



# Mobile Message Access Protocol (MMAP)

Version 1.0-Draft12

**DRAFT**

## **Mobile Message Access Protocol Specification v1.0 Draft12 17-Dec-2002**

©2001-2002 SMS Forum.

### **COPYRIGHT**

All rights reserved. This document or any part thereof may not, without the prior written consent of the SMS Forum, be copied, reprinted or reproduced in any material form including, but without prejudice to the foregoing and not by way of exception photocopying, transcribing, transmitting or storing in any medium or translating into any language, in any form or by any means, including but not limited to, electronic, mechanical, xerographic, optical, magnetic, digital or other methodology.

### **DISCLAIMER**

WHILST THE GREATEST CARE HAS BEEN TAKEN TO ENSURE THE ACCURACY OF THE INFORMATION AND DATA CONTAINED HEREIN, SMS FORUM DOES NOT WARRANT THE ACCURACY OR SUITABILITY OF SAME FOR ANY SPECIFIC USE. SMS FORUM EXPRESSLY DISCLAIMS ALL AND ANY LIABILITY TO ANY PERSON, WHETHER A PURCHASER OR OTHERWISE, IN RESPECT OF ANY CONSEQUENCES OF ANYTHING DONE OR OMITTED TO BE DONE BY ANY SUCH PERSON IN PARTIAL OR TOTAL RELIANCE UPON THE WHOLE OR ANY PART OF THE CONTENTS OF THIS PUBLICATION OR ANY DERIVATIVE THEREOF.

THE INFORMATION CONTAINED HEREIN IS BELIEVED TO BE ACCURATE AND RELIABLE. HOWEVER, SMS FORUM ACCEPTS NO RESPONSIBILITY FOR IT'S USE BY ANY MEANS OR IN ANY WAY WHATSOEVER. SMS FORUM SHALL NOT BE LIABLE FOR ANY EXPENSES, COSTS OR DAMAGE THAT MAY RESULT FROM THE USE OF THE INFORMATION CONTAINED HOWSOEVER ARISING IN THIS DOCUMENT OR ANY DERIVATIVE THEREOF.

NOTE 1: THE INFORMATION CONTAINED IN THE WITHIN DOCUMENT AND ANY DERIVATIVE THEREOF IS SUBJECT TO CHANGE WITHOUT NOTICE.

NOTE 2: THE CORPORATE NAME OF SMS FORUM IS NORTHGROVE LIMITED, COMPANY NUMBER 309113, REGISTERED OFFICE GARDNER HOUSE, WILTON PLACE, DUBLIN 2.

# Table Of Contents

1	Introduction .....	9
1.1	Scope Of This Document .....	11
1.2	Overview .....	11
1.2.1	Supported Cellular Technologies .....	11
1.3	Example .....	11
1.4	Glossary .....	15
1.5	References .....	16
2	MMAP Design Details .....	21
2.1	Overview .....	21
2.2	Background .....	21
2.3	SOAP design decisions .....	22
2.4	Operational modes .....	22
2.5	Session management .....	23
2.6	Security .....	24
3	SOAP Procedures .....	25
3.1	Session Management .....	25
3.1.1	Overview .....	25
3.1.1.1	Identification .....	25
3.1.1.2	Authentication .....	25
3.1.1.3	Capability presentation .....	25
3.1.2	Client Session .....	26
3.1.2.1	Example .....	26
3.1.3	Peer-to-peer Session .....	28
3.1.3.1	Example .....	29
3.1.4	Batch processing .....	33
3.1.4.1	Example .....	33
3.2	Request Handling .....	35
3.2.1	Immediate Mode .....	35
3.2.2	Client Session Mode .....	36
3.2.3	Peer-to-peer Session Mode .....	36
3.3	Error Handling .....	36
4	SOAP data elements .....	37
4.1	MMAP Header .....	37
4.1.1	ApplicationContext element .....	37
4.1.2	ServiceContext element .....	37
4.1.2.1	AccessControl element .....	38
4.1.3	Billing .....	38
4.1.3.1	Example .....	38
4.2	SOAP Body Elements .....	38
4.2.1	.....	38
4.2.2	BindRequest .....	38
4.2.2.1	Example .....	39
4.2.3	BindResponse .....	40
4.2.3.1	Example .....	40
4.2.4	BindBack Request .....	40
4.2.4.1	Example .....	41
4.2.5	Unbind Request .....	41
4.2.6	Unbind Response .....	41
4.2.7	Batch Request .....	41
4.2.7.1	Example .....	41
4.2.8	Batch Response .....	42
4.2.8.1	Example .....	42
4.2.9	EnquireLink Request .....	43
4.2.10	EnquireLink Response .....	43

4.3	Accessing MMAP .....	43
5	SMAP Design Principles.....	47
5.1	Summary of Approach.....	47
5.1.1	General Principles.....	47
5.1.2	Design Decisions .....	47
5.1.2.1	Modularisation .....	47
5.1.2.2	Language and character-set selection .....	48
5.1.3	XML Implementation choices .....	48
5.1.4	XML Naming Guidelines .....	49
6	SMAP Procedures .....	50
6.1	Message Submission .....	50
6.1.1	Submission to a single destination.....	50
6.1.2	Submission to multiple destinations.....	50
6.1.3	Submission using a distribution list.....	51
6.1.3.1	Example.....	51
6.2	Message Delivery .....	52
6.2.1	Examples.....	52
6.3	Message administration.....	53
6.3.1	Querying Message status .....	53
6.3.2	Modifying an SMS after submission.....	54
6.3.3	Cancelling an SMS after submission .....	54
7	XML Data Elements.....	55
7.1	XML Schema document structure .....	55
7.1.1	XML Schema Specification documents .....	55
7.1.2	MMAP Profile Documents .....	55
7.2	Message Transfer.....	55
7.2.1	Submit .....	56
7.2.2	Deliver .....	56
7.2.3	Expanded Submission Response .....	56
7.2.3.1	Example.....	56
7.2.4	Delivery Failure Response .....	57
7.3	Message Administration .....	57
7.3.1	QueryRequest and QueryResponse.....	57
7.3.1.1	Example.....	58
7.3.2	CancelRequest.....	58
7.3.2.1	Example.....	58
7.3.2.1.1	Cancel by service type .....	58
7.3.2.1.2	Cancel by Reference .....	59
7.3.3	ReplaceRequest.....	59
7.3.3.1	Example.....	59
7.4	Short Message Module.....	59
7.4.1	Header.....	60
7.4.1.1	DeliveryOptions .....	60
7.4.1.2	Example.....	60
7.4.2	Body .....	61
7.4.2.1	Examples.....	62
7.4.3	Delivery Info .....	62
7.4.3.1	Example.....	63
7.5	Message Extensions Module .....	63
7.5.1	NetworkHandling Element.....	63
7.5.2	ApplicationOptions element .....	64
7.5.2.1	NumberOfVoiceMails .....	64
7.5.2.2	Callback.....	65
7.5.2.3	ITSSession .....	65
7.5.2.4	ITSReply.....	65
7.5.2.5	Example.....	66
7.5.3	DeviceControl element.....	66
7.5.3.1	GSMOptions .....	66

7.5.3.2	DisplayTime.....	67
7.5.3.3	MSValidity.....	67
7.5.3.4	MessageWaitInd.....	67
7.5.3.5	ReceiveAlert.....	67
7.5.3.6	Example.....	67
7.6	Address Module.....	67
7.6.1	Number.....	70
7.6.1.1	Examples.....	70
7.6.2	Application.....	70
7.6.3	IP.....	70
7.6.4	EMail.....	70
7.6.5	Simple Addressing Examples.....	71
7.6.6	Extended Addressing.....	71
7.6.6.1	Bearer.....	71
7.6.6.2	SubAddress.....	71
7.6.6.3	Examples.....	72
7.7	Protocol Options.....	72
7.8	Protocol Profiles.....	72
8	End-to-End Examples.....	74
8.1	Binding to an SMAP service.....	74
8.2	Submitting a Message.....	76
8.3	Unbinding from the SMAP service.....	77
9	Schema Specification.....	82
9.1	MMAP Modules.....	82
9.1.1	smap.xsd.....	82
9.1.2	service.xsd.....	86
9.2	SMAP modules.....	88
9.2.1	smap.xsd.....	88
9.2.2	ShortMessage.xsd.....	88
9.2.3	MessageTransfer.xsd.....	91
9.2.4	MessageAdmin.xsd.....	93
9.2.5	Addressing.xsd.....	94
9.2.6	MessageExtensions.xsd.....	99
9.2.7	SMAPFeatures.xsd.....	105
10	Change List.....	107
10.1	Changes from version 10.....	107

## List Of Tables

Table 1-1 Glossary	15
Table 1-2 References	17

## List Of Figures

Figure 1-1 SMAP Gateway Configuration.....	9
Figure 2: Client Session.....	26
Figure 3: Asynchronous Session.....	29
Figure 4: Batch session.....	33
Figure 5: Submission to a single destination.....	50
Figure 6: submission to multiple destinations.....	51
Figure 7: Application-Controlled Distribution List Submission.....	51
Figure 8: Message Delivery.....	52
Figure 9: Query Request.....	54



# 1 Introduction

The SMS Forum, a non-profit organisation dedicated to the promotion of SMS within the wireless industry, manages the Short Message Application Protocol (SMAP). The specification and related documentation is available from the SMS Forum Website <http://www.smsforum.net>

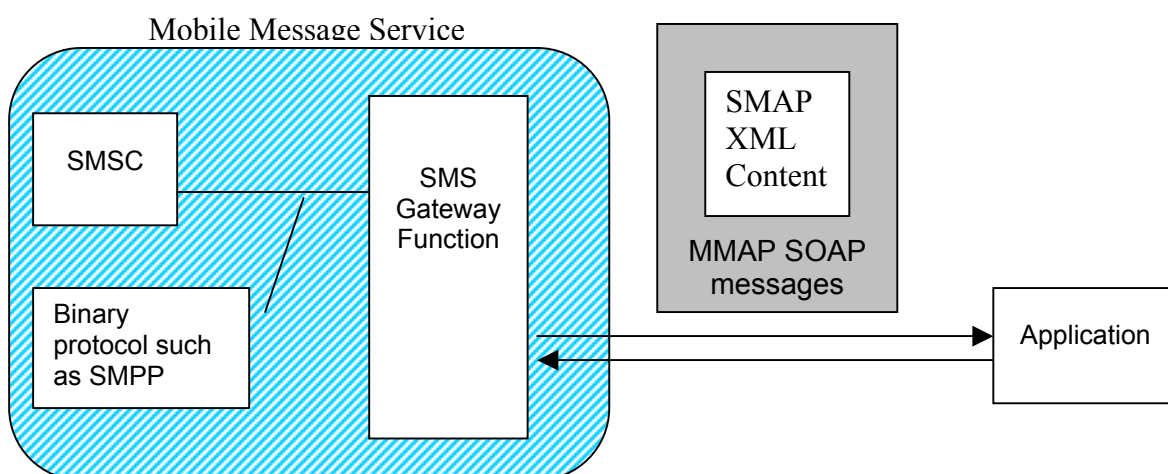
The Hypertext Transport Protocol (HTTP) is an increasingly popular transport mechanism for sending short messages from web applications to mobile phones. Many service providers implement HTTP interfaces. These interfaces are, however, implementation-specific, may offer only a limited set of functions and frequently do not have a full and formal specification.

A number of existing protocols support the transfer of messages between a Short Message Center and External Short Message Entities (ESMEs). These are normally specified as binary protocols accessed over TCP/IP or X.25. It is proposed to define a new text protocol to support application access to message centers and/or messaging gateway nodes over HTTP and other Web protocols. It is further proposed to specify this new protocol in XML. This will allow messages to be specified in a structured format and will also facilitate automatic translation to other messaging formats.

This new Mobile Message Access Protocol (MMAP) provides a standard framework for mobile messaging over SOAP and HTTP. It provides support for service identification and billing and provides a standard way of supporting session-oriented communication. MMAP supports generic XML operations as SOAP body elements.

The Short Message Application Part (SMAP) is a set of abstract XML operations for short messaging. These operations are functionally equivalent to the SMS forum's binary access protocol; SMPP version 3.4. SMAP is defined in XML as a set of abstract operations and requires MMAP or an equivalent framework for implementation

The following diagram illustrates the context of SMAP in a mobile network using an MMAP messaging service:



**Figure 1-1 SMAP Gateway Configuration**

The following diagram illustrates the context of SMAP where an SMSC contains a native-mode MMAP implementation function:

:

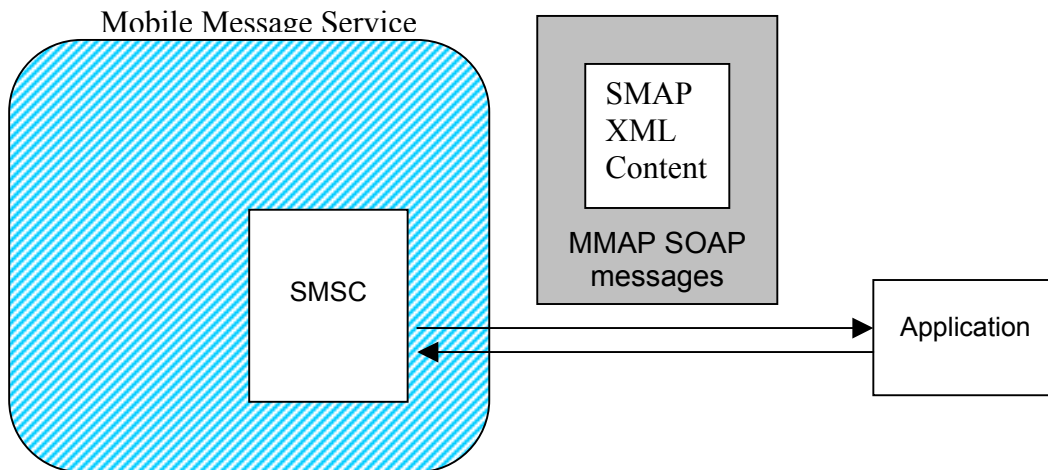


Figure 1-2 SMAP Native-mode SMSC Access

## 1.1 Scope Of This Document

This document defines Version 1.0 of the MMAP protocol. It is intended for designers and implementers of the MMAP v1.0 interface between a Messaging Service and an external Application. The protocol is presented in two parts;

**Part 1:** Mobile Message Application Protocol (MMAP): This defines a Soap Access Framework that performs the transport of mobile messages over SOAP using HTTP.

**Part 2:** Short Message Application Part: This defines SMAP protocol as a set of XML elements for sending, receiving and managing short messages

## 1.2 Overview

The following sub-sections describe briefly the basic concepts and characteristics of SMAP. Many of these characteristics will be discussed in greater detail throughout the document.

### 1.2.1 Supported Cellular Technologies

SMAP is designed to support short messaging functionality for any cellular technology and has specific applications and features for technologies such as:

- GSM
- GPRS
- CDMA
- CDMA2000
- IDEN

## 1.3 Example

The following example shows how SMAP can be used to submit a short message over the MMAP SOAP access framework. The example shows an “immediate mode” MMAP request with HTTP as a bearer.

```

POST servlet/MMAPSOAPHandler HTTP/1.1
Host: www.smsforum.net
Content-Type: Multipart/Related; boundary=MIME_boundary;
type=text/xml;
start="<tesattatat>"
Content-Length: XXXX
SOAPAction: http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd
Content-Description: SMAP Submit example.

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <tesattatat>
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="Request"
sourceOperationReference="123"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <smap:SubmitRequest
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.0
http://www.smsforum.net/schemas/smap/v1.0/smap.xsd">
      <smap:ShortMessage>
        <smap:Header>
          <smap:Destination>
            <smap:Number>+12345679</smap:Number>
          </smap:Destination>
        </smap:Header>
        <smap:Body>
          <smap:Text>Hello world</smap:Text>
        </smap:Body>
      </smap:ShortMessage>
    </smap:SubmitRequest>
  </soap:Body>
</soap:Envelope>

```

The responses is encoded as follows:

```
HTTP/1.1 200 OK
Content-Type: Multipart/Related; boundary=MIME_boundary;
type=text/xml;
start="<tdfdfdfd>"
Content-Length: XXXX
Content-Description: SMAP Submit response.
--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <tdfdfdfd>
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="SuccessResponse"
sourceOperationReference="123"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <smap:SubmitResponse
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.0
http://www.smsforum.net/schemas/smap/v1.0/smap.xsd">
      <smap:MessageRef>55566677</smap:MessageRef>
    </smap:SubmitResponse>
  </soap:Body>
</soap:Envelope>
```



## 1.4 Glossary

Term	Definition
ACK	Acknowledgement
API	Application Programming Interface
CDMA	Code Division Multiple Access
CDR	Call Detail Record
ETSI	European Telecommunications Standards Institute
GSM	Global System for Mobile Communications
IDEN	Integrated Dispatch Enhanced Network
MC	Message Centre - A generic term used to describe various types of SMS Gateways.
MSB	Most Significant Byte
MSC	Mobile Switching Centre
MS	Mobile Station
MWI	Message Waiting Indication
NACK	Negative Acknowledgement
NSAP	Network Service Access Point
PDU	Protocol Data Unit
SME	Short Message Entity
SMSC	Short Message Service Centre
SMPP	Short Message Peer to Peer Protocol
TETRA	Terrestrial Trunked Radio
TDMA	Time Division Multiple Access
UDHI	User Data Header Indicator
UMTS	Universal Mobile Telecommunications Systems
URL	Uniform Resource Locator
VMA	VoiceMail Alert
TIA	Telecommunications Industry Association
WAP	Wireless Application Protocol ( <a href="http://www.wapforum.org">http://www.wapforum.org</a> )
WCMP	Wireless Control Message Protocol
WDP	Wireless Datagram Protocol

**Table 1-1 Glossary**

## 1.5 References

Ref.	Document Title	Document Number	Version Number
[GSM 03.40]	Technical Realisation of the Short Message Service Point to Point	GSM 03.40 <a href="http://www.etsi.fr">http://www.etsi.fr</a>	v5.7.1
[GSM 03.38]	“Digital Cellular telecommunications system (Phase 2+); Alphabets and language specific information”.	[GSM 03.38] <a href="http://www.etsi.fr">http://www.etsi.fr</a>	v5.5.1 Sept. '97
[GSM MAP 09.02]	GSM Mobile Application Part	[GSM MAP 09.02] <a href="http://www.etsi.fr">http://www.etsi.fr</a>	v5.11.0
[IS637]	Short Message Service for Spread Spectrum Systems	TIA/EIA/IS-637-A	Rev A
[TSAR]	Teleservice Segmentation and Reassembly (TSAR)	TIA/EIA-136-620	Rev 0
[CMT-136]	Short Message Service - Cellular Messaging Teleservice	TIA/EIA-136-710-A	Rev A
[GUTS]	General UDP Transport Service (GUTS)	TIA/EIA-136-750	Rev 0
[WAPARCH]	Wireless Application Protocol Architecture Specification	WAP Forum <a href="http://www.wapforum.org">http://www.wapforum.org</a>	Version 30-Apr.- 1998
[WCMP]	Wireless Control Message Protocol Specification	WAP Forum <a href="http://www.wapforum.org">http://www.wapforum.org</a>	Version 12-June- 1998
[WDP]	Wireless Datagram Protocol Specification	WAP Forum <a href="http://www.wapforum.org">http://www.wapforum.org</a>	Version 10-Feb.- 1999
[ITU X.213]	Open Systems Interconnection - Network Service Definition	[ITU X.213]	11/95
[KOR ITS]	PCS operators common standards for handset-SMS functionalities	PCS standardization committee PCS-SMS-97-05-28	1.06 Rev 99-04- 30

Ref.	Document Title	Document Number	Version Number
XML	Extensible Markup Language	W3C ( <a href="http://www.w3c.org/XML">http://www.w3c.org/XML</a> )	1.0
XMLSCHEMA	XML Schema Parts 1 & 2	W3C ( <a href="http://www.w3c.org/XML/Schema">http://www.w3c.org/XML/Schema</a> )	1.0
SOAP1.1	Simple Object Access Protocol	W3C Note 8 May 2000 ( <a href="http://www.w3c.org/TR/SOAP">www.w3c.org/TR/SOAP</a> )	1.1

**Table 1-2 References**



# PART 1

## Mobile Messaging Access Protocol (MMAP): A SOAP framework for messaging services

**DRAFT**



## 2 MMAP Design Details

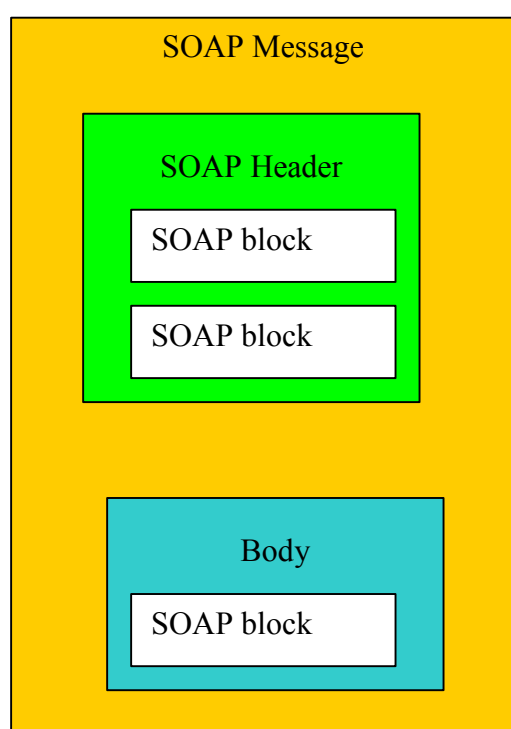
### 2.1 Overview

MMAP provides a set of SOAP elements that allow applications to perform requests and receive responses within a structured framework. MMAP provides a mechanism for validating applications and managing a session over SOAP. It also allows applications to establish an asynchronous session, whereby requests can be initiated by either party. This is particularly useful for delivery of messages to an application where message delivery is only attempted when the application is connected to the server.

### 2.2 Background

SOAP messages contain a header and a body. The SOAP header defines application-specific information which may be modified as a message is passed through different SOAP nodes. The SOAP body contains information for the ultimate recipient of the message. For MMAP this can be considered to be information destined for the SMS application or for the target SMS network.

The following diagram shows an abstract view of a SOAP message:



In an actual implementation of a SOAP protocol, these blocks are further wrapped inside different transport frameworks, such as HTTP.

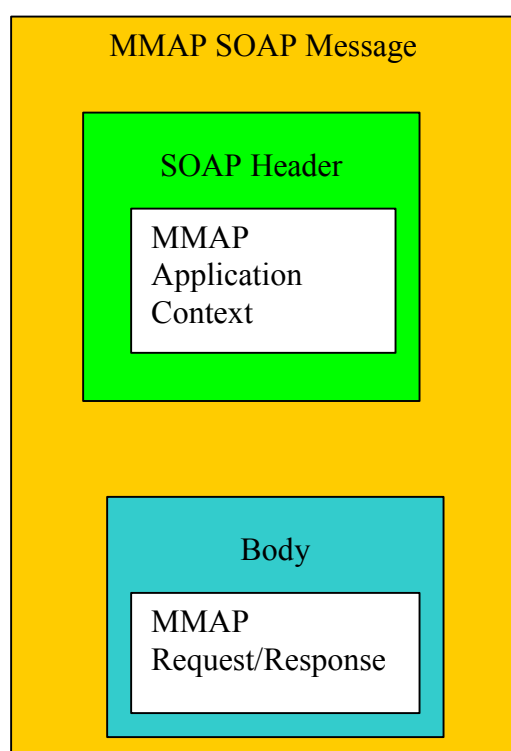
SOAP has two modes of operation: remote procedure call (RPC) and messaging. The RPC mode is essentially a mechanism for mapping procedures calls onto an XML structure in a standard fashion. The messaging model provides a more flexible mode of interaction, with arbitrary messages being exchanged between message agents.

## 2.3 SOAP design decisions

It is proposed to use the SOAP messaging mode for this specification. This has the advantage of being more flexible in both the interaction mechanism and in the data-types used for the specification.

It is proposed to provide a generic request/response framework for messaging applications. This will support both state-less operations and a peer-to-peer protocol. This framework will utilise SOAP message format as follows:

- A dedicated SOAP message header provides application context information for both stateless operations and operations which occur in a peer-to-peer session.
- The SOAP body carries a standard operation request or response.



## 2.4 Operational modes

It has been decided to support the following modes of operation for applications:

### Immediate mode:

In this mode the application does not maintain a session with the mobile message service. Each SOAP message contains a service context that specifies access control information and a single request. The message service sends a response only when it has performed the requested operation. The SOAP response contains as its body element the associated response. This mode is used only for message submission.

### Client session mode:

In this mode the application establishes a session with the message service through a bind request. Each SOAP message contains an application context that identifies the session and a single request. The message service sends a response only when it has performed the requested operation. The response contains as application information the associated MMAP response element. The message service can also establish a client session with the application, in order to invoke operations on the application.

#### **Peer-to-peer session mode**

In this mode, the application and the message service establish a bi-directional session. Once this session is established the message service responds to requests immediately on reception. This response does not contain any application-level information and only indicates reception of the request by the message service. Once the operation has been completed by the gateway, the response is sent to the client in a separate SOAP message. The message service can also generate operations asynchronously on the same session.

#### **Batch mode**

In this mode, the message service receives a set of MMAP requests to process. The MMAP data consists of an application context element followed by a set of requests. The message service processes all of the requests and then generates a new batch element containing the respective response elements. The message service then sends the complete batch response back to the client. The message service can use the same mechanism to perform delivery operations to the application.

Batch mode is normally used when an interactive session is not required or would be unsuitable due to timeout problems. For example, if an application wishes to send an SMS to a distribution list containing many thousands of names, a batch request is probably the most appropriate mechanism.

## **2.5 Session management**

MMAP provides a set of primitives for session management. These allow an application to bind to a server, invoke a series of operations on a server and end the session by unbinding from the server. A session can be a pure client session, where one party invokes operations on the other. A peer-to-peer session can also be established when a client provides bind-back details to the server. This allows the server to bind back to the client using a URL and connection details provided by the client in the original bind request. Once the bindback operation has succeeded the two client-server sessions operate together as a single peer-to-peer messaging session. Each SOAP request is accepted for processing and the immediate response indicates this acceptance. When the request completes, the response is sent back on the peer session.

As part of the bind process, the requesting agent can specify the operations required and can request optional features. In the bind response, the responding agent indicates the operations and features that are actually supported on the session. There is no explicit negotiation process; the exchange of information simply indicates to each agent the operations that are possible on that session.

The establishment of a session also allows each agent to perform throttling on incoming requests. An explicit window of outstanding requests can be specified for each end of a session and the requesting agent should not exceed this window when performing a request.

## **2.6 Security**

SOAP does not in itself specify any security mechanisms. However, the mechanism to extend SOAP headers to provide XML standards for authentication with overlay authorization framework

For initial release simple authentication only will be supported.

## 3 SOAP Procedures

### 3.1 Session Management

#### 3.1.1 Overview

A session is established between an application and a message service through a bind procedure. Binding provides a mechanism for identification and authentication of the communicating entities and also provides facilities for identifying the capabilities of each entity and the type of session required. The session type can be one of "Client" or "Peer".

No session is needed where a request is made in immediate mode. Instead, the application includes the required access control information in an application context element with each request.

If requests are made in batch mode, a single application context element provides the access control information. Since all of the information is supplied in a single XML element, there is no need to establish a session between the two entities.

##### 3.1.1.1 Identification

The Bind procedure provides the message service with

1. The identity of the application binding to it
2. Account information for any billing associated with the session
3. Application-specific settings for the session

##### 3.1.1.2 Authentication

In the initial release, the bind procedure provides simple password authentication. If security is an issue, the underlying transport should provide encryption facilities.

It is envisioned that in a future release, more sophisticated certificate-based authentication could be introduced. This could be an independent mechanism or, preferably, could be bound to an underlying security framework.

##### 3.1.1.3 Capability presentation

As part of the bind procedure, the entities exchange capability information. These define

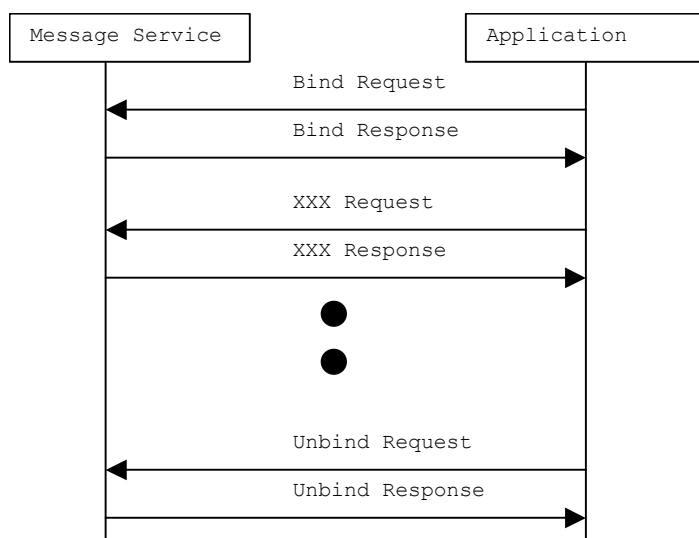
1. The operations required by the originating entity
2. The operations supported by the originating entity
3. Application-specific options and parameters. For these, the originator specifies those required and the responder identifies the set actually supported.

This is simply a capability presentation procedure and not a complete capability negotiation process. If a requested facility is not available, the requesting entity can either continue without it or terminate the session.

### 3.1.2 Client Session

In a client session, the response to each request is only generated when the request has been performed completely. Either the message service or the application initiates a client session. When asynchronous message submission and delivery is required, two independent client sessions must be established. The SM application initiates a session to perform asynchronous message submission. The message service initiates a second session to perform asynchronous message delivery.

The following diagram shows an application initiating and using a client session.



**Figure 2: Client Session**

#### 3.1.2.1 Example

This XML example shows a simple bind request:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="Request"
sourceOperationReference="123"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:BindRequest
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:SessionControl maxOutstandingRequests="3"
sessionType="client">
        </MMAP:SessionControl>
        <MMAP:FeatureSet>
          <MMAP:RequiredOperations>xxxRequest yyyRequest
        </MMAP:RequiredOperations>
        </MMAP:FeatureSet>
      </MMAP:BindRequest>
    </soap:Body>
  </soap:Envelope>
```

The response is encoded as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="SuccessResponse"
sourceOperationReference="123" sessionId="45"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:BindResponse
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:SessionControl maxOutstandingRequests="1"
></MMAP:SessionControl>
      <MMAP:FeatureSet>
        <MMAP:SupportedOperations>xxxRequest yyyRequest
        </MMAP:SupportedOperations>
      </MMAP:FeatureSet>
    </MMAP:BindResponse></soap:Body>
</soap:Envelope>

```

### 3.1.3 Peer-to-peer Session

A peer-to-peer session allows an application and message service to exchange information as peers in a single session. Each entity can generate a request asynchronously. The reception of the request is acknowledged immediately and the response is returned asynchronously to the requesting entity. This mode of operation does not map transparently onto a pure client-server protocol such as HTTP, where it is necessary to maintain the bind over two independent client-server sessions. In one of these sessions, the message service acts as a server and the application acts as a client. In the other, the message service acts as a client and the application acts as a server. In order to support this mode of operation, the initial bind may contain additional addressing and security information to allow the other entity to generate a back bind that can be logically linked to the first session. This then allows the two entities to communicate in a peer-to-peer fashion across two independent client-server links.

The following diagram shows how this can operate

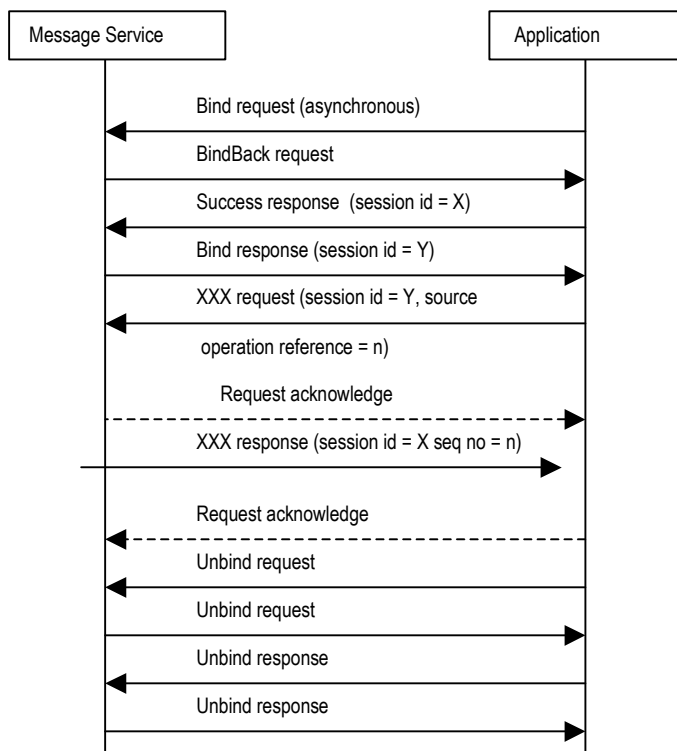


Figure 3: Asynchronous Session

### 3.1.3.1 Example

This XML example shows a peer bind request:

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="Request"
sourceOperationReference="123"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:BindRequest
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:SessionControl maxOutstandingRequests="3"
sessionType="peer">
        <MMAP:ReturnSession>
          <MMAP:SessionReturnPath
link="http://www.aaa.bbb.ccc/">
            <MMAP:Parameters>
              <xxx>custom client info</xxx>
            </MMAP:Parameters>
          </MMAP:ReturnSession>
        </MMAP:SessionControl>
        <MMAP:FeatureSet>
          <MMAP:RequiredOperations>xxxRequest yyyRequest
          </MMAP:RequiredOperations>
        </MMAP:FeatureSet>
      </MMAP:BindRequest>
    </soap:Body>
  </soap:Envelope>

```

The server then generates a bindback request to the URL <http://www.aaa.bbb.ccc>:

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="Request"
sourceOperationReference="1"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:BindBackRequest
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:Parameters>
        <xxx>custom client info</xxx>
      </MMAP:Parameters>
      <MMAP:SessionControl maxOutstandingRequests="3" />
      <MMAP:FeatureSet>
        <MMAP:RequiredOperations>aaaRequest bbbRequest
        </MMAP:RequiredOperations>
      </MMAP:FeatureSet>
    </MMAP:BindBackRequest>
  </soap:Body>
</soap:Envelope>

```

The bindBack response is encoded as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="SuccessResponse"
sourceOperationReference="1" sessionId="99"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <mmap:BindResponse xmlns:mmap="http://www.smsforum.net/mmap"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/mmap/mmap.xsd">
      <mmap:SessionControl maxOutstandingRequests="5"/>
      <mmap:FeatureSet >
        <mmap:SupportedOperations>aaaRequest bbbRequest
        </mmap:SupportedOperations>
      </mmap:FeatureSet>
    </mmap:BindResponse>
  </soap:Body>
</soap:Envelope>

```

Once the message server has received a successful response to the bindback request, it generates a response to the original bind request.

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="SuccessResponse"
sourceOperationReference="123" sessionId="9957"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <mmap:BindResponse xmlns:mmap="http://www.smsforum.net/mmap"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/mmap/mmap.xsd">
      <mmap:SessionControl maxOutstandingRequests="5"/>
      <mmap:FeatureSet >
        <mmap:SupportedOperations>XXXRequest YYYRequest
        </mmap:SupportedOperations>
      </mmap:FeatureSet>
    </mmap:BindResponse>
  </soap:Body>
</soap:Envelope>

```

Both parties now have a remote session id on which they can send requests and a local session id on which they can receive requests. A full peer session is thus established.

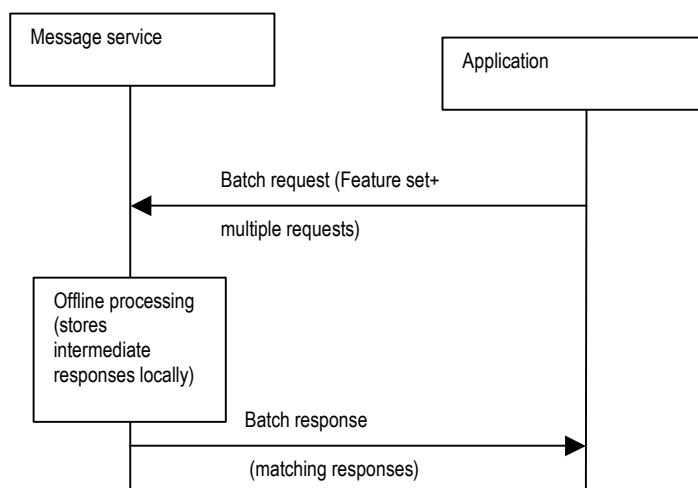
### 3.1.4 Batch processing

Batch mode allows an application to submit one or more requests for offline processing. This can be used, for example, when submitting a large number of messages or submitting a single message to a large set of recipients.

Each batch request contains an application context header and set of requests or responses. These elements would normally be submitted via an asynchronous bearer protocol such as FTP or SMTP. However, there is nothing to prevent a batch session being initiated over HTTP.

A batch response contains an individual response for each request in the original batch request

This can be seen in the following diagram:



**Figure 4: Batch session**

#### 3.1.4.1 Example

This XML example shows a simple batch request:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="Request"
sourceOperationReference="123"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:BatchRequest batchRequestReference="myref121"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:FeatureSet>
        <MMAP:RequiredOperations>xxxRequest yyyRequest
        </MMAP:RequiredOperations>
      </MMAP:FeatureSet>
      <MMAP:Request batchOperationReference="1">
        <XXXRequest>12345</xxxRequest>
      </MMAP:Request>
      <MMAP:Request batchOperationReference="2">
        <XXXRequest>54321</xxxRequest>
      </MMAP:Request>
      <MMAP:Request batchOperationReference="3">
        <yyyRequest/>
      </MMAP:Request>
    </MMAP:BatchRequest>
  </soap:Body>
</soap:Envelope>
```

The batch response is encoded as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="SuccessResponse"
sourceOperationReference="123" sessionId="45"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:BatchResponse batchRequestReference="myref121"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:RequestSucceeded batchOperationReference="1"/>
      <MMAP:RequestSucceeded batchOperationReference="2"/>
      <MMAP:RequestFailed batchOperationReference="3"/>
    </MMAP:BatchResponse>
  </soap:Body>
</soap:Envelope>

```

## 3.2 Request Handling

### 3.2.1 Immediate Mode

When a messaging entity submits an SOAP message with no session information in its MMAP application context element, this is treated as an immediate-mode request. The receiving entity parses the ServiceContext element to authenticate the request and to obtain billing information. It then performs the request and returns the result as the body of a response message. The response message has an MMAP application header containing only an Application context element. This application context element has a bodyType attribute indicating that the message is a response and it contains a sourceOperationReference attribute if this was supplied in the original request.

A receiving entity may support multiple outstanding immediate mode requests from the same source. It then becomes the responsibility of the message transmitter to use the sourceOperationReference attribute to distinguish between different responses. However, it should be noted that there is no requirement on the receiving entity to support multiple outstanding requests and it may choose to reject any new requests while one or more are still outstanding.

It should be noted that an MMAP bind request is itself an immediate mode request and the bind response is only sent when the full bind operation has been completed.

### 3.2.2 Client Session Mode

When a messaging entity binds as a client, it receives a session identity in the application context of the bind response. It must specify this session identity in each subsequent request. Each client request is handled synchronously by the receiving entity. As for immediate mode requests, the `sourceOperationReference` attribute provided by the sender is returned in the response.

In a client session, the receiving entity must support multiple outstanding requests up to the amount specified in the `maxOutstandingRequests` attribute in the `BindResponse` element. This allows the application to manage windowing of requests on a session.

### 3.2.3 Peer-to-peer Session Mode

When a messaging entity binds as a peer, it receives a session identity in the application context of the bind response. It must specify this session identity in each subsequent request. It also supplies a session identity when responding to the associated `BindBack` request. Each client request is handled asynchronously by the receiving entity. A HTTP response is returned immediately the request is accepted for processing. When the request actually completes, the response is sent in a separate SOAP message on the associated `bindback` session. In this message, the MMAP header contains an `applicationcontext` element with the following settings:

- **bodyType**: "SuccessResponse"
- **sessionId**: `bindback` session identity
- **sourceOperationReference**: value as specified in the original request

## 3.3 Error Handling

The standard SOAP fault reporting mechanism allows receiving entities to manage SOAP processing errors. Additionally, a receiving entity may reject a message due to application-level problems. MMAP provides a generic error response mechanism to cater for this. Where errors are completely application-specific, the receiving entity simply sends back a SOAP message containing an application-specific body with a MMAP header containing an `ApplicationContext` element with a `bodyType` attribute value of "ErrorResponse".

MMAP also provides a standard `ErrorResponse` element that is used by MMAP itself for internal protocol errors and may also be extended by any other application.

## 4 SOAP data elements

A SOAP Envelope contains a number of header elements and a body element. MMAP defines a single MMAP header element that must be present in all MMAP messages and also defines a set of body elements for MMAP operations such as binding.

### 4.1 MMAP Header

An MMAP header contains two elements:

- ApplicationContext and
- ServiceContext (optional)

#### 4.1.1 ApplicationContext element

This element is used to provide a context for an operation. This empty element has the following attributes:

**bodyType:** This mandatory attribute specifies the meaning of the SOAP body. It can have one of the following values: "Request", "SuccessResponse", "ErrorResponse".

**sessionId:** This optional attribute is only specified when request or response is within a session. It identifies the session as managed by the target of the operation except when the operation is a response to a Bind or BackBind request. In these two special cases it identifies the newly-created session on the source entity.

**sourceOperationReference:** This optional parameter allows the initiator of a request to identify the associated response. In a response it has the same value as in the associated request. This integer parameter completely under the control of the request initiator and is simply replayed transparently in the response message. The value for this attribute must be an unsigned 32-bit integer.

The ApplicationContext element has one optional child element: AccessControl. This contains an ApplicationIdentity element that identifies the application and an Authentication element currently contains a password element. It is envisioned that these elements may be expanded in the future to contain certificates or other authentication tokens.

```
<mmap:ApplicationContext
  bodyType="Request"
  sourceOperationReference="1235"/>
```

#### 4.1.2 ServiceContext element

This optional element is used to specify the service-level information associated with a request. It may be absent if this information is not required by the request handler (e.g. if the service is free and publically available) or if all the information has been supplied during the session bind procedure.

A ServiceContext has one mandatory attribute: serviceName. This identifies the requested service.

A ServiceContext contains the following optional child elements:

- AccessControl and
- Billing

#### 4.1.2.1 AccessControl element

This element contains an ApplicationIdentity element that identifies the application and an Authentication element currently contains a password element. It is envisioned that these elements may be expanded in the future to contain certificates or other authentication tokens.

```
<mmap:AccessControl>
<mmap:ApplicationIdentity>testApp</mmap:ApplicationIdentity>
  <mmap:Authentication>
    <mmap:Password>12345</mmap:Password>
  </mmap:Authentication>
</mmap:AccessControl>
```

#### 4.1.3 Billing

A "Billing" element can contain any valid XML content. This allows a service provider an an application to transfer whatever information is needed to meet their service agreement.

Some billing datatypes are provided in the mmap service schema. However, these are for reference only and do not constrain implementations in any way.

##### 4.1.3.1 Example

```
<mmap:Billing identity="Billid1">
  <Cost>
    <CostClass type="Premium"/>
  </Cost>
  < ApplyTo>
    <Name>myService</Name>
    <Account>234545990</Account>
  </ApplyTo>
  <Description>
    this is a sample billing element
  </Description>
</mmap:Billing>
```

## 4.2 SOAP Body Elements

### 4.2.1 BindRequest

A **BindRequest** element contains the information needed to establish a session with a gateway. It has no attributes and can have the following child elements:

- **SessionControl**
- **FeatureSet** (multiple instances)

The **SessionControl** element has two attributes:

**sessionType**. This specifies the type of session being established as either “client” (the default) or “peer”. It can contain an optional **ReturnSession** element.

**MaxOutstandingRequests**: This specifies the requested window size for the session

Each **FeatureSet** element can contain the following elements in any order:

**RequiredOperations**: This element contains a space-separated list of the names of the body elements that may be sent as requests in the current session.

**SupportedOperations**: This element contains a space-separated list of the names of the body elements that may be sent or received as requests by the requesting agent.

**Options**: This element contains a space-separated list of application-specific features.

**Parameters**: This empty element contains a set of application-specific attributes indicating parameter settings.

**ApplicationSettings**: This element contains any application settings for the current settings. Any XML content is allowed.

If present, the **ReturnSession** element specifies that a return session should be established by the gateway using the service type identified by the gateway and attaching the parameters specified by the application in the body of the element. The application can specify whatever parameters it wishes to be returned by the gateway.

If a client session is requested with a **ReturnSession** element, the message service establishes a separate client session using the **ReturnSession** parameters. Otherwise, the message service uses the parameters to invoke a **bindBack** request and establish a full peer session.

#### 4.2.1.1 Example

```

<MMAP:BindRequest
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
  <MMAP:SessionControl maxOutstandingRequests="3"
sessionType="client">
    <MMAP:ReturnSession>
      <MMAP:SessionReturnPath
link="http://www.aaa.bbb.ccc/"></MMAP:SessionReturnPath>
    </MMAP:ReturnSession>
  </MMAP:SessionControl>
  <MMAP:FeatureSet>
    <MMAP:ApplicationSettings>
      <MMAP:RequiredOperations>xxxRequest
yyyRequest</MMAP:RequiredOperations>
    </MMAP:ApplicationSettings>
  </MMAP:FeatureSet>
</MMAP:BindRequest>

```

## 4.2.2 BindResponse

A BindResponse element confirms that a session has been established with the server. It also specifies the features supported by the server.

### 4.2.2.1 Example

```

<MMAP:BindResponse
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
  <MMAP:SessionControl maxOutstandingRequests="1"
></MMAP:SessionControl>
  <MMAP:FeatureSet>
    <MMAP:ApplicationSettings>
      <MMAP:SupportedOperations>xxxRequest
yyyRequest</MMAP:SupportedOperations>
    </MMAP:ApplicationSettings>
  </MMAP:FeatureSet>
</MMAP:BindResponse>

```

## 4.2.3 BindBack Request

The BindBack element is used when the message service wishes to bind a return path for a new peer session. The message service makes the request to the bindback URL specified in the ReturnSession element of the original BindRequest. The BindBackRequest can contain the following elements (all optional):

**SessionControl:** This empty element has one attribute  
**MaxOutstandingRequests:** This specifies the requested window size for the return link of the session

**FeatureSet:** This element can contain the same elements as for a bind request

**Parameters:** This element contains any client-supplied parameters from the original bind request.

#### 4.2.3.1 Example

```
<mmap:BindBackRequest>
  <mmap:Parameters>
    <myparam>test</myparam>
  </mmap:Parameters>
</mmap:BindBackRequest>
```

#### 4.2.4 Unbind Request

The UnbindRequest element is used when the message service wishes to unbind from a session. This element is empty.

#### 4.2.5 Unbind Response

The UnbindResponse element is returned when an UnbindRequest completes successfully. In the case of a peer session, this element is only returned after the return link of the session has been unbound. The element body is empty.

#### 4.2.6 Batch Request

The BatchRequest contains a sequence of Request to be processed offline. It has one mandatory attribute: RequestReference. This is used in a BatchResponse to identify the original batch request. A batch request can contain multiple instances of the following child elements:

**FeatureSet:** This element is defined in the bind request section.

**Request:** This can contain any XML content, this being an application-specific operation. The element has one mandatory attribute: OperationReference. This is used to identify the individual operation in a BatchResponse element.

##### 4.2.6.1 Example

This XML example shows a simple batch request:

```

<MMAP:BatchRequest batchRequestReference="myref121" >
  <MMAP:FeatureSet>
    <MMAP:ApplicationSettings>
      <MMAP:RequiredOperations>xxxRequest
yyyRequest</MMAP:RequiredOperations>
    </MMAP:ApplicationSettings>
  </MMAP:FeatureSet>
  <MMAP:Request batchOperationReference="1">
    <XXXRequest>12345</xxxRequest>
  </MMAP:Request>
  <MMAP:Request batchOperationReference="2">
    <XXXRequest>54321</xxxRequest>
  </MMAP:Request>
  <MMAP:Request batchOperationReference="3">
    <yyyRequest/>
  </MMAP:Request>
</MMAP:BatchRequest>

```

## 4.2.7 Batch Response

The BatchResponse element contains a sequence of responses to a previously supplied BatchRequest. It has one mandatory attribute: RequestReference. This is used to identify the original batch request. A batch request can contain multiple instances of the following child elements:

**RequestSucceeded:** This contains the application-generated response to a successful operation. It can contain any XML content supplied by the application. The element has one mandatory attribute: OperationReference. This is used to identify the source operation from the BatchRequest.

**RequestFailed:** This identifies a failed operation. It can contain any XML content supplied by the application. The element has one mandatory attribute: OperationReference. This is used to identify the source operation from the BatchRequest.

### 4.2.7.1 Example

This XML example shows a simple batch response

```

<?xml version="1.0" encoding="UTF-8"?>
<MMAP:BatchResponse batchRequestReference="myref121" >
  <MMAP:RequestSucceeded batchOperationReference="1"/>
  <MMAP:RequestSucceeded batchOperationReference="2"/>
  <MMAP:RequestFailed batchOperationReference="3"/>
</MMAP:BatchResponse>
</soap:Body>
</soap:Envelope>

```

#### **4.2.8 EnquireLink Request**

The EnquireLink element is used by either party to check whether a client or peer session is still active. This element is empty.

#### **4.2.9 EnquireLink Response**

The EnquireLinkResponse element is returned when an EnquireLinkRequest completes successfully. The element body is empty.

### **4.3 Accessing MMAP**

MMAP is specified for a target namespace of

**<http://www.smsforum.net/schemas/mmap/v1.0>".**

The actual schema definitions are accessible at the following URL:

**<http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd>**



# PART 2

## Short Message

Application Part (SMAP):

Abstract XML Operations for  
short messaging applications

**DRAFT**



## 5 SMAP Design Principles

### 5.1 Summary of Approach

#### 5.1.1 General Principles

The following design principles have been established:

**Ease of Use:** Users with little or no knowledge of SMS or telecom protocols should be able to send and receive messages using a simple set of primitives.

**Modularity:** The protocol should be defined as a set of modules, to support interworking with other systems and applications.

**Readability:** Wherever possible attributes and elements should have human readable values rather than “magic” numbers;

**Rich functionality:** All of the functions of a binary access protocol (such as SMPP V3.4) shall be available to advanced users.

**Transport-independence:** it should be possible to provide alternative transport protocols to HTTP, for instance file transfer or SMTP.

#### 5.1.2 Design Decisions

##### 5.1.2.1 Modularisation

The SMAP specification has been split into modules. The following modules have been defined:

1. **Addressing:** This module defines a set of generic addressing formats for short messaging entities. It integrates a number of optional and mandatory addressing parameters as components of a single SMAddress Element.
2. **Short Message:** This module provides a generic definition of a short message and a delivery information element. Together with the addressing module it provides a protocol-independent definition of a short message that can be used as part of any XML-based Short Message protocol. (e.g. for onward forwarding of messages within an enterprise).
3. **Message Transfer:** This module defines a basic set of operations for short message applications. These provide:
  - a. Message submission
  - b. Message delivery
4. **Message Administration:** This module defines the following SMAP operations:
  - Message status query
  - Cancel message in transit
  - Replace message contents in transit

5. **Message Extensions:** This module extends the message header and body with extra information for mobile messaging. This includes:
- Network handling options
  - Application information
  - Device control options
  - User Data Header information

### 5.1.2.2 Language and character-set selection

This is handled as follows:

- a. If the user specifies the message body as containing text, only the language may be specified and the text is standard XML Unicode as defined in The Unicode Standard, Version 2.0. The gateway will handle any character-set conversion from Unicode to other network or device-specific encodings (e.g. default GSM alphabet).
- b. If the user specifies the body as BinaryData, the user can specify the encoding of the data explicitly. The user supplies in a hex string the actual data to be sent to the SMSC.
- c. The standard `xml:lang` attribute shall be used to specify language. The allowable values for this attribute are defined in IETF RFC1766.

### 5.1.3 XML Implementation choices

Use of XML requires some overall design decisions to be made. In particular XML offers three different possibilities for handling information within an XML element. Information can be specified as;

1. Body text in the element
2. As another element (i.e. a child of the original element)
3. As an attribute of the element

In this specification the following approach is proposed:

- Where parameters may be extended in the future or can be decomposed into simpler elements, they will be specified as child elements.
- Where parameters can be specified as one of a bounded range of values, these values will be specified as a choice of attribute values.
- Where a parameter is an integer or is a fixed-size binary value of less than 4 octets, it will be specified as an attribute of an element.
- Where a parameter is accessed in more than one place it will normally be defined as an element.

- Where a parameter contains human-readable text or is a larger binary value (> 4 octets) it will normally be specified as the body text of an element.

*Note: the word “normally” in the above list indicates that the recommended approach is not mandatory.*

#### **5.1.4 XML Naming Guidelines**

The following guidelines are followed in this specification:

- XML element names shall use Upper Camel Case convention, where words are concatenated to form an element name with the first letter of each word in upper case (e.g. ElementName) . No hyphens or underscores shall be permitted in the name.
- XML attribute names shall use Lower Camel Case convention, where words are concatenated to form an attribute name. The first word in the element shall be in lower case and all the other words have their first letter in upper case (e.g. attributeName). No hyphens or underscored shall be permitted in the name.
- The only exception to these case conventions is where an acronym is used instead of an English word (e.g. GSM). In this case all of the letters of the acronym shall be in upper case even where the acronym forms the first or only word of an attribute name.
- All XML schema datatype names shall use the suffix “Type” to distinguish them from elements and attributes. The only exception to these rules is where an acronym is used instead of an English word (e.g. GSM). In this case all of the letters of the acronym shall be in upper case even where the acronym forms the first or only word of an attribute name.

The XML definitions are specified for a target namespace of <http://www.smsforum.net/schemas/smap/v1.0>.

## 6 SMAP Procedures

### 6.1 Message Submission

The message submission procedure allows the application to submit a Short Message for delivery to one or more mobile stations (MS). The address of each MS may be specified as part of the submission request or it may be specified in a separate distribution list accessed via an XML link.

#### 6.1.1 Submission to a single destination

When an application submits a message with a single destination specified, the handles the delivery of the message to the specified destination. Once the message is under the control of the message service, it returns a submission response to the application. This contains a unique identity for the submitted message and can be used for further queries on the message or to track delivery receipts.

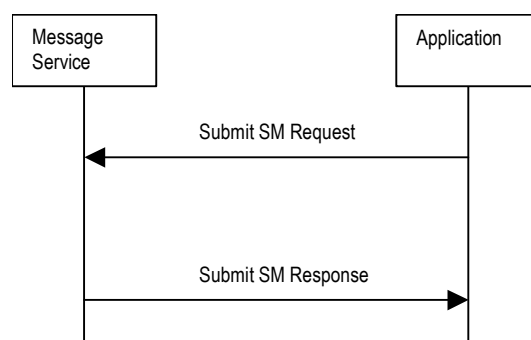


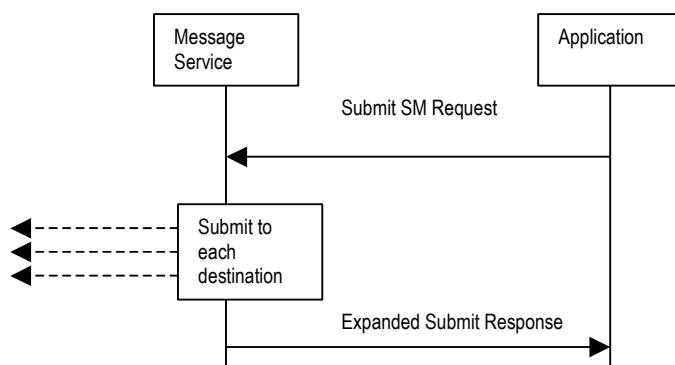
Figure 5: Submission to a single destination

#### 6.1.2 Submission to multiple destinations

An application can identify multiple destinations for a message. The message service manages the delivery of a separate message to each destination on behalf of the application..

The message service may return an expanded submission response element when a submission request specifies multiple destinations. This response identifies the destinations to which submission has failed and the destinations to which submission succeeded. For each successful submission the destination address and the associated message identity is returned. For each failed submission, the destination address and the associated error information is returned.

If the message service returns a normal submission response containing a single message identifier, there is no guarantee that the message has been submitted to all destinations.

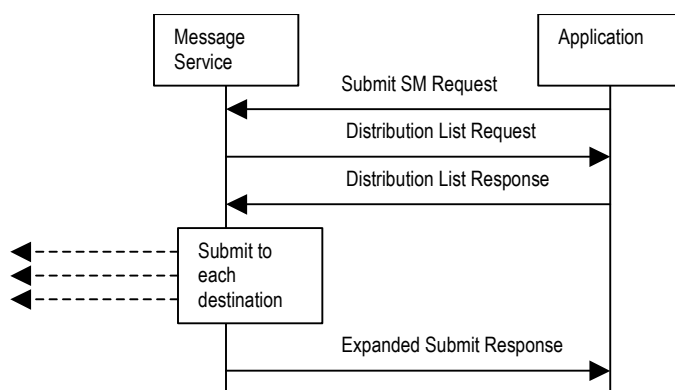


**Figure 6: submission to multiple destinations**

### 6.1.3 Submission using a distribution list

Instead of specifying the destinations within the header of the short message, the application can specify a link to a distribution list containing the destination addresses. Since this link is a standard XML Xlink attribute it can identify

- An XML structure contained within the request itself (i.e. The link contains the content id of a separate MiME component in the request)
- XML data within the gateway itself (i.e. a gateway-maintained distribution list)
- XML data within the application (i.e. the URL of an application-controlled distribution list)



**Figure 7: Application-Controlled Distribution List Submission**

#### 6.1.3.1 Example

A simple submission request can be encoded as follows

```
<?xml version="1.0" encoding="UTF-8"?>
<smap:SubmitRequest
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.0
http://www.smsforum.net/schemas/smap/v1.0/MessageTransfer.xsd">
  <smap:ShortMessage>
    <smap:Header>
      <smap:Destination>
        <smap:Number>+44123456</smap:Number>
      </smap:Destination>
    </smap:Header>
    <smap:Body>
      <smap:Text>hello world</smap:Text>
    </smap:Body>
  </smap:ShortMessage>
</smap:SubmitRequest>
```

The message service will then return a submission response as follows:

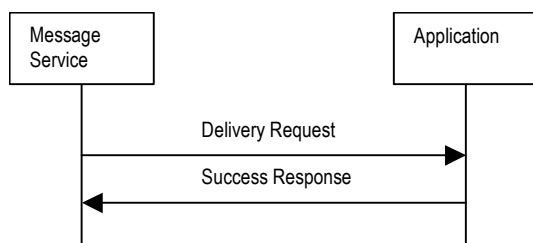
```
<?xml version="1.0" encoding="UTF-8"?>
<smap:SubmitResponse>
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.0
http://www.smsforum.net/schemas/smap/v1.0/MessageTransfer.xsd">
  <smap:MessageRef>4A5678ABFF88</smap:MessageRef>
</smap:SubmitResponse >
```

## 6.2 Message Delivery

The Short Message Service delivers short messages and delivery information to applications using a `DeliveryRequest` element. The application replies with a `SuccessResponse` element to acknowledge delivery.

The body of a `DeliveryRequest` contains either a `ShortMessage` element or else a `DeliveryInfo` element. This allows the application to distinguish messages that contain delivery information those that contain actual application data and, if required, process them separately.

The following diagram shows the operation of a delivery request.



**Figure 8: Message Delivery**

### 6.2.1 Examples

The Short Message Service sends a Message Delivery request to the application in the following format

```
<?xml version="1.0" encoding="UTF-8"?>
<smap:DeliverRequest
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.0
MessageTransfer.xsd">
  <smap:ShortMessage>
    <smap:Header>
      <smap:Destination>
        <smap:Application>555577</smap:Application>
      </smap:Destination>
      <smap:Source>
        <smap:Number TON="International"
NPI="ISDN">+44123456</smap:Number>
      </smap:Source>
    </smap:Header>
    <smap:Body>
      <smap:Text>hello world</smap:Text>
    </smap:Body>
  </smap:ShortMessage>
</smap:DeliverRequest>
```

The application then sends a success response to the Short Message Service with the following format

```
<?xml version="1.0" encoding="UTF-8"?>
<smap:SuccessResponse
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.00
http://www.smsforum.net/schemas/smap/v1.0/MessageTransfer.xsd"/>
```

## 6.3 Message administration

These facilities allow an application to manage a message after it has been submitted to the Message Service. An application can:

- Query the status of a submitted message
- Modify the contents of a submitted message
- Delete a message that has been submitted but is not yet delivered to its destination.

Since these operations provide direct access to messages within the short message network, they are defined as privileged operations and the Message Service may choose to disable them completely or only provide certain applications with access to them.

### 6.3.1 Querying Message status

An application can query the status of a message within the short message network. The response indicates whether the message is still in transit or has been delivered to its final destination. If delivered, the response may include the delivery time of the message.

The following diagram shows the operation of a query request.

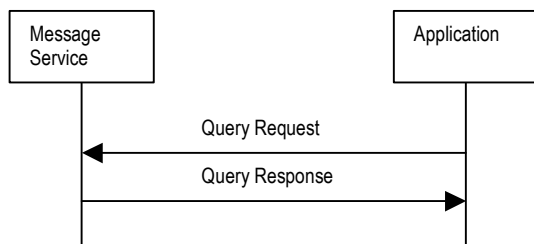


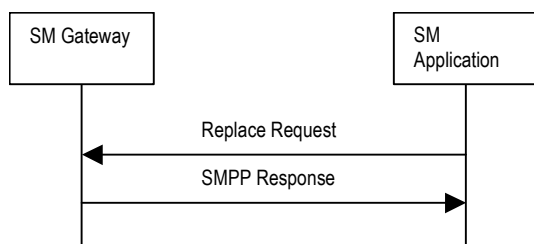
Figure 9: Query Request

### 6.3.2 Modifying an SMS after submission

An application may modify certain fields of a message after submission if the message is not yet delivered to its final destination. The application can identify the message explicitly through its message identifier or implicitly through specifying the destination address and the service type of the message.

The application can modify the message contents, the scheduled delivery time and the delivery options of the message.

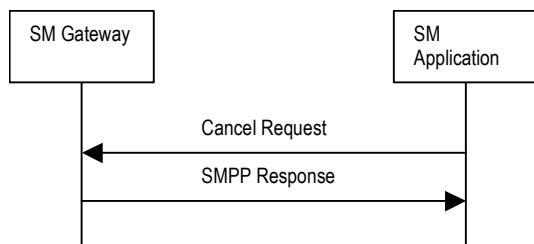
The following diagram shows the operation of a replace request.



### 6.3.3 Cancelling an SMS after submission

The application can cancel an SMS that has been submitted to the network, but is not yet delivered to its final destination. The application can identify the message explicitly through its message identifier or implicitly through specifying the destination address and the service type of the message.

The following diagram shows the operation of a cancel request.



## 7 XML Data Elements

### 7.1 XML Schema document structure

This protocol specification is divided into two sets of documents:

- Protocol specifications: These define the basic data elements of the specification.
- MMAP Profile specifications: Each of these defines a standard set of operations, options and parameters that is supported completely or not at all by a given messaging service and application.

#### 7.1.1 XML Schema Specification documents

The protocol specification is structured into the following set of documents:

Filename	Description
MessageTransfer.xsd	Defines the requests and responses associated with sending and receiving short messages.
ShortMessage.xsd	Defines the basic structure of a short message
Addressing.xsd	Defines the addressing elements needed to identify mobile devices and applications.
MessageExtensions.xsd	Defines an extended set of header parameters and body types for a short-message.
MessageAdmin.xsd	Defines message administration requests and responses.

#### 7.1.2 MMAP Profile Documents

The following profile documents are defined:

*SMAPBasic.xml*

*SMAPAdmin.xml*

In addition, messaging services may define their own MMAP profiles and make them available to applications

## 7.2 Message Transfer

This operations provided by this module allow an application to send and receive short messages. The following top-level elements are defined:

- SubmitRequest
- DeliverRequest
- ExpandedSubmissionResponse

### 7.2.1 Submit

A “SubmitRequest” is used to submit a short message or delivery acknowledgement message to Short Message Service. It contains one of the following child element:

- ShortMessage or
- DeliveryInfo

The ShortMessage and DeliveryInfo elements are specified in the “Short Message” section in this document.

### 7.2.2 Deliver

The “DeliverRequest” element is used by the message service to deliver a short message or delivery information to the application. It contains the following mandatory element:

- ShortMessage or DeliveryInfo

These are both described in other sections of this document.

### 7.2.3 Expanded Submission Response

The “ExpandedSubmissionResponse” element may be used to deliver a response to a submission request that has been expanded by the message service into multiple submission requests. This may occur when the submission request contains multiple destinations or links to a distribution list.

This element may contain

- zero or more SubmitElementError elements that specify the destinations to which the submission failed and give details of the error.
- Zero or more SubmitElementOK elements providing a message reference and a destination number for each successful submission. The destination number is necessary to qualify the message reference.

#### 7.2.3.1 Example

```

<smap:ExpandedSubmitResponse>
  <smap:SubmissionError>
    <smap:Destination><smap:Number>+4412345</smap:Number></smap:De
    stination>
    <smap:ErrorCode value="23"></smap:ErrorCode>
    </smap:SubmissionError>
    <smap:SubmissionOK>
    <smap:Destination><smap:Number>+4423478</smap:Number></smap:De
    stination>
    <smap:MessageRef>4567FFDEE2</smap:MessageRef>
    </smap:SubmissionOK>
  </smap:ExpandedSubmitResponse>

```

## 7.2.4 Delivery Failure Response

A submit operation to a single subscriber will normally result in a standard SuccessResponse or FailureResponse being returned to the application. In the special case of a submission that involves an immediate message delivery to the subscriber (a “transaction-mode” message), a delivery failure may be returned to the application. This indicates that the message delivery requested in the submission has failed.

The DeliveryErrorResponse element is derived from a standard ErrorResponse datatype and has one extra element:

The DeliveryFailureReason element indicates the reason a short message with Application header parameter of transferMode=“Immediate” failed to be delivered

## 7.3 Message Administration

This module defines the set of elements that allow an application to manage messages while they are in transit within the network. These are

- QueryRequest
- QueryResponse
- CancelRequest
- ReplaceRequest

The elements are described in more detail below:

### 7.3.1 QueryRequest and QueryResponse

QueryRequest is used to query the status of a short message after submission. It contains two child elements:

- **MessageRef**: This identifies the message by the identity supplied by the network on submission
- **Source**: This optional element identifies the original source address supplied in the message submission.

- **Destination:** This optional element identifies the destination address to which the query applies. This is only required when the original submission was to multiple destinations and an unexpanded submit result was returned.

QueryResponse contains the result of a successful query operation. It contains the following child elements:

- **MessageRef:** This identifies the message being queried
- **MessageState:** This specifies the current state of the message. This element is also used in the DeliveryInfo element (section 7.4.3)
- **MessageFinalDate:** If present, This element specifies the date at which the processing of the message by the network was completed and message delivery either succeeded or failed.
- **NetworkDeliveryErrorStatus:** This element is present if message delivery failed and a network-specific error value is available.

### 7.3.1.1 Example

#### Query submission

```
<smap:QueryRequest>
  <smap:MessageRef>456789FFFF</smap:MessageRef>
</smap:QueryRequest>
```

#### Response

```
<smap:QueryResponse>
  <smap:MessageRef>456789FFFF</smap:MessageRef>
  <smap:MessageState value="Delivered"/>
</smap:QueryResponse>
```

### 7.3.2 CancelRequest

This element is used to cancel a short message while it is still in the network. A message can be cancelled either by its service type or by specifying its message identifier explicitly.

#### 7.3.2.1 Example

##### 7.3.2.1.1 Cancel by service type

```
<smap:CancelRequest>
  <smap:CancelByService serviceType="VoiceMailAlert">
    <smap:Source>
      <smap:Number>+44234545</smap:Number>
    </smap:Source>
    <smap:Destination>
      <smap:Number>+445555</smap:Number>
    </smap:Destination>
  </smap:CancelByService>
</smap:CancelRequest>
```

### 7.3.2.1.2 Cancel by Reference

```
<smap:CancelRequest>
  <smap:CancelByReference>
    <smap:Source>
      <smap:Number>12345</smap:Number>
    </smap:Source>
    <smap:MessageRef>4577788FF2D</smap:MessageRef>
  </smap:CancelByReference>
</smap:CancelRequest>
```

## 7.3.3 ReplaceRequest

This element is used to replace a message before the network delivers it. It contains as child elements the identity of the message to be replaced and a ReplaceMessage element.

The ReplaceMessage element has two child elements;

- **ReplaceHeader:** This contains a source address that must match that of the original message and, optionally, a DeliveryInfo element.
- **ReplaceMessage:** This specifies text or binary data to be replaced in the original message.

### 7.3.3.1 Example

```
<smap:ReplaceRequest>
  <smap:MessageRef>458721FFAA1D</smap:MessageRef>
  <smap:ReplaceMessage>
    <smap:ReplaceHeader>
      <smap:Source>
        <smap:Number>12345</smap:Number>
      </smap:Source>
      <smap:DeliveryOptions deliveryReceipt="on"/>
    </smap:ReplaceHeader>
    <smap:ReplaceBody>
      <smap:ReplaceText>replacement message</smap:ReplaceText>
    </smap:ReplaceBody>
  </smap:ReplaceMessage>
</smap:ReplaceRequest>
```

## 7.4 Short Message Module

This module defines a Short Message as a set of XML elements. It can be used to create, manage and transfer short messages independently of any devices or protocols. It requires the addressing elements from the SMEAddress Module.

This module defines two top-level elements

- ShortMessage and
- DeliveryInfo

The “ShortMessage” element is defined as having two child elements; “header” and “body”. These will be considered separately in the sections following.

The “DeliveryInfo” element provides a mechanism for signaling status information related to message delivery. It is described separately in a section following.

### 7.4.1 Header

The header element defines the attributes of a short message. These are

1. One or more destination addresses or a link to a distribution list
2. Source address
3. Delivery options

Each Destination element is defined as an SMAddress entity. The contents of this entity were defined in the previous section.

The DistributionListRef element defines an xml link element that links to a set of destination elements. This link can either identify an element in the same document or be the URL of a different XML document containing the actual distribution list.

The DeliveryOptions element defines the scheduling of message delivery and the status information returned as the message delivery proceeds.

#### 7.4.1.1 DeliveryOptions

This element can have the following attributes:

validityPeriod: This specifies the period during which the message is valid in the network. This value is of XML schema type “xs:duration”.

scheduleDeliveryTime: This specifies a time at which delivery should first be attempted by the network. This value is of XML schema type “xs:dateTime”. Where no timezone is specified the messaging service uses the timezone at its own location. If the application does not know or wishes to be isolated from the location of the messaging service it is recommended that it provides an explicit timezone value or uses UTC time.

deliveryReceipt: This can have one of the following values: “off”, “on”, “FailureOnly”.

#### 7.4.1.2 Example

```
<smap:Header>
  <smap:Destination>
    <smap:Number>123456</smap:Number>
  </smap:Destination>
  <smap:DeliveryOptions
    scheduleDeliveryTime="2001-01-01T09:30:00"
    validityPeriod="P1D"
    deliveryReceipt="on"/>
</smap:Header>
```

## 7.4.2 Body

The body of a message contains one of the following:

1. Text and/or External content
2. XMLContent
3. BinaryData

The Text element contains human-readable text to be sent in the short message. This element has an `xml:lang` attribute that defines the language used in the text. This element is encoded according to the language specified in the `xml:lang` attribute and the capabilities of the destination device.

The XMLContent element is a means of passing XML content directly in a short message. The handling of this element is implementation-specific. It could send the content transparently to the SME (e.g. if the content is WML) or it could transcode it in some form (e.g. extract only body text). The default behaviour is to extract and forward the body text of the XML content.

The BinaryContent element allows the user full control of the data in the short message. This element contains the hexadecimal encoding of the message user data. The user can specify the encoding directly or accept the default encoding of "binary". The user can also specify the language attribute.

The ExternalContent element allows the user to specify a links to external content. It contains one or more reference elements, each with the following attributes:

- `link` specifies the URL of some external content. This may be a reference to another MIME part in a MIME multi part message or it could be the URL of some external text or XML.
- `format` provides a mechanism for specifying the format of the external content. It can have one of the following values: "Bitmap\_BW", "Bitmap\_greyscale", "Bitmap\_64colour", "Ringtone\_iMelody", "Ringtone\_SmartMessaging", "GIF", "PNC", "other".
- `BitmapWidth` specifies in pixels the width of a supplied bitmap image
- `BitmapHeight` specifies in pixels the height of a supplied bitmap image.

Where the external content is a MiME multipart message, The ExternalContent element will have a “link” attribute containing a URL of the format [cid://content-identifier](#) where “content-identifier” is the identity of the MiME part. This MiME part contains the actual encapsulated MiME message that is delivered to the user

#### 7.4.2.1 Examples

Text

```
<smap:Body>
  <smap:Text>hello world</smap:Text>
</smap:Body>
```

XML Content

```
<smap:Body>
  <smap:XMLContent>
    <myMarkup>
      <test>helllo hello</test>
    </myMarkup>
  </smap:XMLContent>
</smap:Body>
```

External Content

```
<smap:Body>
  <smap:ExternalContent link=http://www.aaa.bbb/xxxxx/>
</smap:Body>
```

Binary Content

```
<smap:Body>
  <smap:BinaryData>23FF457F</smap:BinaryData>
</smap:Body>
```

Text + Binary Content (iMelody ringtone)

See section 7.5.4.2 below (final example).

#### 7.4.3 Delivery Info

This element contains the information supplied as part of a delivery receipt, device or user delivery acknowledgement or an intermediate delivery notification. The “type” attribute of the element defines which of these is being provided.

This body of this element contains the following:

- Destination
- Source

- MessageInfo
- Text

Of these, only MessageInfo is a new element, the others have all been described in the previous section. It provides information about the message through the following elements:

**MessageRef** identifies the message through an SMSC-supplied identifier in its body. It may also supply the original user message reference as an optional attribute.

**MessageState**: identifies the current state of the message. This empty element has a single attribute "Value". This attribute can be set to one of "Enroute", "Delivered", "Expired", "Deleted", "Accepted", "Unknown" or "Rejected".

**NetworkErrorCode** optionally provides network-specific error information.

### 7.4.3.1 Example

```
<smap:DeliveryInformation>
  <smap:Destination>
    <smap:Number>12345</smap:Number>
  </smap:Destination>
  <smap:Source>
    <smap:Number>5555</smap:Number>
  </smap:Source>
  <smap:MessageInfo>
    <smap:MessageRef>452377FFAA</smap:MessageRef>
    <smap:MessageState value="Deleted"/>
    <smap:NetworkErrorCode networkType="ANSI-136"
value="345"/>
  </smap:MessageInfo>
  <smap:Text>standard delivery info text</smap:Text>
</smap:DeliveryInformation>
```

## 7.5 Message Extensions Module

The standard ShortMessage module contains of the possible message header and body parameters. The message extensions module provides a set of XML schema definitions that extend the type definitions for Header and Body information.

The header data type is extended with the following extra elements:

- NetworkHandling: This element specifies any other network-specific handling of the message.
- ApplicationOptions: This element specifies any end-to-end application information
- DeviceControl: This element specifies any information directed to the device receiving the message.

The body type is extended to support User Data Headers:

### 7.5.1 NetworkHandling Element

This element contains a set of information that supports the handling of a message in the network. It has the following attributes:

**serviceType:** This identifies the service provided by the application from a set of the following; "NULL", "CellularMessaging", "CellularPaging", "VoiceMailNotify", "VoiceMailAlert", "WAP", "WAP\_WDP", "WAP\_WCMP", "USSD" or "Custom". If the service is specified as "custom", the attribute "customServiceName" contains the actual service name.

**storageMode:** This specifies whether or not more than one message with the source and destination address should be stored on the network when the messages have the same value for their ServiceType parameter. Possible values are "LatestOnly" and "StoreAll".

**transferMode:** This specifies how the application would like the message to be transferred through the network. The allowed values are: "Immediate", "Datagram" and "StoreAndForward". A message with a transferMode of "immediate" is delivered directly to the subscriber as part of the submission procedure. A message with a transferMode of "Datagram"

**QOSTimeToLive:** This specifies the lifetime in seconds that the application requests the system to keep the message if it cannot be delivered immediately.

Example

```
<smap:Header xsi:type="smap:ExtendedHeaderType">
  <smap:Destination>
    <smap:Number>+44123456</smap:Number>
  </smap:Destination>
  <smap:NetworkHandling
serviceType="CellularPaging"
storageMode="LatestOnly"
transferMode="StoreAndForward"
QOSTimeToLive="1000"/>
</smap:Header>
```

## 7.5.2 ApplicationOptions element

This element contains a set of information that supports end-to-end operation of specific application protocols. These protocols are specific to individual networks and handsets and applications should use them with caution. This element can contain the following optional elements in sequence:

**NumberOfVoiceMails:** This specifies to the number of messages stored in a mailbox.

**Callback:** This specifies a callback number associated with the message. A number of callback elements can be specified within one ApplicationOptions element. This corresponds to a message having multiple callback numbers.

**ITSSession:** This provides the information an ITS application needs to maintain a session with an ITS application on a mobile device.

**ITSReplyType:** This element specifies the ITS reply type to be used by the receiving mobile application in generating a response to this message.

These will now be defined in more detail.

### 7.5.2.1 NumberOfVoiceMails

This element indicates the number of messages in the subscriber's voice mailbox. This element is empty and has one mandatory "value" attribute which contains a number in the range 0-99.

### 7.5.2.2 Callback

A callback element has the following attributes:

- **presentation**: This optional attribute specifies how the callback number is presented to the user on the device. It can have one of the following values: "Allowed", "Restricted" or "NumberNotAvailable".
- **screening**: This optional attribute specifies what screening the network has performed on the callback number. It has one of the following values: "UserProvided-Unscreened", "UserProvided-Verified", "UserProvided-VerificationFailed", "NetworkProvided".
- **numberEncoding**: This mandatory attribute specifies how the callback number is encoded. The possible values are: "DTMF" or "ASCII".

A callback element has the following child elements:

**Number**: This is a mandatory text element containing the callback number.

**Display**: This optional element contains information to be displayed to the user on the device. It has one of the following formats:

```
<Display><Text>display text</Text></Display>
```

```
<Display><BinaryData>112277FFAA55</BinaryData></Text>
```

### 7.5.2.3 ITSSession

The ITSSession element provides application information for a CDMA Interactive Teleservice as defined by the Korean PCS carriers [KORITS]. This empty element has the following attributes:

- **sessionNumber**: This mandatory attribute identifies the session between the application and the mobile device. It is a numeric value in the range 0-255.
- **sequenceNumber**: This mandatory attribute identifies the position of the message in a dialog with the mobile device. It is a numeric value in the range 0-127.
- **finalMessage**: This attribute specifies whether or not the message is the final one of the session. It has a value of "TRUE" or "FALSE". The default value is "FALSE".

### 7.5.2.4 ITSReply

The ITSReply element provides response information for a CDMA Interactive Teleservice as defined by the Korean PCS carriers [KORITS]. This empty element has one mandatory attribute: "value". This attribute has a number in the range 0-8, where the selected number specifies the type of ITS content in the message.

### 7.5.2.5 Example

```
<smap:Header xsi:type="smap:ExtendedHeaderType">
  <smap:Destination>
    <smap:Number>+44123456</smap:Number>
  </smap:Destination>
  <smap:ApplicationOptions>
    <smap:Callback numberEncoding="DTMF">
      <smap:Number>12345</smap:Number>
    </smap:Callback>
    <smap:ITSSession sessionNumber="5" sequenceNumber="3"/>
  </smap:ApplicationOptions>
</smap:Header>
```

### 7.5.3 DeviceControl element

The DeviceControl element specifies information directed to the device receiving the message. This information is generally particular to specific technology types and application developers should use these features with care. This element has two optional attributes:

- **deviceAckReq:** This specifies whether or not the device should acknowledge receipt of the message. It can have values "on" or "off". Note: This is a specialised form of delivery receipting only available in TDMA and CDMA networks.
- **userAckReq:** This specifies whether or not the device should prompt the user to acknowledge receipt of the message. It can have values "on" or "off". Note: This is a specialised form of delivery receipting only available in TDMA and CDMA networks.

This element can contain one each of the following child elements (in any order):

- **GSMOptions:** This can specify either GSM Message type 0-7 or full GSM Protocol Identifier(see [GSM 03.40]. It can also specify that the replyPath parameter should be set by the network.
- **displayTime:** This specifies a display time for the message to the device.
- **MSValidity:** This provides the device with validity information related to storing the message on the device.
- **MessageWaitInd:** This provides a mechanism whereby the sender can change the message waiting indicator on the mobile device when the message is delivered.
- **ReceiveAlert:** This specifies whether a custom alert should be generated on the mobile device when the message is delivered and can also specify the tone of the alert.

#### 7.5.3.1 GSMOptions

This element has one attribute: replyPath. This can have a value of "Set" or "NotSet". The default value is "NotSet".

This element can have one only of the following child elements:

- **messageType**. This specifies the message type component of a GSM Protocol Identifier. A GSM device uses this information to perform a replace on handset operation (i.e. if there is a stored message of this type, replace it with the new message). This element has one mandatory "value" attribute that accepts numbers in the range 0-7.
- **protocolId**: This allows the application to specify the GSM protocol identifier completely according to [GSM 03.40]. It has one mandatory "value" attribute. This is a hex number comprised of two hex digits.

### 7.5.3.2 DisplayTime

This empty element has one mandatory "value" attribute. It takes a value of "0", "1" or "2". This corresponds to display values of "temporary", "default" and "invoke" respectively.

### 7.5.3.3 MSValidity

This element controls the storage of the message on the mobile device. This element is empty and has one mandatory "value" attribute. It can have one of the following values : "Indefinite", "PowerDown", "SIDRegistrationArea" or "DisplayOnly".

### 7.5.3.4 MessageWaitInd

This element allows the application to control an indication on the mobile device that there are messages waiting for the user. This element is empty and has two attributes:

- **messageType**: this mandatory attribute takes one of the following values: "VoiceMail", "Fax", "eMail", "Other".
- **setting**: This attribute can have one of the following values: "Active" or "Inactive". The default value is "Active".

### 7.5.3.5 ReceiveAlert

This presence of element indicates to the mobile device that should alert the user in some fashion on message arrival. The optional "tone" attribute contains a numeric value that specifies an alert tone on a TDMA network (see [CMT-136]).

### 7.5.3.6 Example

```
<smap:Header xsi:type="smap:ExtendedHeaderType">
  <smap:Destination>
    <smap:Number>+44123456</smap:Number>
  </smap:Destination>
  <smap:DeviceControl>
    <smap:GSMOptions replyPath="NotSet">
      <smap:MessageType value="3"/>
    </smap:GSMOptions>
    <smap:MessageWaitInd messageType="VoiceMail"
setting="Inactive"/>
  </smap:DeviceControl>
</smap:Header>
```

## 7.5.4 UserDataHeader element

A message Body element can be extended to contain a UserDataHeader element. This allows the application to specify control information for such features as ringtones and operator logos. The UserDataHeader element either contains a single EncodedUDH element or a sequence containing a number of HexInfoElement and StringInfoElement elements in any order.

An EncodedUDH element contains a single fully encoded user data header in Hex format.

HexInfoElement and StringInfoElement contain hex or string data that the message service must encode to generate a user data header. They both have a single mandatory parameter:

**ident:** This specifies the Information Element Identifier (IEI) of the UDH element in Hex. E.g. an ident of "0C" identifies an EMS iMelody information element.

#### 7.5.4.1 Implementation details

User data header implementation has a number of constraints which must be considered by application developers in choosing how information is encoded. This can be summarized as follows: The UDH contributes to the overall size of a message so developers should calculate carefully the full size of user data including the encoded header and other text.

- b) Message concatenation uses the user data header as well. Consequently, large user data headers cannot be guaranteed to work in a consistent fashion across all messaging services.
- c) Extended objects are not supported directly by SMAP. Developers can either
  - a. use the "externalContent" element to pass images or ringtones to the messaging service for encoding or
  - b. perform their own encoding and segmentation of UDH extended objects before transmission.

#### 7.5.4.2 Examples

The following example shows how an EMS iMelody information element is encoded explicitly (note: the melody is absent):

```
<smap:Body xsi:type="ExtendedBodyType">
  <smap:Text>This is an EMS iMelody ringtone</smap:Text>
  <smap:UserDataHeader>
    <smap:StringInfoElement ident="0C" >BEGIN:iMELODY
VERSION:1.2
FORMAT:CLASS1.0
MELODY:melody goes here
END:IMELODY</smap:StringInfoElement>
  </smap:UserDataHeader>
</smap:Body>
```

The next example shows how a Nokia smart messaging operator logo can be encoded. using an EncodedUDH element.

```

<smap:Body xsi:type="ExtendedBodyType">
  <BinaryData>42F45000480E01FFFFFFFFFFFFFFFFFFFFFFFF00000000
0000000000FFFFFFFFFFFFFFFFFFFFFFFF0000000000000000000010F00000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
000000001</BinaryData>
  <smap:UserDataHeader>
    <smap:EncodedUDH>050415820000</smap:EncodedUDH>
  </smap:UserDataHeader>
</smap:Body>

```

The next example shows how an iMelody ringtone can be passed transparently to the messaging service for encoding. The messaging service takes care of generating the UDH.

```

      <smap:Body>b
        <smap:Text>see this ringtone</smap:Text>
        <smap:ExternalContent type=iMelody
          link=cid:tesattatat />
      </smap:Body>
    </smap:ShortMessage>
  </smap:SubmitRequest>
</soap:Body>
</soap:Envelope>
--MIME_boundary
Content-Type: audio/imelody; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <tesattatat>
BEGIN:IMELODY
VERSION:1.2
FORMAT:CLASS1.0
MELODY:
  melody goes here
END:IMELODY

```

## 7.6 Address Module

This module defines the addressing formats used to access different Messaging Entities. The main address datatype "SMAddressType" can have one of a number of child elements:

- Number
- ESMAApplication
- IPAddress
- EMail Address

These will each be considered in turn

## 7.6.1 Number

The element “Number” contains text specifying the address in the form of a telephone number.

Two attributes can be specified “TON” and “NPI”. These can have the following values:

TON: “Unknown” “International” “National” “Network” “Subscriber” “Alphanumeric” “Abbreviated”

NPI: “Unknown” “ISDN” “Data” “Telex” “LandMobile” “National” “Private” “ERMES” “WAPClient”

When both attributes are absent the number is taken to be a mobile ISDN number in either national or international format. If the first character of the text is a “+”, the number is taken to be an international number (i.e. “+CCCN” where CCC is the countrycode) and is given a TON of “International” and an NPI of “ISDN”. Otherwise the number is taken to be a national number to be interpreted according to national numbering scheme used by the Short Message Service. In the latter case it is given a TON of “National” and an NPI of “Unknown”.

### 7.6.1.1 Examples

```
<Number>+3531234567</Number>
<Number TON="National" NPI="ISDN" >771234567</Number>
<Number TON="Alphanumeric">Voicemail</Number>
```

## 7.6.2 Application

The application element defines the address of an application.

This can be either an externally-visible shortcode or an actual application name. When this element is specified as a short code, The gateway is responsible for selecting the appropriate TON and NPI values. Where the application is given as a name, the message service is presumed to have a standard mechanism for mapping this name onto a shortcode.

*In the future additional elements may be added to support applications with different addresses in different networks and features such as gateway-assigned addressing (e.g. the application does not care what its address is so long as it is assigned to it for the duration of a bind)*

## 7.6.3 IP

The IP element defines the IP address of a mobile device. This simple type is a union of the following types: IPV4AddressType and IPV6AddressType. The body contains the IP address in either “aaa.bbb.ccc.ddd” notation or in IPV6 format “A:B:C:D:E:F:G:H”. The only optional attribute is a port number.

## 7.6.4 EMail

The EMail element defines an RFC2822 address. This simple type is a currently defined as an NMTOKEN. This may be further constrained in the future to perform more validity-checking on the address format.

### 7.6.5 Simple Addressing Examples

```

<smap:Destination>
  <smap:Number>12345</smap:Number>
</smap:Destination>
<smap:Destination>
  <smap:IP port="34">23.56.89.58</smap:IP>
</smap:Destination>
<smap:Destination>
  <smap:IP>45:68:97:56:33:67:66:33</smap:IP>
</smap:Destination>
<smap:Destination>
  <smap:Application>TESTAPP</smap:Application>
</smap:Destination>
<smap:Destination>
  <smap:Application
xsi:type="smap:ApplicationShortCodeType">12345</smap:Application>
</smap:Destination>
<smap:Destination>
  <smap:EMail>test@smsforum.net</smap:EMail>
</smap:Destination>

```

### 7.6.6 Extended Addressing

The base SMAddressType can be extended for full control of addressing. The new ExtendedAddress Type has the following extra attributes:

- **networkType**: This attribute can have one of the following values: "Unknown" (default), "GSM", "TDMA", "CDMA", "PDC", "PHS", "iDEN", "AMPS", "Paging".
- **subUnit**: This attribute identifies a sub unit within the device being addressed. It can have one of the following values: "Unknown" (default), "Display", "Equipment", "SmartCard", "ExternalUnit".

An element of this type can have the following extra parameters: Bearer and SubAddress. These are defined in the sections following:

#### 7.6.6.1 Bearer

This element specifies the bearer network to be used in accessing the device. This element is empty and has the following attributes:

**Type**: This identifies the type of the bearer network. It can have one of the following values: "Unknown" (default), "SMS", "CSD", "Packet", "USSD", "CDPD", "DataTAC", "FLEX", "CellBroadcast".

**TelematicsId**: This identifies the telematics interworking to be used by the specified bearer network. This is an integer value.

#### 7.6.6.2 SubAddress

This element contains an X.213 subaddress (NSAP) or else a user-specified sub address. The body of the element is a set of hex digits and it has one attribute: addressType that can have a value of "NSAP" (default) or "UserSpecified".

### 7.6.6.3 Examples

```
<smap:Destination xsi:type="smap:ExtendedAddressType">
  <smap:Number>12345</smap:Number>
  <smap:Bearer telematicsId="34" type="CSD"/>
</smap:Destination>

<smap:Destination xsi:type="smap:ExtendedAddressType"
networkType="GSM" subUnit="SmartCard">
  <smap:Number TON="Network" NPI="Telex">12345</smap:Number>
</smap:Destination>

<smap:Destination xsi:type="smap:ExtendedAddressType">
  <smap:Number>234545</smap:Number>
  <smap:SubAddress
addressType="NSAP">34AD23FFFF</smap:SubAddress>
</smap:Destination>
```

## 7.7 Protocol Options

The following features of SMAP are defined as optional and can be requested explicitly as MMAP options as part of session establishment:

**ExtendedHeaders:** By default the application does not submit messages with extended header information and messaging services do not deliver messages with this information included. Specifying this option enables message submission and delivery with extended header information.

**ExtendedAddressing:** By default, the application does not submit messages with ExtendedAddressType destinations and does not expect delivered messages to contain this information. Specifying this option enables message submission and delivery with extended addressing information.

SMAP defines the following attributes for use in the MMAP parameter element.

**expandedSubmitResponse:** This attribute can have a value of "true" or "false" and specifies whether the response to a message submission will contain multiple message identifiers when multiple destinations are specified or will only contain one message identity.

**maxMessageSize:** This attribute specifies in bytes the maximum message size for the session.

## 7.8 Protocol Profiles

The following MMAP Profiles are defined for SMAP

**SMAPBasic.xml:**

This xml file contains the following MMAP profile definition

```
<smap:SMAPFeatures
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.0
http://www.smsforum.net/schemas/smap/v1.0/SMAPFeatures.xsd">
  <smap:RequiredOperations>SubmitRequest</smap:RequiredOperations>
  <smap:SupportedOperations>SubmitRequest
DeliverRequest</smap:SupportedOperations>
  <smap:Options/>
  <smap:Parameters maxMessageSize="140"
expandedSubmitResponse="true"/>
</smap:SMAPFeatures>
```

It specifies that applications and message services obeying the profile must support submit and deliver operations with a maximum message size of 140 octets and no optional features enabled.

## 8 End-to-End Examples

### 8.1 Binding to an SMAP service

The following example shows an SMAP client application binds to an SMAP message service over SOAP.

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="Request"
sourceOperationReference="123"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:BindRequest
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:SessionControl maxOutstandingRequests="3"
sessionType="client"/>
      <MMAP:FeatureGroup
href="http://www.smsforum.net/schemas/smap/v1.0/SMAPBasic.xml"/>
    </MMAP:BindRequest>
  </soap:Body>
</soap:Envelope>
```

The SMAP server returns the following bind response to the SMAP application:

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="SuccessResponse"
sourceOperationReference="123" sessionId="799"/>
      <MMAP:ServiceContext serviceName="test"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:BindResponse
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:SessionControl maxOutstandingRequests="1"
></MMAP:SessionControl>
      <MMAP:FeatureSet>

        <MMAP:SupportedOperations>SubmitRequest</MMAP:SupportedOperati
ons>
      </MMAP:FeatureSet>
    </MMAP:BindResponse>
  </soap:Body>
</soap:Envelope>
```

## 8.2 Submitting a Message

Using the session established in the previous example, the application can now submit messages to the message service: A simple submission request can be encoded as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="Request"
sourceOperationReference="124" sessionId="799"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <smap:SubmitRequest
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.0
http://www.smsforum.net/schemas/smap/v1.0/MessageTransfer.xsd">
      <smap:ShortMessage>
        <smap:Header>
          <smap:Destination>
            <smap:Number>+44123456</smap:Number>
          </smap:Destination>
        </smap:Header>
        <smap:Body>
          <smap:Text>hello world</smap:Text>
        </smap:Body>
      </smap:ShortMessage>
    </smap:SubmitRequest>
  </soap:Body>
</soap:Envelope>
```

The message service will then return a submission response as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="SuccessResponse"
sourceOperationReference="124" sessionId="799"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <smap:SubmitResponse
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/smap/v1.00
http://www.smsforum.net/schemas/smap/v1.0/MessageTransfer.xsd">
      <smap:MessageRef>4A5678ABFF88</smap:MessageRef>
    </smap:SubmitResponse>
  </soap:Body>
</soap:Envelope>
```

### 8.3 Unbinding from the SMAP service

To unbind, the application sends an MMAP UnbindRequest as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="Request"
sourceOperationReference="125" sessionId="799"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:UnbindRequest
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd"/>
  </soap:Body>
</soap:Envelope>
```

The message service will then return an unbind response as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <MMAP:MMAPHeader soap:mustUnderstand="1"
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd">
      <MMAP:ApplicationContext bodyType="SuccessResponse"
sourceOperationReference="125" sessionId="799"/>
    </MMAP:MMAPHeader>
  </soap:Header>
  <soap:Body>
    <MMAP:UnbindResponse
xmlns:MMAP="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0
http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd"/>
  </soap:Body>
</soap:Envelope>
```







## 9 Schema Specification

### 9.1 MMAP Modules

#### 9.1.1 mmap.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include
schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0/service
.xsd"/>
  <xs:import
namespace="http://schemas.xmlsoap.org/soap/envelope/"
schemaLocation="http://schemas.xmlsoap.org/soap/envelope/">
    <xs:element name="MMAPHeader">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ApplicationContext"
type="ApplicationContextType"/>
          <xs:element name="ServiceContext"
type="ServiceContextType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute ref="soap:mustUnderstand"
use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="ApplicationContextType">
      <xs:attribute name="sourceOperationReference"
type="xs:unsignedInt"/>
      <xs:attribute name="sessionId" type="xs:NMTOKEN"/>
      <xs:attribute name="bodyType" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="Request"/>
            <xs:enumeration
value="SuccessResponse"/>
            <xs:enumeration
value="ErrorResponse"/>
            <xs:enumeration
value="RequestAccepted"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
    <!--
The elements following define the SOAP body for MMAP
operations. They may be used by other protocols but are not
necessary, as the MMAP header defines the type of the element.
-->
    <xs:element name="Request" type="RequestType"/>
    <xs:element name="Response" type="ResponseType"/>
    <xs:element name="SuccessResponse" type="SuccessResponseType"
substitutionGroup="Response"/>
    <xs:element name="BindResponse" type="BindResponseType"/>

```

```

<xs:element name="BatchResponse" type="BatchResponseType"/>
<xs:element name="UnbindResponse" type="SuccessResponseType"/>
<xs:element name="EnquireLinkResponse"
type="SuccessResponseType"/>
<xs:element name="ErrorResponse" type="ErrorResponseType"/>
<xs:complexType name="RequestType" abstract="true"/>
<xs:complexType name="ResponseType" abstract="true"/>
<xs:complexType name="SuccessResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseType"/>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ErrorResponseType">
  <xs:complexContent>
    <xs:extension base="ResponseType">
      <xs:sequence>
        <xs:element name="ErrorCode"
type="ErrorCodeType"/>
        <xs:element ref="ErrorDescription"
minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="ErrorDescription" type="xs:string"/>
<xs:complexType name="ErrorCodeType">
  <xs:attribute name="value" type="xs:QName"
use="required"/>
</xs:complexType>
<xs:element name="BindRequest" type="BindRequestType"/>
<xs:element name="BindBackRequest"
type="BindBackRequestType"/>
<xs:element name="UnbindRequest" type="UnbindRequestType"/>
<xs:element name="EnquireLinkRequest"
type="EnquireLinkRequestType"/>
<xs:element name="BatchRequest" type="BatchRequestType"/>
<xs:complexType name="BindRequestType">
  <xs:complexContent>
    <xs:extension base="RequestType">
      <xs:sequence>
        <xs:element name="SessionControl"
type="SessionControlType"/>
        <xs:group ref="FeatureGroup"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SessionControlType">
  <xs:sequence>
    <xs:element name="ReturnSession"
type="ReturnSessionInformationType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="sessionType" default="client">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="client"/>
        <xs:enumeration value="peer"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="maxOutstandingRequests"
type="xs:unsignedInt"/>
</xs:complexType>
<xs:group name="FeatureGroup">
  <xs:choice>
    <xs:element ref="FeatureSet"/>
  </xs:choice>
</xs:group>

```

```

        <xs:element name="FeatureProfile">
            <xs:complexType>
                <xs:attribute name="href"
type="xs:anyURI"/>
            </xs:complexType>
        </xs:element>
    </xs:choice>
</xs:group>
<xs:element name="FeatureSet" type="FeatureSetType"/>
<xs:complexType name="FeatureSetType">
    <xs:all>
        <xs:element name="RequiredOperations"
type="FeatureListType" minOccurs="0"/>
        <xs:element name="SupportedOperations"
type="FeatureListType" minOccurs="0"/>
        <xs:element name="Options" type="FeatureListType"
minOccurs="0"/>
        <xs:element name="Parameters" minOccurs="0">
            <xs:complexType>
                <xs:anyAttribute namespace="##any"
processContents="lax"/>
            </xs:complexType>
        </xs:element>
        <xs:element name="ApplicationSettings">
            <xs:complexType>
                <xs:sequence>
                    <xs:any namespace="##any"
processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:anyAttribute namespace="##any"
processContents="lax"/>
            </xs:complexType>
        </xs:element>
    </xs:all>
    <xs:attribute name="namespace" type="xs:anyURI"/>
    <xs:attribute name="location" type="xs:anyURI"/>
</xs:complexType>
<xs:simpleType name="FeatureListType">
    <xs:list itemType="xs:NMTOKEN"/>
</xs:simpleType>
<xs:complexType name="ReturnSessionInformationType">
    <xs:sequence>
        <xs:element name="SessionReturnPath">
            <xs:complexType>
                <xs:attribute name="link"
type="xs:anyURI" use="required"/>
            </xs:complexType>
        </xs:element>
        <xs:element name="ServiceContext"
type="ServiceContextType" minOccurs="0"/>
        <xs:element name="Parameters"
type="ApplicationSuppliedParamsType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="BindBackRequestType">
    <xs:complexContent>
        <xs:extension base="RequestType">
            <xs:sequence>
                <xs:element name="Parameters"
type="ApplicationSuppliedParamsType"/>
                <xs:element name="SessionControl">
                    <xs:complexType>
                        <xs:attribute
name="maxOutstandingRequests" type="xs:unsignedInt"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

                <xs:element ref="FeatureSet"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="ApplicationSuppliedParamsType"
mixed="true">
    <xs:sequence maxOccurs="unbounded">
        <xs:any processContents="skip"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="UnbindRequestType">
    <xs:complexContent>
        <xs:extension base="RequestType"/>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="EnquireLinkRequestType">
    <xs:complexContent>
        <xs:extension base="RequestType"/>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="BindResponseType">
    <xs:complexContent>
        <xs:extension base="SuccessResponseType">
            <xs:sequence>
                <xs:element name="SessionControl">
                    <xs:complexType>
                        <xs:attribute
name="maxOutstandingRequests" type="xs:unsignedInt"/>
                    </xs:complexType>
                </xs:element>
                <xs:element ref="FeatureSet"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="BatchRequestType">
    <xs:complexContent>
        <xs:extension base="RequestType">
            <xs:sequence>
                <xs:element ref="FeatureSet"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="Request"
type="BatchOperationType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="batchRequestReference"
type="xs:token" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="BatchResponseType">
    <xs:complexContent>
        <xs:extension base="SuccessResponseType">
            <xs:sequence minOccurs="0"
maxOccurs="unbounded">
                <xs:choice>
                    <xs:element
name="RequestSucceeded" type="BatchOperationType" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element name="RequestFailed"
type="BatchOperationType" minOccurs="0" maxOccurs="unbounded"/>
                </xs:choice>
            </xs:sequence>
            <xs:attribute name="batchRequestReference"
type="xs:token" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="BatchOperationType" mixed="true">
    <xs:sequence maxOccurs="unbounded">
        <xs:any processContents="skip"/>
    </xs:sequence>
    <xs:attribute name="batchOperationReference"
type="xs:unsignedInt" use="required"/>
</xs:complexType>
</xs:schema>

```

### 9.1.2 service.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.smsforum.net/schemas/mmap/v1.0"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:complexType name="ServiceContextType">
        <xs:sequence>
            <xs:element name="AccessControl"
type="AccessControlType" minOccurs="0"/>
            <xs:element name="Billing"
type="GenericBillingType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="serviceName" type="xs:token"
use="required"/>
    </xs:complexType>
    <xs:complexType name="AccessControlType">
        <xs:sequence>
            <xs:element ref="ApplicationIdentity"/>
            <xs:element name="Authentication"
type="AuthenticationType" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ApplicationIdentity" type="xs:NMTOKEN"/>
    <xs:complexType name="AuthenticationType">
        <xs:sequence>
            <xs:element ref="Password"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="Password" type="xs:string"/>
    <xs:complexType name="GenericBillingType">
        <xs:sequence>
            <xs:any namespace="##any" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="sourceReference" type="xs:NMTOKEN"
use="required"/>
        <xs:anyAttribute namespace="##any"
processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="BillingType">
        <xs:sequence>
            <xs:element name="Cost" type="CostType"
minOccurs="0"/>
            <xs:element name="ApplyTo"
type="BillingDetailsType" minOccurs="0"/>

```

```

    <xs:element name="Description" type="xs:string"
minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="sourceReference" type="xs:NMTOKEN"
use="required"/>
</xs:complexType>
<xs:complexType name="BillingDetailsType">
  <xs:sequence>
    <xs:element name="Name" type="xs:NMTOKEN"
minOccurs="0"/>
    <xs:element name="Account" type="xs:NMTOKEN"
minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="billedEntity"
default="externalUser">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="externalUser"/>
        <xs:enumeration value="subscription"/>
        <xs:enumeration value="service"/>
        <xs:enumeration value="other"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
<xs:complexType name="CostType">
  <xs:choice>
    <xs:element name="CostClass"
type="CostClassType"/>
    <xs:element name="CostAmount"
type="CostAmountType"/>
    <xs:element name="CustomCost"
type="CustomCostType"/>
  </xs:choice>
  <xs:attribute name="type" default="debit">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="debit"/>
        <xs:enumeration value="credit"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
<xs:complexType name="CostAmountType">
  <xs:attribute name="units" type="xs:string"/>
  <xs:attribute name="amount" type="xs:string"/>
</xs:complexType>
<xs:complexType name="CostClassType">
  <xs:attribute name="type" default="standard">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="Free"/>
        <xs:enumeration value="Standard"/>
        <xs:enumeration value="Premium"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
<xs:complexType name="CustomCostType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any"
processContents="lax"/>

```

```
</xs:complexType>
</xs:schema>
```

## 9.2 SMAP modules

### 9.2.1 smap.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/smap/v1.0"
xmlns="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/Message
Transfer.xsd"/>
  <xs:include
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/Message
Admin.xsd"/>
  <xs:include
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/Message
Extensions.xsd"/>
</xs:schema>
```

### 9.2.2 ShortMessage.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/smap/v1.0"
xmlns="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xlink="http://www.w3.org/1999/xlink"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/Address
ing.xsd"/>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>
  <xs:import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/xlink.x
sd"/>
  <xs:element name="ShortMessage">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header"
type="MessageHeaderType"/>
        <xs:element name="Body"
type="MessageBodyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DeliveryInformation"
type="DeliveryInfoType"/>
  <xs:complexType name="DeliveryInfoType">
    <xs:sequence>
      <xs:element name="Destination"
type="SMAddressType"/>
      <xs:element name="Source" type="SMAddressType"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

        <xs:element name="MessageInfo"
type="MessageInfoType"/>
        <xs:element name="Text" type="MessageTextType"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MessageHeaderType">
    <xs:sequence>
        <xs:group ref="DestinationGroup"/>
        <xs:element name="Source" type="SMAddressType"
minOccurs="0"/>
        <xs:element name="MessageIdentity"
type="MessageRefType" minOccurs="0"/>
        <xs:element name="SecurityOptions"
type="SecurityInfoType" minOccurs="0"/>
        <xs:element name="Deliveryoptions"
type="DeliveryOptionsType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="priority" type="PriorityType"/>
    <xs:attribute name="subject" type="xs:string"/>
</xs:complexType>
<xs:complexType name="DeliveryOptionsType">
    <xs:attribute name="validityPeriod" type="xs:duration"/>
    <xs:attribute name="scheduleDeliveryTime"
type="xs:dateTime"/>
    <xs:attribute name="deliveryReceipt" default="off">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="off"/>
                <xs:enumeration value="on"/>
                <xs:enumeration value="FailureOnly"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="MessageBodyType">
    <xs:sequence>
        <xs:choice>
            <xs:sequence>
                <xs:element name="Text"
type="MessageTextType" minOccurs="0"/>
                <xs:element name="ExternalContent"
type="ExternalContentType" minOccurs="0"/>
            </xs:sequence>
            <xs:element name="BinaryData"
type="BinaryDataType"/>
            <xs:element ref="XMLContent"/>
        </xs:choice>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MessageTextType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute ref="xml:lang"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:element name="XMLContent">
    <xs:complexType>
        <xs:sequence>
            <xs:any/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="ExternalContentType">
    <xs:sequence>
        <xs:element name="Reference"
type="ContentReferenceType" minOccurs="0" maxOccurs="unbounded"/>

```

```

        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ContentReferenceType">
        <xs:attribute ref="xlink:href"/>
        <xs:attribute name="format" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="Bitmap_Bw"/>
                    <xs:enumeration
value="Bitmap_greyscale"/>
                    <xs:enumeration
value="Bitmap_64colour"/>
                    <xs:enumeration
value="Ringtone_imelody"/>
                    <xs:enumeration
value="Ringtone_SmartMessaging"/>
                    <xs:enumeration value="GIF"/>
                    <xs:enumeration value="PNG"/>
                    <xs:enumeration value="other"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="bitmapwidth" use="optional"
type="xs:integer" />
        <xs:attribute name="bitmapheight" use="optional"
type="xs:integer"/>
    </xs:complexType>
    <xs:complexType name="BinaryDataType">
        <xs:simpleContent>
            <xs:extension base="xs:hexBinary">
                <xs:attribute ref="xml:lang"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="MessageInfoType">
        <xs:sequence>
            <xs:element name="MessageRef"
type="MessageRefType"/>
            <xs:element name="MessageState"
type="MessageStateType"/>
            <xs:element name="NetworkErrorCode"
type="NetworkErrorCodeType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="priority" type="PriorityType"/>
    </xs:complexType>
    <xs:simpleType name="PriorityType">
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="3"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="SecurityInfoType">
        <xs:attribute name="privacyLevel"
default="Unrestricted">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="Unrestricted"/>
                    <xs:enumeration value="Restricted"/>
                    <xs:enumeration value="Confidential"/>
                    <xs:enumeration value="Secret"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="MessageRefType">
        <xs:simpleContent>
            <xs:extension base="xs:token">

```

```

        <xs:attribute name="userMsgRef"
type="xs:string"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="MessageStateType">
    <xs:attribute name="value" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="Enroute"/>
                <xs:enumeration value="Delivered"/>
                <xs:enumeration value="Expired"/>
                <xs:enumeration value="Deleted"/>
                <xs:enumeration value="Accepted"/>
                <xs:enumeration value="Unknown"/>
                <xs:enumeration value="Rejected"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="NetworkErrorCodeType">
    <xs:attribute name="networkType" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="ANSI-136"/>
                <xs:enumeration value="IS-95"/>
                <xs:enumeration value="GSM"/>
                <xs:enumeration value="reserved"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="value" type="xs:integer"
use="required"/>
</xs:complexType>
</xs:schema>

```

### 9.2.3 MessageTransfer.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/smap/v1.0"
xmlns="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:include schemaLocation="ShortMessage.xsd"/>
    <xs:element name="SubmitRequest" type="SubmitRequestType"/>
    <xs:element name="DeliverRequest" type="DeliverRequestType"/>
    <xs:element name="SuccessResponse">
        <xs:complexType/>
    </xs:element>
    <xs:element name="ErrorResponse" type="ErrorResponseType"/>
    <xs:element name="ExpandedSubmitResponse"
type="ExpandedSubmissionResponseType"/>
    <xs:element name="DeliveryErrorResponse"
type="DeliveryErrorResponseType"/>
    <xs:complexType name="SubmitRequestType">
        <xs:sequence>
            <xs:element name="Billing"
type="ExternalBillingType" minOccurs="0"/>
            <xs:choice>
                <xs:element ref="ShortMessage"/>
                <xs:element ref="DeliveryInformation"/>
            </xs:choice>
        </xs:sequence>
    </xs:complexType>

```

```

        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="DeliverRequestType">
        <xs:choice>
            <xs:element ref="ShortMessage"/>
            <xs:element ref="DeliveryInformation"/>
        </xs:choice>
    </xs:complexType>
    <xs:complexType name="SubmissionResponseType">
        <xs:sequence>
            <xs:element name="MessageRef"
type="MessageRefType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ErrorResponseType">
        <xs:sequence>
            <xs:element name="ErrorCode" type="xs:QName"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ExpandedSubmissionResponseType">
        <xs:sequence>
            <xs:element name="SubmissionError" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="Destination"/>
                        <xs:element name="ErrorCode"
type="xs:QName"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="SubmissionOK" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="Destination"/>
                        <xs:element name="MessageRef"
type="MessageRefType"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="DeliveryErrorResponseType">
        <xs:sequence>
            <xs:element name="DeliveryFailureReason"
type="DeliveryFailureReasonType" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="DeliveryFailureReasonType">
        <xs:attribute name="value" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration
value="DestUnavailable"/>
                    <xs:enumeration
value="DestAddressInvalid"/>
                    <xs:enumeration
value="PermNetworkError"/>
                    <xs:enumeration
value="TempNetworkError"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="ExternalBillingType">
        <xs:sequence>

```

```

        <xs:any namespace="##any" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any"
processContents="lax">
        <xs:annotation>
            <xs:documentation>
                Prose in the spec does not specify that attributes are
allowed on the Body element
            </xs:documentation>
        </xs:annotation>
    </xs:anyAttribute>
</xs:complexType>
</xs:schema>

```

## 9.2.4 MessageAdmin.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.smsforum.net/schemas/smap/v1.0"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:include
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/ShortMe
ssage.xsd"/>
    <xs:element name="CancelRequest" type="CancelRequestType"/>
    <xs:element name="QueryRequest" type="QueryRequestType"/>
    <xs:element name="ReplaceRequest" type="ReplaceRequestType"/>
    <xs:element name="QueryResponse" type="QueryResponseType"/>
    <xs:complexType name="CancelRequestType">
        <xs:choice>
            <xs:element name="CancelByReference"
type="CancelByReferenceType"/>
            <xs:element name="CancelByService"
type="CancelByServiceType"/>
        </xs:choice>
    </xs:complexType>
    <xs:complexType name="CancelByReferenceType">
        <xs:sequence>
            <xs:element ref="Source"/>
            <xs:element name="MessageRef"
type="MessageRefType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="CancelByServiceType">
        <xs:sequence>
            <xs:element ref="Source"/>
            <xs:element ref="Destination"/>
        </xs:sequence>
        <xs:attribute name="serviceType" type="xs:string"
use="required"/>
    </xs:complexType>
    <xs:complexType name="MessageFinalDateType">
        <xs:attribute name="value" type="xs:string"
use="required"/>
    </xs:complexType>
    <xs:complexType name="NetworkDeliveryErrorStatusType">
        <xs:attribute name="value" type="xs:string"
use="required"/>
    </xs:complexType>

```

```

<xs:complexType name="QueryRequestType">
  <xs:sequence>
    <xs:element name="MessageRef"
type="MessageRefType"/>
    <xs:element ref="Source" minOccurs="0"/>
    <xs:element ref="Destination" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="QueryResponseType">
  <xs:sequence>
    <xs:element name="MessageRef"
type="MessageRefType"/>
    <xs:element name="MessageState"
type="MessageStateType"/>
    <xs:element name="MessageFinalDate"
type="MessageFinalDateType" minOccurs="0"/>
    <xs:element name="NetworkDeliveryErrorStatus"
type="NetworkDeliveryErrorStatusType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ReplaceRequestType">
  <xs:sequence>
    <xs:element name="MessageRef"
type="MessageRefType"/>
    <xs:element name="ReplaceMessage"
type="ReplaceMessageType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ReplaceBinaryData" type="xs:string"/>
<xs:complexType name="ReplaceBodyType">
  <xs:choice>
    <xs:element ref="ReplaceText"/>
    <xs:element ref="ReplaceBinaryData"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="ReplaceHeaderType">
  <xs:sequence>
    <xs:element ref="Source"/>
    <xs:element name="DeliveryOptions"
type="DeliveryOptionsType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ReplaceMessageType">
  <xs:sequence>
    <xs:element name="ReplaceHeader"
type="ReplaceHeaderType"/>
    <xs:element name="ReplaceBody"
type="ReplaceBodyType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ReplaceText" type="xs:string"/>
</xs:schema>

```

## 9.2.5 Addressing.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
xmlns="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

```

```

<xs:complexType name="SMAddressType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Number"
type="NumberType"/>
      <xs:element name="Application"
type="ApplicationAddressType"/>
      <xs:element name="IP" type="IPAddressType"/>
      <xs:element name="EMail"
type="RFC2822AddressType"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="NumberType">
  <xs:annotation>
    <xs:documentation>Phone number format SME
address</xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:token">
      <xs:attribute name="TON"
default="International">
        <xs:simpleType>
          <xs:restriction
base="xs:NMTOKEN">
            <xs:enumeration
value="Unknown"/>
            <xs:enumeration
value="International"/>
            <xs:enumeration
value="National"/>
            <xs:enumeration
value="Network"/>
            <xs:enumeration
value="Subscriber"/>
            <xs:enumeration
value="Alphanumeric"/>
            <xs:enumeration
value="Abbreviated"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="NPI" default="ISDN">
        <xs:simpleType>
          <xs:restriction
base="xs:NMTOKEN">
            <xs:enumeration
value="Unknown"/>
            <xs:enumeration
value="ISDN"/>
            <xs:enumeration
value="Data"/>
            <xs:enumeration
value="Telex"/>
            <xs:enumeration
value="LandMobile"/>
            <xs:enumeration
value="National"/>
            <xs:enumeration
value="Private"/>
            <xs:enumeration
value="ERMES"/>
            <xs:enumeration
value="WAPClient"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>

```

```

        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="BearerType">
    <xs:attribute name="type" default="Unknown">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="Unknown"/>
                <xs:enumeration value="SMS"/>
                <xs:enumeration value="CSD"/>
                <xs:enumeration value="Packet"/>
                <xs:enumeration value="USSD"/>
                <xs:enumeration value="CDPD"/>
                <xs:enumeration value="DataTAC"/>
                <xs:enumeration value="FLEX"/>
                <xs:enumeration
value="CellBroadcast"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="telematicsId" type="xs:integer"/>
</xs:complexType>
<xs:simpleType name="ApplicationShortCodeType">
    <xs:restriction base="ApplicationAddressType">
        <xs:pattern value="(\+)?([0-9])+"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ApplicationNameType">
    <xs:restriction base="ApplicationAddressType">
        <xs:pattern value="(\+)?([0-9])+"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ApplicationAddressType">
    <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="ExtendedAddressType">
    <xs:complexContent>
        <xs:extension base="SMAddressType">
            <xs:sequence>
                <xs:element name="Bearer"
type="BearerType" minOccurs="0"/>
                <xs:element name="SubAddress"
type="SubAddressType" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="networkType"
default="Unknown">
                <xs:simpleType>
                    <xs:restriction
base="xs:NMTOKEN">
                        <xs:enumeration
value="Unknown"/>
                        <xs:enumeration
value="GSM"/>
                        <xs:enumeration
value="TDMA"/>
                        <xs:enumeration
value="CDMA"/>
                        <xs:enumeration
value="PDC"/>
                        <xs:enumeration
value="PHS"/>
                        <xs:enumeration
value="iDEN"/>
                        <xs:enumeration
value="AMPS"/>
                        <xs:enumeration
value="Paging"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="subUnit"
default="Unknown">
        <xs:simpleType>
            <xs:restriction
base="xs:NMTOKEN">
                <xs:enumeration
value="Unknown"/>
                <xs:enumeration
value="Display"/>
                <xs:enumeration
value="Equipment"/>
                <xs:enumeration
value="SmartCard"/>
                <xs:enumeration
value="ExternalUnit"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:simpleType name="IPCombinedAddressType">
    <xs:union memberTypes="IPV4AddressType
IPV6AddressType"/>
</xs:simpleType>
<xs:complexType name="IPAddressType">
    <xs:simpleContent>
        <xs:extension base="IPCombinedAddressType">
            <xs:attribute name="port"
type="xs:integer"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="SubAddressBufferType">
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9a-fA-f]{0,44}"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="SubAddressType">
    <xs:annotation>
        <xs:documentation>contains an NSAP or user-
specified subaddress in hex format. An odd number of hex digits
is allowed.</xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="SubAddressBufferType">
            <xs:attribute name="addressType"
default="NSAP">
                <xs:simpleType>
                    <xs:restriction
base="xs:NMTOKEN">
                        <xs:enumeration
value="NSAP"/>
                        <xs:enumeration
value="UserSpecified"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="IPV4AddressType">
    <xs:annotation>

```

```

        <xs:documentation
source="http://www.ietf.org/rfc/rfc2396.txt"
xml:lang="en">SOURCE: IETF "RFC 2396: Uniform Resource
Identifiers (URI): Generic Syntax" [1998-08]
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
    <xs:pattern value="((([0-9]{1,2}|[1][0-9]{2}|[2][0-
4][0-9]| [2][5][0-5])\.){3}([0-9]{1,2}|[1][0-9]{2}|[2][0-4][0-
9]| [2][5][0-5]))"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="IPV6AddressType">
    <xs:annotation>
        <xs:documentation
source="http://www.ietf.org/rfc/rfc2373.txt"
xml:lang="en">SOURCE: IETF "RFC 2373: IP Version 6 Addressing
Architecture" [1998-7]
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
    <xs:pattern value="([A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-
9]{1,4}"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="RFC2822AddressType">
    <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:element name="SMAddress" type="SMAddressType"/>
<xs:element name="Destination" type="SMAddressType"/>
<xs:element name="Destinations">
    <xs:complexType>
        <xs:sequence maxOccurs="unbounded">
            <xs:element ref="Destination"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Source" type="SMAddressType"/>
<xs:element name="ExtendedSMAddress"
type="ExtendedAddressType"/>
<xs:group name="DestinationGroup">
    <xs:choice>
        <xs:element ref="Destination"/>
        <xs:element ref="Destinations"/>
        <xs:element name="DestinationList"
type="URLType"/>
    </xs:choice>
</xs:group>
<xs:complexType name="BindAddressType">
    <xs:choice>
        <xs:element ref="SMAddress"/>
        <xs:element name="Range">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:restriction
base="NumberType">
                        <xs:pattern value="[+0-9A-
Z^$]+"\>
                    </xs:restriction>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xs:choice>
</xs:complexType>
<xs:complexType name="URLType">
    <xs:attribute name="link" type="xs:anyURI"/>
</xs:complexType>

```

```
</xs:schema>
```

## 9.2.6 MessageExtensions.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/smap/v1.0"
xmlns="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/Addressing.xsd"/>
  <xs:include
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/ShortMessage.xsd"/>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>
  <xs:complexType name="ExtendedHeaderType">
    <xs:complexContent>
      <xs:extension base="MessageHeaderType">
        <xs:sequence>
          <xs:element name="NetworkHandling"
type="NetworkHandlingType" minOccurs="0"/>
          <xs:element name="ApplicationOptions"
type="ApplicationOptionsType" minOccurs="0"/>
          <xs:element name="DeviceControl"
type="DeviceControlType" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ExtendedDeliveryInfoType">
    <xs:complexContent>
      <xs:extension base="DeliveryInfoType">
        <xs:sequence>
          <xs:element name="UserResponseCode"
type="UserResponseCodeType"/>
        </xs:sequence>
        <xs:attribute name="InfoType"
use="required">
          <xs:simpleType>
            <xs:restriction
base="xs:NMTOKEN">
              <xs:enumeration
value="NetworkReceipt"/>
              <xs:enumeration
value="ReadReceipt"/>
              <xs:enumeration
value="DeviceAck"/>
              <xs:enumeration
value="UserAck"/>
              <xs:enumeration
value="IntermediateStatusNotification"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="NetworkHandlingType">
```

```

    <xs:attribute name="serviceType" use="optional"
default="NULL">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="NULL"/>
            <xs:enumeration
value="CellularMessaging"/>
            <xs:enumeration
value="CellularPaging"/>
            <xs:enumeration
value="VoiceMailNotify"/>
            <xs:enumeration
value="VoiceMailAlert"/>
            <xs:enumeration value="WAP"/>
            <xs:enumeration value="WAP_WDP"/>
            <xs:enumeration value="WAP_WCMP"/>
            <xs:enumeration value="USSD"/>
            <xs:enumeration value="Custom"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
    <xs:attribute name="customServiceName" type="xs:string"
use="optional"/>
    <xs:attribute name="storageMode" use="optional"
default="StoreAll">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="LatestOnly"/>
            <xs:enumeration value="StoreAll"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
    <xs:attribute name="transferMode" use="optional">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="Immediate"/>
            <xs:enumeration value="Datagram"/>
            <xs:enumeration
value="StoreAndForward"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
    <xs:attribute name="QOSTimeToLive"
type="xs:nonNegativeInteger"/>
</xs:complexType>
    <xs:complexType name="ApplicationOptionsType">
        <xs:sequence>
            <xs:element name="NumberOfVoiceMails"
type="NumberOfMessagesType" minOccurs="0"/>
            <xs:element name="Callback" type="callbackType"
minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="ITSSession"
type="ITSSessionType" minOccurs="0"/>
            <xs:element name="ITSReplyType"
type="ITSReplyTypeType" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="DeviceControlType">
        <xs:all>
            <xs:element name="GSMOptions"
type="GSMOptionsType" minOccurs="0"/>
            <xs:element name="DisplayTime"
type="DisplayTimeType" minOccurs="0"/>
            <xs:element name="MSValidity"
type="MSValidityType" minOccurs="0"/>
            <xs:element name="MessageWaitInd"
type="MessageWaitIndType" minOccurs="0"/>

```

```

        <xs:element name="ReceiveAlert"
type="ReceiveAlertType" minOccurs="0"/>
    </xs:all>
    <xs:attribute name="deviceAckReq" use="optional">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="off"/>
                <xs:enumeration value="on"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="userAckReq" use="optional">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="off"/>
                <xs:enumeration value="on"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="GSMOptionsType">
    <xs:sequence>
        <xs:choice>
            <xs:element name="ProtocolId">
                <xs:complexType>
                    <xs:attribute name="value"
type="HexByteType"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="MessageType">
                <xs:complexType>
                    <xs:attribute name="value">
                        <xs:simpleType>
                            <xs:restriction
base="xs:integer">
                                <xs:minInclusive value="0"/>
                                <xs:maxInclusive value="7"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:sequence>
    <xs:attribute name="replyPath" default="NotSet">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="Set"/>
                <xs:enumeration value="NotSet"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="CallbackType">
    <xs:sequence>
        <xs:element name="Number" type="NumberType"/>
        <xs:element name="Display"
type="CallbackDisplayType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="presentation">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="Allowed"/>
                <xs:enumeration value="Restricted"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>

```

```

        <xs:enumeration
value="NumberNotAvailable"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="screening">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="UserProvided-
Unscreened"/>
            <xs:enumeration value="UserProvided-
Verified"/>
            <xs:enumeration value="UserProvided-
VerificationFailed"/>
            <xs:enumeration
value="NetworkProvided"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="numberEncoding" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="DTMF"/>
            <xs:enumeration value="ASCII"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="RegisteredDeliveryType"/>
<xs:complexType name="CallbackDisplayType">
    <xs:choice>
        <xs:element name="Text" type="MessageTextType"/>
        <xs:element name="BinaryData"
type="BinaryDataType"/>
    </xs:choice>
</xs:complexType>
<xs:complexType name="ITSSessionType">
    <xs:attribute name="sessionNumber" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="255"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="sequenceNumber" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="127"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="finalMessage" default="FALSE">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="TRUE"/>
                <xs:enumeration value="FALSE"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="ITSReplyTypeType">
    <xs:attribute name="value" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>

```

```

        <xs:maxInclusive value="8"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="UserResponseCodeType">
    <xs:attribute name="value" type="xs:string"
use="required"/>
</xs:complexType>
<xs:complexType name="DisplayTimeType">
    <xs:attribute name="value" default="1">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="0"/>
                <xs:enumeration value="1"/>
                <xs:enumeration value="2"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="MSvalidityType">
    <xs:attribute name="value" default="Indefinite">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="Indefinite"/>
                <xs:enumeration value="PowerDown"/>
                <xs:enumeration
value="SIDRegistrationArea"/>
                <xs:enumeration value="DisplayOnly"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="MessageWaitIndType">
    <xs:attribute name="setting" default="Active">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="Active"/>
                <xs:enumeration value="Inactive"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="messageType" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="VoiceMail"/>
                <xs:enumeration value="Fax"/>
                <xs:enumeration value="eMAIL"/>
                <xs:enumeration value="Other"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="NumberOfMessagesType">
    <xs:attribute name="value" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="99"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="ReceiveAlertType">
    <xs:attribute name="tone" type="xs:integer"
use="optional"/>
</xs:complexType>

```

```

<xs:complexType name="ExtendedBodyType">
  <xs:complexContent>
    <xs:extension base="MessageBodyType">
      <xs:sequence>
        <xs:element name="UserDataHeader"
type="UserDataHeaderType" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="charEncoding">
        <xs:simpleType>
          <xs:restriction
base="xs:NMTOKEN">
            <xs:enumeration
value="DefaultAlphabet"/>
            <xs:enumeration
value="IA5"/>
            <xs:enumeration
value="OCTET"/>
            <xs:enumeration
value="Latin1"/>
            <xs:enumeration
value="JIS"/>
            <xs:enumeration
value="Cyrillic"/>
            <xs:enumeration
value="Latin-Hebrew"/>
            <xs:enumeration
value="UCS2"/>
            <xs:enumeration
value="Pictogram"/>
            <xs:enumeration
value="ISO-2022-JP"/>
            <xs:enumeration
value="Extended-Kanji-JIS"/>
            <xs:enumeration
value="KCS-C-5601"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute ref="xml:lang"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="UDHHexElementType">
  <xs:simpleContent>
    <xs:extension base="xs:hexBinary">
      <xs:attribute name="ident" type="HexByteType"
use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="UDHStringElementType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="ident" type="HexByteType"
use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="UserDataHeaderType">
  <xs:choice>
    <xs:element ref="EncodedUDH"/>
    <xs:sequence maxOccurs="unbounded">
      <xs:choice>
        <xs:element name="HexInfoElement"
type="UDHHexElementType" />
        <xs:element name="StringInfoElement"
type="UDHStringElementType" />
      </xs:choice>
    </xs:sequence>
  </xs:choice>
</xs:complexType>

```

```

        </xs:choice>
    </xs:sequence>
</xs:choice>
</xs:complexType>
<xs:element name="EncodedUDH" type="xs:hexBinary"/>
<xs:simpleType name="HexByteType">
    <xs:restriction base="xs:hexBinary">
        <xs:length value="1"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

## 9.2.7 SMAPFeatures.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.smsforum.net/schemas/smap/v1.0"
xmlns:mmap="http://www.smsforum.net/schemas/mmap/v1.0"
xmlns:smap="http://www.smsforum.net/schemas/smap/v1.0"
elementFormDefault="unqualified">
    <xs:import
namespace="http://www.smsforum.net/schemas/mmap/v1.0"
schemaLocation="http://www.smsforum.net/schemas/mmap/v1.0/mmap.xsd"/>
    <xs:include
schemaLocation="http://www.smsforum.net/schemas/smap/v1.0/Addressing.xsd"/>
    <xs:complexType name="SMAPFeatureSetType">
        <xs:complexContent>
            <xs:restriction base="mmap:FeatureSetType">
                <xs:all>
                    <xs:element name="RequiredOperations"
type="mmap:FeatureListType" minOccurs="0"/>
                    <xs:element name="SupportedOperations"
type="mmap:FeatureListType" minOccurs="0"/>
                    <xs:element name="Options"
type="mmap:FeatureListType" minOccurs="0"/>
                    <xs:element name="Parameters" minOccurs="0">
                        <xs:complexType>
                            <xs:attribute name="maxMessageSize"
type="xs:unsignedInt"/>
                            <xs:attribute
name="expandedSubmitResponse" type="xs:boolean"/>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="ApplicationSettings"
minOccurs="0">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element
name="ApplicationAddress" type="BindAddressType"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:all>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="SMAPFeatures" type="SMAPFeatureSetType"/>
</xs:schema>

```



## 10 Change List

Version	Sections	Description
1.11	4.2.6, 4,27	Batch definitions now in specification
1.11	7.5.4	UDH specification changed to reflect need for string content in EMS. Full description of UDH provided.
1.11	3.1.2.1	Example response provided in introduction
1.11	4.1.3	Billing content now completely flexible (uses xs:any)
1.11	1.5	References to XML schema and SOAP added
1.11	GENERAL	Examples extended and improved
1.11	8	Full end-to-end example now provided
1.11	GENERAL	Examples now use actual URLs for schema definitions
1.12	1.6	Added change list
1.12	3.1.2, 3.1.3, 3.1.4, 4.2.1.1, 4.2.2.1, 4.2.6.1, 8.1	Removed ApplicationSettings information element. Required and supported operations should not be nested inside this element..
1.12	4.1.1	Added explicit description of sourceOperationReference type
1.12	7.4.1.1	Clarification of timezone selection for scheduled delivery time
1.12	7.5.1	Clarification of storage mode operation
1.12	7.5.2, 7.5.3	Clarified network and device-specific nature of the elements defined in these sections.
1.12	7.6.1	Clarified operation when TON and NPI attributes not set for Number.
1.12	7.8	Fixed maxMessageSize example
1.12	7.5.4	New section on UDH implementation details.
1.12	7.4.2	Changed External content attributes. Text and external content now allowed in same message.