

LEMONADE WG
Internet Draft

Document: [draft-stebrose-mmsarch-00.txt](#)
Expires: September 2003

Alan K. Stebbens
Milt Roselinsky
Openwave Systems
March 2003

Mobile Messaging Architectures and Requirements

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#) [1].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

Abstract

The increasing importance of messaging as a potential source of revenue for mobile networks has led operators to build or procure mobile messaging solutions. Some operators have built mobile messaging solutions using IETF standards (IMAP, SMTP, MIME). Another solution has been developed by consensus in the 3GPP and OMA groups, based on MIME, HTTP methods and WAP PUSH, which has been deployed by many operators.

This document presents a taxonomy of messaging architecture components and models, providing a comparison of their feature sets. It also identifies the commonalities of these mobile messaging solutions and abstracts from these a set of mobile messaging requirements.

The information is provided to inform and encourage future discussion of and improvements to Internet messaging in order to increase their applicability to mobile messaging systems.

Conventions used in this document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119](#) [2].

Table of Contents

1.	Introduction.....	2
2.	Components and Models of Mobile Messaging.....	3
2.1	Mobile Messaging Components.....	3
2.2	Mobile Messaging Transaction Models.....	3
2.3	Mobile Messaging Transactions.....	4
2.4	Submission.....	4
2.5	Notification.....	5
2.6	Retrieval.....	6
3.	Mobile Messaging Architectures.....	6
3.1	Short Message Service (SMS).....	7
3.2	Mobitex.....	7
3.3	Extended Messaging Service (EMS).....	8
3.4	Web/WAP-based Messaging.....	8
3.5	Mobile Messaging Based on IMAP4 and SMTP.....	9
3.6	MMS Messaging.....	10
4.	Messaging Call Flows.....	11
5.	Analysis of Mobile Messaging Requirements.....	12
5.1	Mobile Messaging Requirements.....	12
6.	Security Considerations.....	13
7.	References.....	13
8.	Acknowledgments.....	15
9.	Author's Addresses.....	15

[1. Introduction](#)

Messaging has proven to be an important application, developed and extensively used for more than a decade over wire line networks. Over the last few years, messaging applications have also been deployed across wireless networks worldwide. Initially, the mobile messaging facilities were simple: text-based and person-to-person (or device-to-device). Correspondingly, the mechanisms used to transport the simple text-based messages were also simple.

Subsequently, additional features have been added to the mobile messaging services, resulting in rich multimedia services with many

features. Photo messaging services, for example, first appeared in Japan in 2000 (based on IMAP4 [3]), then, in 2002, in Europe and in the Americas (based on OMA MMS [4]).

These rich multimedia mobile messaging services have required richer messaging transport protocols which are capable of transporting and managing the varied multimedia objects, as well as providing the appropriate messaging semantics. IMAP4, with its rich messaging feature set, has been used as the message transport protocol in several mobile messaging architectures. OMA MMS has been developed by 3GPP([5],[6]) and OMA to provide for some of the mobile messaging service functions, and is widely supported by mobile vendors and their operator-customers.

2. Components and Models of Mobile Messaging

From a survey of existing messaging architectures, there are is a common client server architecture with mobile clients and operator servers. This architecture is differentiated by whether the clients have push model or pull model transactions between them.

2.1 Mobile Messaging Components

The client, also known as a "user agent", presents messages to and accepts messages from the user, and performs messaging functions on behalf of the user, sometimes automatically, sometimes with interrogation of the user.

The server, also known as a "relay/server" or a "proxy-relay", generally resides in the operator's network and manages messages on behalf of some or all of the operator's subscribers. It accepts connections from clients, typically authenticated, performs requested functions, and, importantly, generates billing records appropriate to the requested function. The server also provides storage of the message for some length of time (depending on service definition).

2.2 Mobile Messaging Transaction Models

As stated above, there are two basic transactional models. The "pull" model is where the component with the message data initiates the transaction. For example a client may initiate a connection to a server and issue requests to the server to discover and retrieve messages as appropriate. Conventional e-mail clients, web-mail clients, and WAP-based mobile clients use the "pull" model.

The "push" model is where the component that wishes to operate on the data initiates the transaction. For example, it is common that the

arrival of a new message at the terminating server causes a notification to be sent ("pushed") to a messaging client.

The push model has the advantage of being event or message driven. User interaction only occurs when message data is available. Unlike the pull model where the user must poll for new message data (or configure their client to poll for them).

2.3 Mobile Messaging Transactions

The most common functions are: "submission", "notification", and "retrieval". There may be other functions, such as "delivery reports", "read-reply reports", "forwarding", "view mailbox", "store message", etc. Each of these transactions can be implemented in either a pull or push model. However, some transactions are more naturally suited to one model or another.

The following Figure 1 is a simple depiction of the basic messaging model:

- (1) Message submission
- (2) Message notification
- (3) Message retrieval

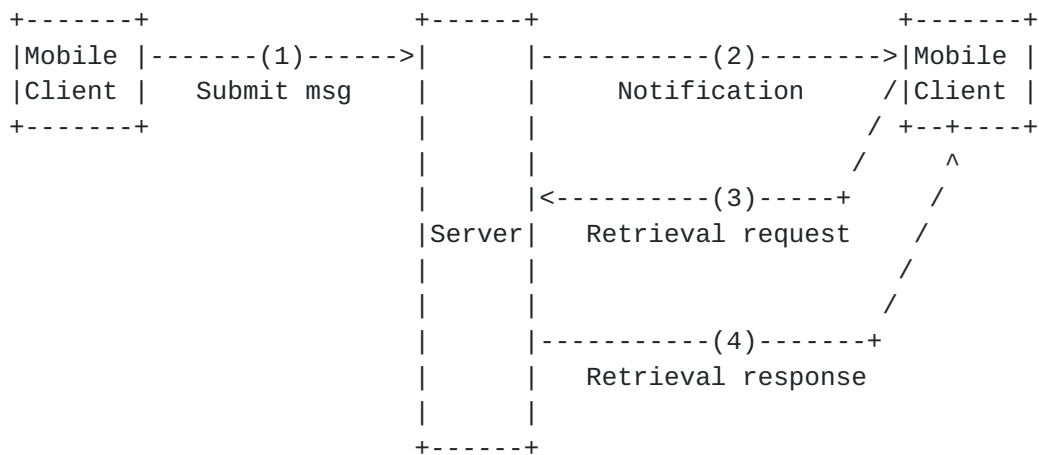


Figure 1 - Basic Messaging Model

2.4 Submission

The "submission" transaction is the transaction between a client and a server by which the user of the former can send a new message to another user.

This transaction is most commonly implemented in the push model. The when the user has composed a message, the user directs the client to initiate a connection to the server and push the message up to the server.

In a pull model implementation, the server would connect to the client from time to time and query for the presence of new messages to send. In practice, this is never done.

2.5 Notification

The "notification" transaction is the transaction by which the server notifies the client that it has received messages intended for that client.

This transaction is also most commonly implemented as a push transaction. In this model, the server will initiate a connection to the client and send data to the client indicating that new message data is available for the client. SMS [7], by virtue of already being widely available, has become the ubiquitous protocol that provides for a basic notification service. Given that SMS packets are limited in size, the prevalent use of SMS is to send a summary of multimedia messages, including a reference, in the notification message.

The WAP Forum, now part of the Open Mobile Alliance (OMA), published a flexible "Push Architecture" (WAP Push [8], PAP [9]), with which services can have messages pushed to mobile devices, either over HTTP [10] or SMS. There is some talk of adapting the Push Proxy-Gateway [11] to use SIP NOTIFY for notification transport, but this, too, is still work-in-progress.

Regardless of the actual transport method, the essential ingredient to all of them is that either the message itself (if it is a short one) or a summary of and reference to the message must be sent to the client. In the case of a short message being sent via a notification, the client can then present it to the user. In the case of a long message, the client can either then present a summary of the message, or, by using the message reference, retrieve additional information about the message, or even the complete message, and present it appropriately to the user.

The significant advantage of the push model notifications is that data is presented to the user without the user needing to be aware of network/transport latencies, and without tying up network resources for polling when there is no new data. All of the larger mobile messaging systems implement a push model for the notification transaction.

The "pull" model of PC-based or WAP-based mobile e-mail clients has not required a notification method per-se. When a new message arrives at the server, the client learns of it only when it next initiates a connection to the server and requests a list of messages (or, more optimally, a list of new message). Clients can be configured to automatically connect to the server and determine whether or not new messages are available, and notify the user as desired. This has been completely acceptable to the client community and so, there has been no protocol development in this area.

There has been recent work to define "e-mail notification" methods SIP-MWI [12], EMN [13], SNAP [14], but these are still works-in-progress.

2.6 Retrieval

The "retrieval" transaction is the transaction between a client and a server by which the client can obtain one or more messages from the server.

In a push model implementation, the server will initiate a connection to the client and push the message data to the client. This is done in OMA [4] in "Immediate Mode." The advantage of this is that the user is not necessarily aware of transport or network latencies. The disadvantage is that the client may not be able to store the message due to size constraints.

In a pull model implementation, the client will initiate a connection to the server and retrieve the message from the server. Often the client will first retrieve information about the message (e.g. IMAP4 BODYSTRUCTURE [3]), and allow the user to selectively download the message from the server to the client. The advantage of this method is that the user can control what data is actually sent to and stored by the client.

In both implementations (push or pull) the server can be configured to keep a copy of the message data in server-managed storage. This is considered very valuable by many users. The server-managed message data is available for other operations without requiring upload from the client. For example, it can be referenced and forwarded as part of another message, or it may be viewed from another client, whether mobile or fixed.

3. Mobile Messaging Architectures

Although many of the transport protocols and interfaces have been standardized, mobile operators compete with how they combine the standard protocols and interfaces into an innovative, comprehensive

messaging architecture that fulfills their service requirements. The following sections are a brief survey of several mobile messaging architectures, their features, and their problems.

3.1 Short Message Service (SMS)

The Short Message Service (SMS) allows the exchange of short text-only messages on mobile telephone networks. SMS emerged in the early 1990s and by 2001 had become a very popular consumer-oriented messaging service, when users exchanged more than 100 billion messages.

SMS was originally standardized as part of the GSM phase 1 ETSI technical specifications.

Features:

- . short-messages of less than 140 octets
- . E.164-based[15] or "short code" addressing
- . Text only
- . Client presentation is easy (because text-only)
- . First over-the-air immediate messaging protocol
- . Additional functions (CANCEL, REPLACE) to support content-providers (VASPs)
- . Available on almost every handset: TDMA, CDMA, GSM, GPRS

Problems:

- . Requires expensive SS7[16] network interface
- . Each SMS data packet requires SS7 signaling, which is also used for voice signaling
- . No network mailbox support
- . No standard Internet interworking (e-mail gateways exist but are ad-hoc or commercial products)
- . Requires specialized hardware systems called SMSCs
- . There are several "standard" interfaces for inter-SMSCs, creating the need for inter-SMSC switches

3.2 Mobitex

Mobitex is a proprietary service, created by Ericsson, that is a packet-switched, narrowband PCS network, designed for wide-area wireless data communications. Mobitex networks offer a variety of paging solutions, the most advanced of which is called "interactive paging", a 2 way text messaging solution.

Features:

- . Short-messages of less than 500 octets
- . Device addressing only
- . Text-only
- . Transport supports submit, delivery, & read reports

Problems:

- . Proprietary Ericsson network (based on X.25 [17])
- . Device addressing only
- . No network mailbox support
- . Limited client availability
- . No standard Internet interworking interface

[3.3](#) Extended Messaging Service (EMS)

The Extended Messaging Service (EMS [18]) is an enhancement to the SMS service that enables the exchange of multimedia messages. EMS utilizes the same network infrastructure as SMS [7].

Features:

- . Long messages sent via concatenated SMS packets
- . E.164-based [15] or "short code" addressing
- . Text, bit-map images, and vector graphics

Problems:

- . Requires expensive SS7 [16] network interface
- . Each EMS "packet" is one or more SMS data packets, each of which requires SS7 signaling that is also used for voice signaling
- . No network mailbox storage
- . No standard interworking with Internet e-mail systems
- . Client presentation of media less simple
- . No capability discovery or content negotiation between EMS devices (what happens if a bitmap image is sent to another address that does not support that image type?)
- . Requires specialized hardware systems called EMSCs (upgraded SMSCs)
- . EMSCs continue the need for SMSC/EMSC switches

[3.4](#) Web/WAP-based Messaging

Perhaps the first architecture realized for mobile messaging that allowed for the possibility of presenting a unified interface for multimedia messages was the WAP-based mobile e-mail service.

Features:

- . Leverages ubiquitous HTTP [10] as transport
- . [RFC2822](#) addressing [20]
- . Large and multimedia message support
- . Client capability discovery and content adaptation based on USERAGENT string [10]
- . Can support delivery and read reports (server-based)
- . Mailbox functions fully supported
- . Client requires only a WAP browser
- . Message transported via IP network, using Internet protocols

- . Interoperates with Internet e-mail systems

Problems:

- . No E.164 [7] or "short code" addressing, except possibly for only the mobile operator's network
- . Requires network connectivity for all operations, even message composition
- . User typically exposed to the network latencies (e.g., no "background" submission or retrieval)
- . Server-based message composition not integrated with client address book
- . No notifications of newly arrived messages - user or client must request new messages
- . Mobile radio access billing not necessarily integrated with message service billing

It is interesting to note that one of the major complaints with mobile web mail services was network latencies, which is a consequence of the design of the relatively stateless, browser-based client, and not necessarily a consequence of using WSP and/or HTTP.

3.5 Mobile Messaging Based on IMAP4 and SMTP

Several mobile operators have built a mobile messaging architecture based on IETF messaging protocols: SMTP [19], IMAP4 [3], and used SMS[7] for their notification protocol. Today, these operators have proven, successful, revenue-generating multimedia messaging services.

Features:

- . [RFC2822](#) [20] and E.164 [15] addressing, including "short codes"
- . Large and multimedia message support, MIME [21] encapsulation
- . Can support delivery reports (DSNs [22]) and read reports (MDNs [23])
- . Messages submitted, retrieved, and forwarded via IP network, using Internet protocols
- . Wide interoperability with Internet e-mail systems.
- . Notifications via WAP Push or SMS
- . Complete mailbox support available now
- . Client and server developers have ready access to widely deployed protocol development kits, test tools, etc.
- . Billing can be based on either per-message or per-byte
- . Persistent storage model supported
- . Authentication can be network-based or application-based

Problems:

- . IMAP4 too "chatty"; operators have made proprietary adaptations to avoid this.
- . Messages are base64 encoded, expanding the payload.

. Notification methods not standardized.

Stebbens & Roselinsky Expires - September 2003

[Page 9]

- . No client capability discovery or exchange
- . No content adaptation
- . No content protection mechanisms currently

3.6 MMS Messaging

The 3GPP T2 and OMA MAG MMS groups developed the Multimedia Messaging Service (MMS) specifications independently ([5], [6]), based on new HTTP [10] methods and WAP Push ([24], [25]).

The MMS Submission and Retrieval methods are fulfilled with HTTP methods. The MMS Notification is based on the WAP Push architecture, which can delivery push messages over HTTP or over SMS [7].

Features:

- . [RFC2822](#) and E.164 addressing, including "short codes"
- . Large and multimedia message support, MIME encapsulation
- . Delivery reports and read reports supported
- . Client capability discovery (via UAPROF [26] or HTTP USERAGENT [10] strings)
- . Content adaptation supported
- . Delayed delivery feature supported
- . Forward without download
- . Tightly integrated billing, supporting both pre- and post-pay operations
- . Some mailbox functions supported now
- . Messages submitted, retrieved, and forwarded via IP network, using SMTP with MMS headers
- . Internet e-mail interworking
- . Well-specified content provider API that allows partnering relationships between content owners and messaging service providers.
- . Notifications via WAP Push or SMS
- . Message is compressed to save bandwidth
- . Well-specified IOT program within OMA IOP
- . Authentication is network-based

Problems:

- . Delivery reports are not DSNs and do not interoperate with SMTP e-mail systems in a standard fashion
- . Read-reply reports are not MDNs and do not interoperate with SMTP e-mail systems in a standard fashion
- . Some mailbox functions not supported (server-based sorting, message folders)
- . Over-the-air transport protocols not widely deployed; lack of development kits and test tools
- . Content protection mechanisms are limited to a "forward lock" (OMA DRM is still a work-in-progress)

4. Messaging Call Flows

In Figure 2 (below), the basic call flows for the IETF messaging and MMS technologies are depicted, similar to the basic messaging model from Figure 1:

- (1) Message submission (from the client)
- (2) Message notification (to the client)
- (3) Message retrieval request (from the client)
- (4) Message retrieval response (deliver msg to the client)

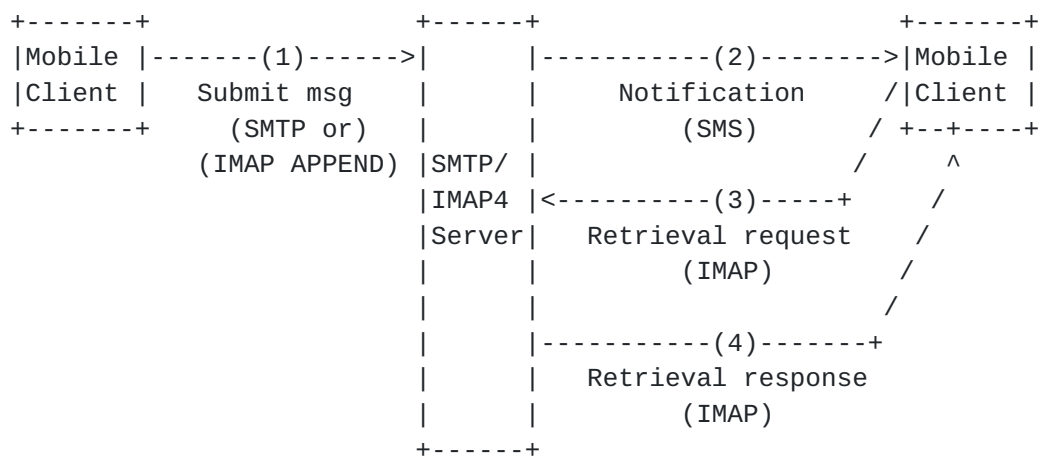


Figure 2 - Messaging Call Flow with SMTP, SMS, & IMAP

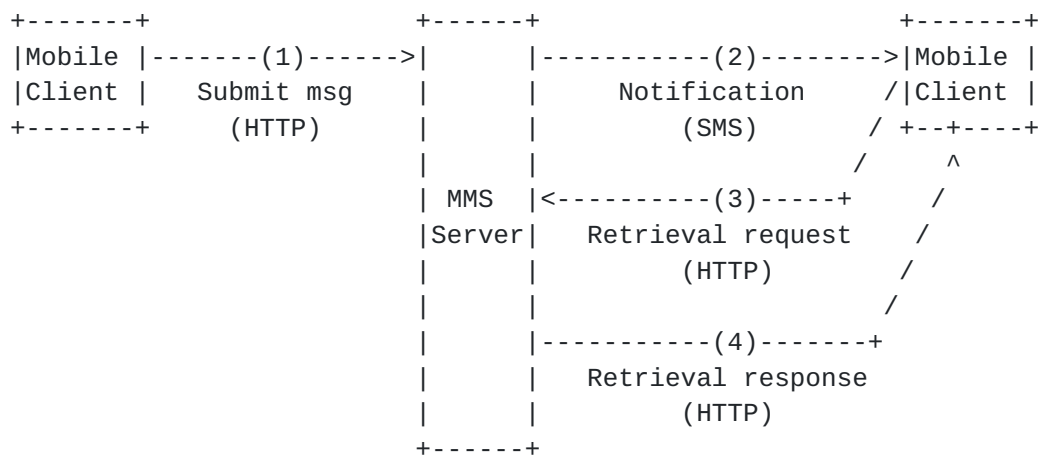


Figure 3 - MMS with specialized HTTP methods & SMS.

Please note that the call flows for both are identical except for the actual transport protocols.

5. Analysis of Mobile Messaging Requirements

As identified in the preceding descriptions, modern mobile messaging systems have a number of requirements. Some of these differ significantly from those seen in the desktop oriented email systems deployed in today's Internet.

The mobile user marketplace is targeted at a large consumer audience and requires a simple, immediate and compelling messaging experience.

Users of today's mobile messaging user agents desire a rich end user experience that features multimedia content pushed to the end user device. The devices available today are constrained as to which media codecs are supported, limited screen size, limited or expensive network bandwidth. The mobile devices are often roaming in and out of network coverage and the messaging system must take this into consideration.

Mobile network providers often operate on a "pay for use" service model. This brings in requirements for clearly delineated service transactions that can be reported to billing systems, and for positive end-to-end acknowledgement of delivery or non-delivery of messages.

5.1 Mobile Messaging Requirements

The following are requirements of a successful mobile messaging offering:

- . Push based notifications.
- . Delivery model where messages "just show up" on the device (when appropriate), based on push based notification and end-user preference settings. In other words, hide network latencies from the user, and reduce user interaction with profile-based automations.
- . [RFC2822](#) and E.164 addressing, including "short codes"
- . Large message and multimedia message support, MIME encapsulation
- . Support for end-to-end delivery reports and read reports
- . Client capability discovery/exchange and content adaptation
- . User configurable filters for selective downloading and SPAM control
- . Message exchange with existing Internet email systems
- . Mailbox (persistent storage model) support
- . Network-based and/or application-based authentication
- . Bandwidth saving features such as binary transfers, data compression, forward without download and streamlined client-server message submission and retrieval

6. Security Considerations

This draft describes mobile messaging networks based on IMAP4, SMTP and HTTP. Although all offer application-level authentication, many mobile operators have deployed mobile messaging services network-based authentication. In other words, the sessions are implicitly authenticated by the mobile device network access parameters.

7. References

- 1 Bradner, S., "The Internet Standards Process -- Revision 3", [BCP 9](#), [RFC 2026](#), October 1996.
- 2 Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997
- 3 Crispin, M., "Internet Message Access Protocol - Version 4rev1", [RFC 2060](#), University of Washington, December 1996.
- 4 Open Mobile Alliance (OMA) "Multimedia Messaging Service; Architectural Overview Version 1.1", OMA, 2002
- 5 3GPP TS 22.140 "Third Generation Partnership Project; Technical Specification Group Services and System Aspects; Service aspects; Functional description; Stage 1 Multimedia Messaging Service", 3GPP, 2001
- 6 3GPP TS 23.140 "Third Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional description; Stage 2", 3GPP, 2001
- 7 C.S0015-A: Short Message Service (SMS), December 1999, 3GPP2.
- 8 Open Mobile Alliance (OMA) "Push Architectural Overview, WAP-250-PushArchOverview-20010703-a", OMA, 2001
- 9 Open Mobile Alliance (OMA) "Push Architectural Overview Push Access Protocol Specification, WAP-247-PAP-20010429-a", OMA, 2001
- 10 Fielding, Gettys, Berners-Lee, et. al., "Hypertext Transfer Protocol - HTTP 1.1", [RFC 2616](#), June 1999.
- 11 Open Mobile Alliance (OMA) "Push Proxy Gateway Service Specification", WAP-249-PPGService-20010425.pdf, OMA, 2001

- 12 Mahy, R. "A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)", [draft-ietf-sipping-mwi-01.txt](#)
- 13 Open Mobile Alliance (OMA) "E-mail Notification Version 1.0", OMA, 2002
- 14 Shapira, N. and Aloni, E. "Simple Notification and Alarm Protocol (SNAP)", Comverse Technology, 2002.
- 15 ITU-T Recommendations Series E: "E.164: The international public telecommunication numbering plan"; ITU, May 1997.
- 16 CCITT White Book, Volume VI, Fascicle VI.7, Recommendations Q.700-Q.716: Specifications of Signalling System No. 7.

CCITT White Book Volume VI, Fascicle VI.8, Recommendations Q.721-Q.766: Specifications of Signalling System No.7.

ITU White Book, ITU-T Recommendation Q.763: Specifications of Signalling System Number 7.

GR-246-CORE, Issue 1, December 1994: Specifications of Signalling System Number 7.
- 17 ITU-T Recommendation Series X: X.25: "Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit", ITU, Oct 1996.
- 18 S.R0051-0 v1.0, "Enhanced Message Service (EMS) Stage 1 Description", 3GPP2, July 2001.
- 19 Klensin, J. "Simple Mail Transfer Protocol", [RFC 2821](#), April 2001.
- 20 Resnick, P. "Internet Message Format", [RFC 2822](#), April 2001.
- 21 Freed, N. and Borenstein, N. "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), November 1996.
- 22 Moore, K. "SMTP Service Extension for Delivery Status Notifications", [RFC 1891](#), January 1996.
- 23 R. Fajman, "An Extensible Message Format for Message Disposition Notifications", [RFC 2298](#), March 1998.

- 24 Open Mobile Alliance (OMA) "Multimedia Messaging Service; Client Transactions Version 1.1", OMA-MMS-v1_1, OMA, 2002
- 25 Open Mobile Alliance (OMA) "Multimedia Messaging Service; Encapsulation Protocol Version 1.1", OMA-MMS-v1_1, OMA, 2002
- 26 Open Mobile Alliance (OMA) "User Agent Profile, Version 1.1", OMA-UAPROF-v1_1, OMA, December 2002.

8. Acknowledgments

Benjamin Ellsworth,
Openwave Systems, Inc.
530 E. Montecito St., Santa Barbara, CA 93103
Phone: 805-884-3153
Email: benjamin.ellsworth@openwave.com

9. Author's Addresses

Alan K. Stebbens
Openwave Systems, Inc.
530 E. Montecito St., Santa Barbara, CA 93103
Phone: 805-884-3162
Email: alan.stebbens@openwave.com

Milt Roselinsky
Openwave Systems, Inc.
530 E. Montecito St., Santa Barbara, CA 93103
Phone: 805-884-6207
Email: milt.roselinsky@openwave.com

