-standard specified by OMA, we realized that the reasons for the limitations are the MMS-terminals. They have large differences in terms of what images and audio they support. MMS created on a specific terminal may not be shown correctly on another terminal. To ensure complete compatibility OMA had to specify the MMS-standard with as basic features as possible. For example the only audio format specified in the standard is AMR, which is the standard speech coding in GSM.

When the MMS-specification include more audio and image formats and when MMS-terminals can handle these formats, MMS usage will grow even faster.

4 The future

4.1 Further development

The editor developed in this thesis contains basic functionality to handle MMS content, create and send MMS messages. The MMS portal and the editor can be further developed by adding more features to the existing functions.

- Painting functions
  By adding painting and image manipulation functions it is possible to change the images and let the user create new images of the uploaded and provided images.
- Improved security
  The MMS editor does not automatically logs out the user if the user is inactive for a certain time, which can lead to another user that comes to the same computer can use the first users account if the editor is active. This is solved by letting the editor check the validity of the session by accessing a PHP page and if the session has timed out, the applet should close.
- Send in MMS
  Instead of just being able to upload images in the web portal, the users should be able send in MMS messages to the system and the content is stored in the users account.
- Other content types
  The editor currently just support the content formats that are staked in the MMS Conformance document, but more and more cell phones support more advanced content types. Examples of content types that could be implemented are polyphonically sounds and video. The editor should also be able to convert between different content types, i.e. convert an mp3 sound file to AMR encoded file that can be listened to at any cell phone with MMS capabilities.
- Database servlet
  The current solution demands that the applet source and the database are located on the same machine due to security limitations in Java. By using a servlet that connects to the database the applet source and the database does not need to be on the same machine.
- Other improvements
  The editor currently does not show how long an added sound file is. The configuration of the MMS editor should also be moved outside the program code to enable faster and easier configuration of the editor.

The editor was built as a Java Applet and all the content where stored in a MySQL database. This solution may not ideal because when the applet loads the content from the database it takes a lot of time.

4.2 The future of MMS

When SMS was introduced in the GSM standard, which was on the market 1991 (in Sweden in 1992) [7], not many thought it would be such success as it is today. It is not many years ago when cell phones were not common at all, but more and more

people bought a cell phone and today almost everyone has a cell phone. By time people, mostly younger people understood how easy and convenient it is to communicate via SMS and the success was a fact.
The key to get a similarly progress for MMS is to continue to get users to change their cell phones to new cell phones that supports MMS. And also create interesting uses to the customers so it will be interesting to use the services. A problem here can be that users not changes cell phones as often as in the beginning of the cell phone era. If prices on cell phones with MMS capabilities keep falling and interesting, cheaper services are created, the process will be faster. Another thing manufactures of cell phones and providers of software for cell phones have to work on is the user friendliness of using MMS. Today it is complicated to create and send MMS; it has to be easier. I.e. when you take a photo with your cell phone's camera you should only have to click on one button to send it to a friend.

MMS seem to have a pretty bright future where MMS can be used for many commercial purposes besides the private where you just send images to your friends. MMS extends what SMS has started in marketing and notifying services.
Examples of services and uses of MMS, where images and sounds will extend the experience from a plain text message:

- Advertising – Advertising will be more efficient with images and sounds promoting the product.
- Description of a suspect – Polices "on the field" can get images and descriptions of suspected criminals to look for.
- Weather – Weather predictions will be more attractive with images showing weather maps and a short audio comment.
- News – Image, sound and text will provide great news to people's cell phones.
- Photo competition – People can send in their images to a central for storage and judging.
- Idol pictures – Get a photo from your sports or music idol, maybe a mobile newsletter to tell how the last performance went.
- Surveillance – A surveillance camera can send a MMS with images if something odd happens.

4.2.1 Mobile messaging future in numbers

IntroMobile is a large Korean company that is specialized in mobile solutions and analyses. IntroMobile has forecasted the following mobile trends. [6]

	2001	2002	2003	2004	2005	2006
MMS messages sent	0	115	1 099	2 938	7 926	12 450
Total messages sent	102 907	147 273	174 745	179 111	186 591	190 745
Percent of MMS	0%	0.08%	0.63%	1.64%	4.25%	6.53%

Table 4-1: Forecast of sent MMS-messages in millions

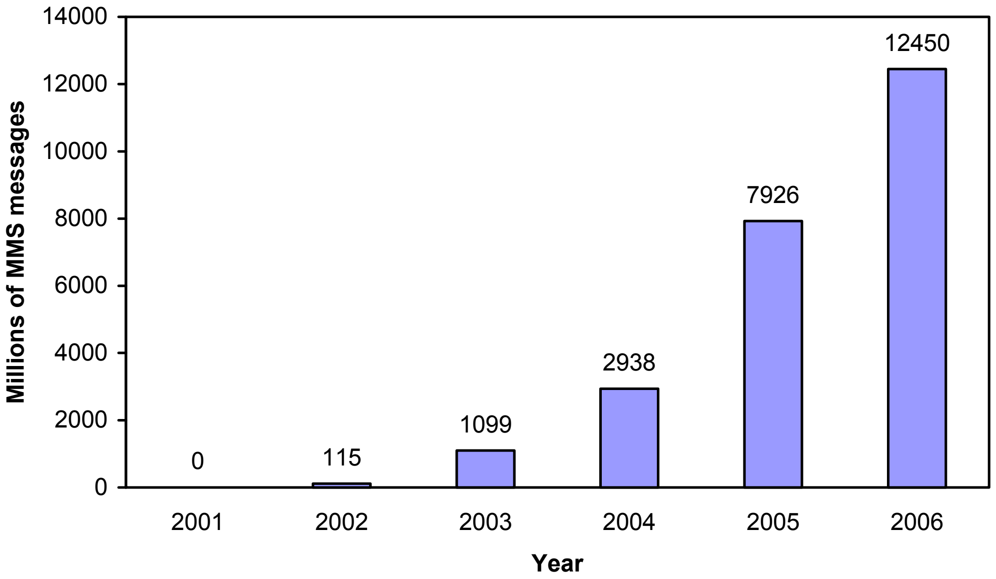


Figure 4-1: Forecast of sent MMS-messages in millions

5 References

[1] MMS Conformance Document, version 2.0.0. Open Mobile Alliance (2002) [www document] URL http://www.openmobilealliance.org

[2] Multimedia Messaging Service - Architecture Overview, version 1.1, Open Mobile Alliance (2002) [www document] URL http://www.openmobilealliance.org

[3] 3GPP TS 23.140 – Multimedia Messaging Service (MMS) – Functional description. (2002) [www document] URL http://www.3gpp.org

[4] Hypertext Pre-processor (PHP) – Documentation (2002) [www document] URL http://www.php.net

[5] GD 2.0.27 - GD Library documentation [www document] URL http://www.boutell.com/gd/manual2.0.27.html

[6] Mobile Messaging Forecast. IntroMobile (2002) [www document] URL http://www.intromobile.com

[7] GSM Global Networks on Air. GSMWorld [www document] http://www.gsmworld.com/news/statistics/networks_complete.shtml

6 Abbreviations

3GPP	Third Generation Partner Program
AMR	Adapted Multi Rate
GIF	Graphic Interchange Format
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transfer Protocol
JAR	Java Archive
JFIF	JPEG File Interchange Format
JPEG	Joint Photographic Experts Group
MIME	Multipurpose Internet Mail Extensions
MM	Multimedia Message
MMC	Mobile Messaging Center
MMS	Multimedia Messaging Services
MMSC	Multimedia Messaging Services Centre
MOSES	MObile SErvice System
OMA	Open Mobile Alliance
PCM	Pulse Code Modulation
PHP	Hypertext Pre-processor
SMIL	Synchronised Multimedia Integration Language
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
VAS	Value Added Service
VASP	Value Added Service Provider
W3C	WWW Consortium
WAV	Waveform Audio
WBMP	Wireless Bitmap
XML	Extensible Mark-up Language