Wireless Netw (2009) 15:163–176
DOI 10.1007/s11276-007-0019-8

# Implementation and performance evaluation for a ubiquitous and unified multimedia messaging platform

**Phone Lin · Shan-Hung Wu · Chung-Min Chen · Ching-Feng Liang**

Published online: 8 February 2007

**Abstract** As networking technology advances, more advanced message services are provided. Users may have one or more different message accounts and devices. Before sending messages, the sender must make sure which message service the receipt currently uses. Any misjudgement may delay the time when the messages are received. To make users be able to receive messages anytime and anywhere with any kind of devices, we propose a *Ubiquitous and Unified Multimedia Messaging* (UMM) platform. The UMM platform integrates different message services and provides a more efficient way for message delivery. Our design does not modify the existing protocols of message services and need not involve the network operators. An analytical model is proposed to evaluate the performance of the implemented platform. Our study shows that with a large number of message services the user subscribes and long message processing time in the network, the delayed message probability can be limited within 1.5%. This performance is considered satisfactory.

**Keywords** Message service · Performance evaluation · Unified messaging

## 1 Introduction

The *Global System for Mobile Communications* (GSM) association reported that 45.6 billion *Short Message Service* (SMS) messages/month have been sent worldwide in 2004. The message service has become one of the major applications for people to communicate with each other. With the improvement of the networking technologies, such as higher network bandwidth and powerful mobile devices, the message service is not limited to the notification with pure text. The *Multimedia Message Service* (MMS) [3, 4, 26, 35] is standardized to deliver rich content, e.g., images and audio/video clips. As estimated by a recent study from Strategy Analytics, the MMS market will scale to more than 22 billion U.S. dollars by 2008 [28]. Besides MMS, the *Instant Messaging* (IM) service is proposed to allow users to exchange messages instantly. The Radicati Group predicted that the population of IM subscribers will grow up to 1.4 billion by 2007 [29].

The current message service environments, such as MMS, IM, and E-mail, are developed independently. Users may have one or more different message service accounts and different devices. Before sending messages, the sender should make sure which message service the receipt currently uses.

This paper is an extension of the work that has won the second place of the Mobile Hero contest sponsored by Industrial Development Bureau of Ministry of Economic Affairs, Taiwan, R.O.C., and was awarded USD 15,000.

P. Lin (✉) · S.-H. Wu
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei 106, R.O.C.
e-mail: plin@csie.ntu.edu.tw

S.-H. Wu
e-mail: brandonwu@pcs.csie.ntu.edu.tw

C.-M. Chen
Department of Research Scientist, Telcordia
Technologies-Applied Research, 1 Telcordia Drive, Piscataway,
NJ 08854, U.S.A.
e-mail: cchen2@telcordia.com

C.-F. Liang
Department of Computer and Communications Research Labs,
Industrial Technology Research Institute, Hsinchu, Taiwan,
R.O.C.
e-mail: cfliang@itri.org.tw

Any misjudgement may delay the time when the messages are received. To make users be able to get messages anytime and anywhere with any kind of devices, it is crucial to have a *Unified Messaging* (UM) platform to integrate different message services.

Some previous works [7, 17, 25, 32, 33] have touched on the UM concept. They are summarized as follows: In [7], Arbanowski and van Der Meer proposed the "Service Personalization for Unified Messaging Systems". They studied how to personalize different message services in a unified way. The authors only considered the message services, such as Fax, E-mail, and Paging services. The advanced message services, such as MMS and IM, were not addressed. In [25], Miao, Impey, and Baozong discussed a seamless message service to integrate the traditional message services. In [17], Healy, Barber, and Nolan addressed how to use the JAVA language to develop a UM platform and its applications. How to realize the UM concept is not mentioned in this work. In [32, 33], Wams and van Steen proposed a middleware-based unification architecture. Their architecture does not follow the standard protocols. The implementation cost for this architecture is considered high.

In this paper, we provide a more general solution, a *Ubiquitous and Unified Multimedia Messaging* (UMM) platform, for the UM concept. The UMM platform adopts a gateway-based approach. It provides an open interface to the message services so that it is easy to integrate message services with the UMM platform. Our design does not modify the existing protocols of message services and need not involve the network operators. We implement the *Ubiquitous Message Forwarding* (UMF) algorithm to select a suitable message service to deliver the messages to the recipient. We also implement the *interactive Multimedia Message Service* (iMMS) mechanism, which allows the application services on UMM to deliver their service templates to the message senders, and enables efficient message exchanges between the end users. We conduct an analytic model to evaluate the performance of the UMM platform.

The rest of this paper is organized as follows: Section 2 presents the system architecture and the software components of the UMM platform. Section 3 describes the signaling protocol named *Unified Message Interaction Protocol* (UMIP) for delivery of a unified message. Section 4 illustrates the iMMS mechanism. Section 5 details the UMF algorithm. Section 6 proposes the analytical model. Finally, Section 7 gives a concluding remark.

## 2 The UMM system architecture

Figure 1 illustrates the system architecture of the UMM platform, which consists of three major parts and is separated by two open interfaces, OI-HN (Open Interface for Heterogeneous Networks; Fig. 1(2)) and OI-AP (Open Interface for Applications; Fig. 1(9)). The functionalities of each part are described in the following subsections.

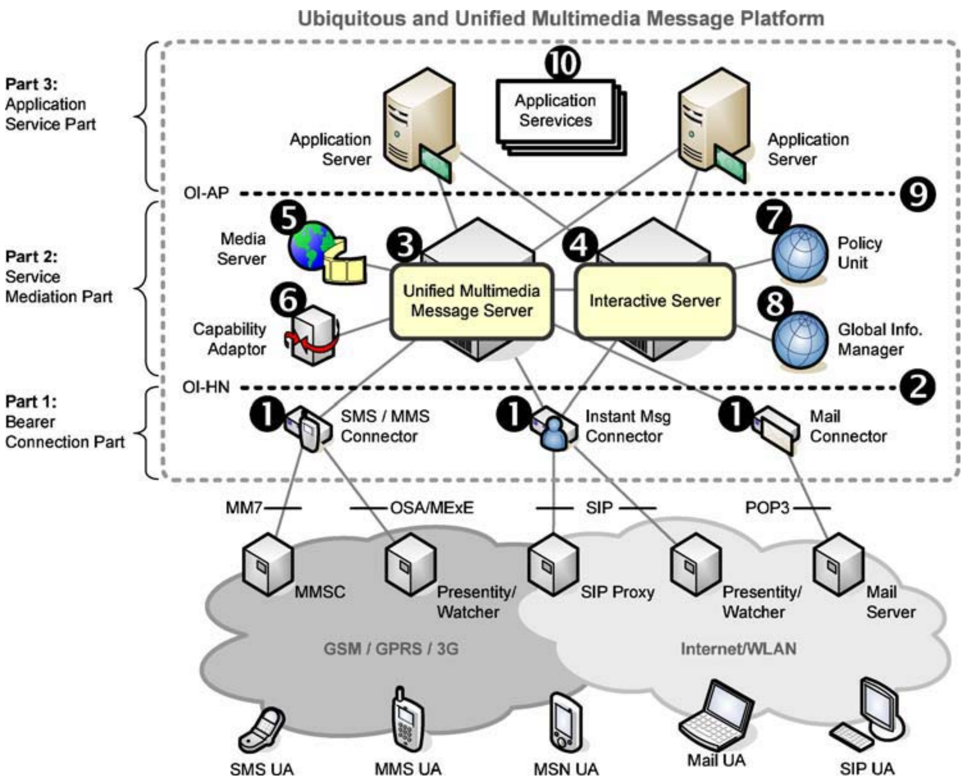


**Fig. 1** The UMM system architecture

**Table 1** The OI-HN interface

| Function name | Intended usage |
|---|---|
| InteractiveServer.User_Sub_REQ() | for user registration |
| InteractiveServer.AP_Sub_REQ() | for service subscription |
| InteractiveServer.Info_Update_REQ() | to update subscriber's information |
| InteractiveServer.Msg_Init_REQ() | to get the templates of an Interactive Multimedia Message (iMM) |
| UMMServer.Msg_Send_REQ() | to send a unified message |
| Connector.UAProf_REQ() | to instruct the connector to retrieve the user agent profile (e.g., OMA UAProf) of a terminal device |
| Connector.Msg_Deliver_IND() | to instruct the connector to deliver a unified message |

### 2.1 Bearer connection part

The Bearer Connection Part (Part1, Fig. 1) maintains the bearer connections between the UMM platform and the terminals residing in the heterogeneous networks. This part implements the SMS Connector, MMS Connector, IM Connector, and Mail Connector (Fig. 1(1)). A connector handles two major tasks:

- It talks to the message servers (e.g., *Multimedia Message Service Center* (MMSC) [4], SIP Proxy server [31], and Mail server) in heterogeneous networks by standard protocols. The UMM Server (Fig. 1(3)) can send/receive the messages to/from the message servers through the connector.
- It collects the user related information (including user presence information and terminal capability information) and then reports them to the Interactive Server (Fig. 1(4)).

The MMS Connector connects to the MMSC through the MM7 interface[1] [4]. Then the MMS Connector registers itself to the Presence Server [5] as a Presence *Service Capability Server*[2] (SCS) [1]. The registration follows the *Open Service Access* (OSA) protocol [6]. The MMS Connector can be notified of any change of the user-related information. The IM Connector connects to the *Session Initiation Protocol* (SIP) Proxy through the SIP protocol [31]. The IM Connector registers itself to the SIP Presentity[3] as a watcher. The SIP Presentity can report the latest presence information of a SIP user to the IM Connector. Following the POP3 protocol [27], the Mail Connector connects to the E-mail server.

A timeout monitor is implemented in the connector to detect any failure message transmission to/from heterogeneous networks. In the Interactive Server and UMM Server, we implement the OI-HN API (Fig. 1(2)) for installation of different message services (e.g., MMS, SMS, and IM) on the UMM platform. The functions of OI-HN API are listed in Table 1. Using the OI-HN API, we can connect UMM to a new message service by creating a correspondent connector.

From operator's point of view, the UMM platform is a third-party service node connecting the heterogeneous networks following the standard protocols. Hence, no modification of the existing protocols of message services is required. Furthermore, it is easy to come out a new Connector (connecting the new applications/services to UMM), which can be done by implement the seven functions of OI-HN API for the new Connector. The implementation overhead is minor.

### 2.2 Service mediation part

The Service Mediation Part (Part2, Fig. 1) encapsulates the messages with different formats into a unified format and exercises the UFS algorithm (to be elaborated later) to deliver the message to the receipt. There are six software components in this part: the UMM Server (Fig. 1(3)), Interactive Server (Fig. 1(4)), Media Server (Fig. 1(5)), Capability Adaptor (Fig. 1(6)), Policy Unit (Fig. 1(7)), and Global Information Manager (Fig. 1(8)). The functionalities of each component are described below:

*UMM Server* (Fig. 1(3)) converts different types of messages to/from the unified messages defined by the unified XML schema[4] [40–42, 44]. The conversion follows *eXtensible Stylesheet Language Transformation* (XSLT) standard [39, 45]. Figure 2 illustrates the structure of a unified message, which consists of three major parts. The header part (Fig. 2(1)) contains the message ID (MID), routing information, and subject, etc. for the message. It also contains information (e.g., preMID, APID, STID, LTID, and Message Type) for the iMMS mechanism, whose details will be given in Sections 3 and 4. The description part (Fig. 2(2)) contains the timing control and synchronization information to render the message content on the

[1] The MM7 is run over the HTTP layer. Before delivered through MM7, the multimedia messages are encapsulated with the *Simple Object Access Protocol* (SOAP) format [38] and attached in the payload of an HTTP message.

[2] The Presence SCS is a watcher responsible to query the user presence information from the network.

[3] The SIP Presentity maintains the presence information for a SIP user.

[4] The XML schema (consisting of elements and attributes) defines the structure of an XML document.

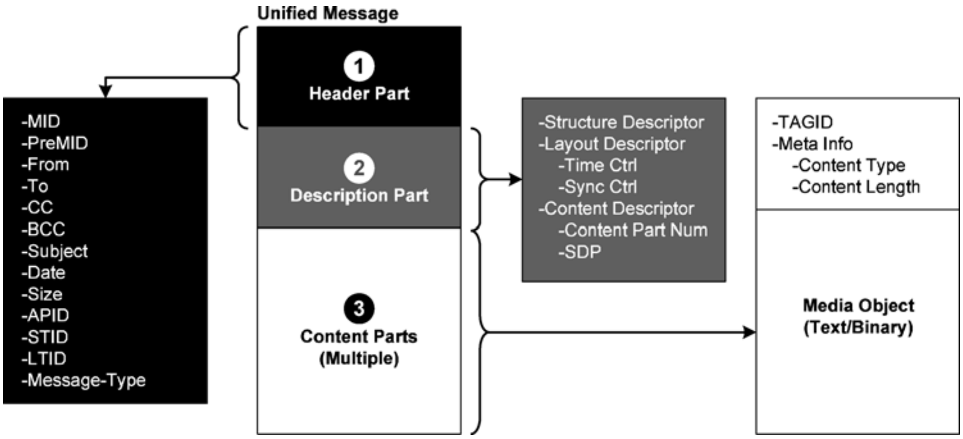


**Fig. 2** The structure of a unified message

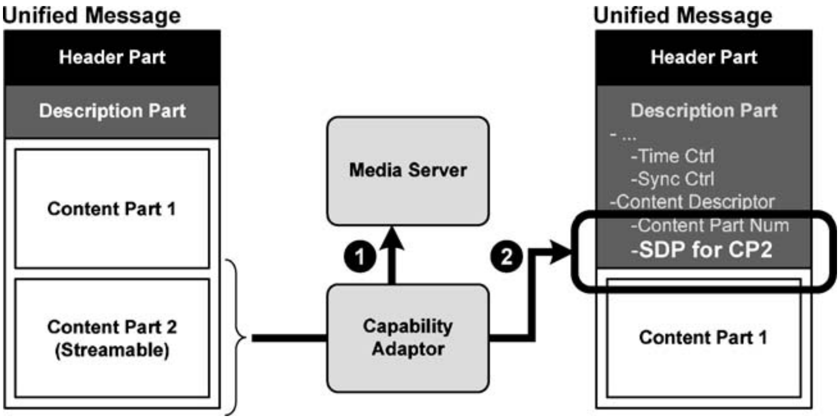


**Fig. 3** Process for streaming contents

terminal device. For this part, we define a description language compatible with other description languages (e.g., *Multipurpose Internet Mail Extensions* (MIME) [11–15] for the E-mail service, *Synchronized Multimedia Integration Language* (SMIL) [43] for MMS, and XHTML *Mobile Profile* (XHTML MP) [36] for iMMS). A unified message may have one or more content parts. The content part (Fig. 2(3)) contains the meta information and the media objects (which may be text-based or binary-based, e.g., images and video/audio clips) of the message.

*Capability Adaptor* (Fig. 1(6)) adapts the description and content parts of a unified message to the format that can be displayed on the terminal device. The adaptation consists of two stages. At the first stage, the Capability Adaptor queries the OMA UAProf[5] stored in the Global Information Manager (to be elaborated later). Based on the queried results, the Capability Adaptor adjusts the content part of the message (including the image size and video/audio transcoding) and updates the meta-information. At the second stage, the Capability Adaptor adjusts the description part of the message according to the updated meta-information at the first stage. Note that the previous work for message adaptation such as [8] only handles the content part adjustment of a message, and leaves the description part unchanged. This limits the message adaptation to only support terminal devices using the same description language.

*Media Server* (Fig. 1(5)) stores the streamable contents (e.g., video or audio files) and plays the files for the users. As shown in Fig. 3, the Capability Adaptor adjusts a message (by extracting the streamable contents from the original message), moves them into the Media Server (Fig. 3(1)), and then inserts a *Session Description Protocol* (SDP) [16] descriptor into the description part of the message (Fig. 3(2)). When the terminal device receives a message with an SDP descriptor, it establishes a *Real-Time Streaming Protocol* (RTSP) [30] connection to the Media Server. The multimedia contents can be delivered to the terminal device through the RTSP connection.

*Interactive Server* (Fig. 1(4)) processes the OI-HN requests from the connectors based on our designed protocol, *Unified Message Interaction Protocol* (UMIP), and then forwards the requests to the application services through the OI-AP interface. The details of UMIP will be discussed in Section 3.

[5] The OMA UAProf is used to specify the capability of a mobile terminal. For the details of the OMA UAProfile, readers may refer to [34].

**Table 2** The OI-AP interface

| Function name | Intended usage |
|---|---|
| InteractiveServer.AP_Dep_REQ() | to register a application service on the UMM platform |
| APService.AP_Sub_IND() | to inform the application service about the user's subscription of an application service |
| APService.Msg_Init_IND() | to request the templates (of an iMM message) from the application service |
| APService.Msg_Forward_IND() | to request the application service to derive the message receiver |

*Global Information Manager* (Fig. 1(8)) is a database which maintains *User Profiles* (UserProfs) for the subscribers. A UserProf record consists of three types of information:

- *User-Related Information* includes user identities, the presence information, user preference setting, and privacy control. Each user may have multiple user identities, one for each type of message service, including *Unified ID* (UID; used in the UMM platform to identify a user), MSISDN (phone number; used in MMS), SIP address (used in the instant message service), and E-mail address. The presence information (e.g., on-line or off-line) indicates whether the user is on the instant message service or not. The user can set the priority for different kinds of message services through the preference setting. By setting privacy control, users can specify which personal information will be released to application services.
- *Terminal information* includes the capability and battery status of a terminal device. The terminal capability (including user interface, communication protocol, computing capability, maximum size of a message, the type of streamable message content) is encapsulated in the form of OMA UAProf.
- *Service information* pertains to the services the user subscribes to, including authentication and customization data.

Figure 4(1) shows an example for a UserProf record, where the user has two kinds of terminal devices, MMS *User Agent* (UA) and IM UA.

Note that the UMM platform takes the ENUM-based approach [10] to manage user's different accounts. When a user subscribes to the UMM platform, a UID is assigned to the user, and all of his contact addresses will be associated with this UID. The UID will be used to identify the user in the UMM platform.

*Policy Unit* (Fig. 1(7)) stores the policies for the UMF algorithm. For each entry in a UserProf record, we assign a priority to it. The values of the priorities are stored in the policy. The priority can be either set by users or specified by application services, whose values are stored in U-Policy and S-Policy, respectively, as shown in Fig. 4(2). The UMF algorithm references the Policy Unit to select a suitable target device for the message receiver. Details of UMF are given in Section 5.

### 2.3 Application part

The Application Part (Part3, Fig. 1) contains application services (Fig. 1(10)) created by the third-party service providers. These application services can be installed on the UMM platform through the OI-AP interface (Fig. 1(9)). Table 2 lists the functions of the OI-AP interface and their corresponding usages.

## 3 The unified message interaction protocol

This section describes the UMIP protocol, which consists of four procedures: user registration, service subscription, terminal information update, and message forward procedures. The user registration procedure is for a user to register on UMM, and get a UID. The service subscription procedure is for a user to subscribe to an application service on UMM. The terminal information update procedure is responsible for updating users' presence and terminal capability information. The message forward procedure deliveries a message through the UMM platform. Due

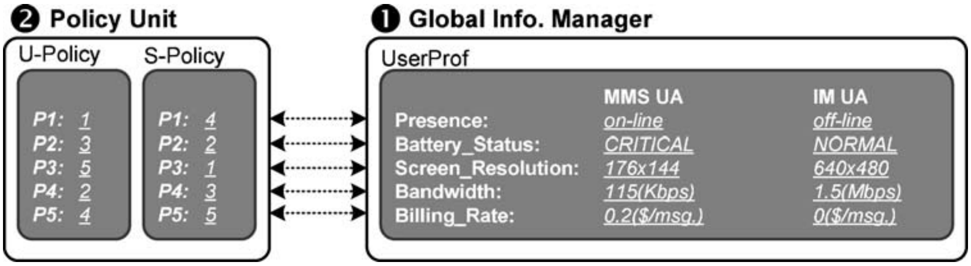

❷ Policy Unit
U-Policy
S-Policy
P1: 1
P2: 3
P3: 5
P4: 2
P5: 4
P1: 4
P2: 2
P3: 1
P4: 3
P5: 5
❶ Global Info. Manager
UserProf
MMS UA
IM UA
Presence: on-line off-line
Battery_Status: CRITICAL NORMAL
Screen_Resolution: 176x144 640x480
Bandwidth: 115(Kbps) 1.5(Mbps)
Billing_Rate: 0.2($/msg.) 0($/msg.)

**Fig. 4** An example for the policy unit and global information manager

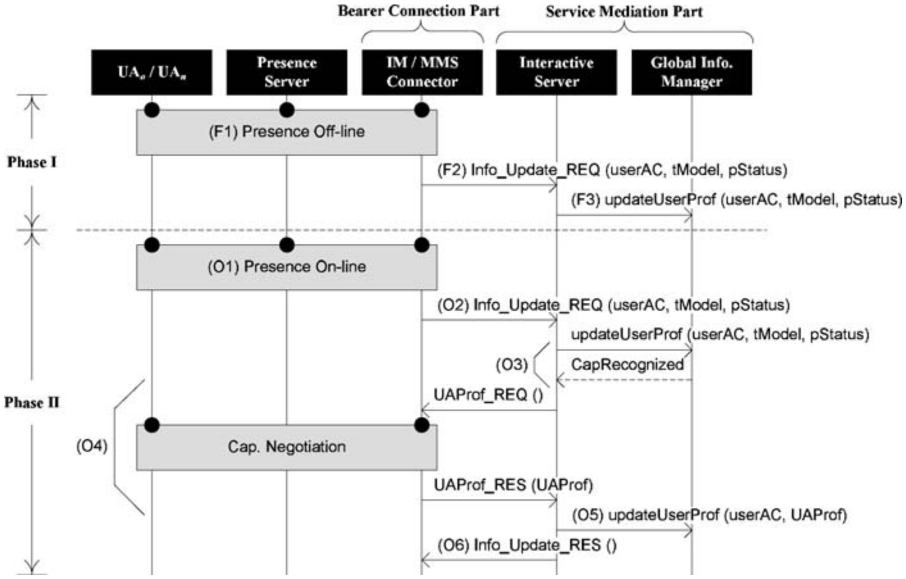


**Fig. 5** The message flow for the terminal information update procedure

to the page limitation of this paper, the descriptions for user registration and service subscription procedures are omitted. Details of the terminal information update and message forward procedures are given in the following subsections.

### 3.1 The terminal information update procedure

The terminal information update procedure is exercised whenever a user turns on or turns off the UA of the terminal device. We use a scenario to illustrate the procedure, where the user turns off an IM UA, $UA_o$, installed on a PDA, and then turns on an MMS UA, $UA_n$, installed on a mobile terminal. For other scenarios, the behavior of the procedure is similar, which is not presented in this paper. Figure 5 illustrates the message flow for the terminal information update procedure that consists of two pahses. The details are explained below.

*Phase I.* This phase is executed when the user turns off the IM UA, $UA_o$.

*Step F1.* When the user turns off $UA_o$, the $UA_o$ detaches from the Internet following the IETF Presence and Instant Messaging protocol [9]. The Presence Server (i.e., the SIP Presentity) is informed of $UA_o$'s turn off. The Presence Server reports this event to the IM Connector by sending a SIP message, NOTIFY.

*Step F2.* The IM Connector invokes the Info_Update_REQ(userAC, tModel, pStatus) function to request the Interactive Server to update the UserProf record, where userAC stores the user ID used in Internet (i.e., SIP address), tModel stores the terminal model number (i.e., a serial number of the PDA), and pStatus carries the presence status (in this case, pStatus = off-line).

*Step F3.* The Interactive Server calls the updateUserProf(userAC, tModel, pStatus) function of the Global Information Manager to update the presence status for the user. The Global Information Manager first finds out user's UID according to user's SIP address, then it retrieves the UserProf record by using the UID and updates the presence information field.

*Phase II.* This phase is executed when the user turns on the MMS UA, $UA_n$.

*Step O1.* When the user turns on $UA_n$, the $UA_n$ attaches to the cellular network through Peu interface defined in 3GPP 23.141 [5]. The Presence Server is informed of $UA_n$'s turn on. The Presence Server reports this event to the MMS Connector through the Pw interface defined in 3GPP 23.141 [5].

*Step O2.* This step is similar to Step F2, except that the parameter pStatus is set to on-line.

*Step O3.* Similar to Step F3, the updateUerProf(userAC, tModel, pStatus) function is invoked to update the presence information. The Global Information Manager determines whether the capability information of the device exists in the UserProf record by checking the tModel parameter. The result is reported to the Interactive Server by setting the CapRecognized flag. If CapRecognized is true (i.e., the capability information exists), then Step O6 is performed. Otherwise (i.e., CapRecognized is false), Steps O4 and O5 are executed to get the capability information of the mobile device.

*Step O4.* The Interactive Server calls the UAProf_REQ() function to get the OMA UAProf for the mobile

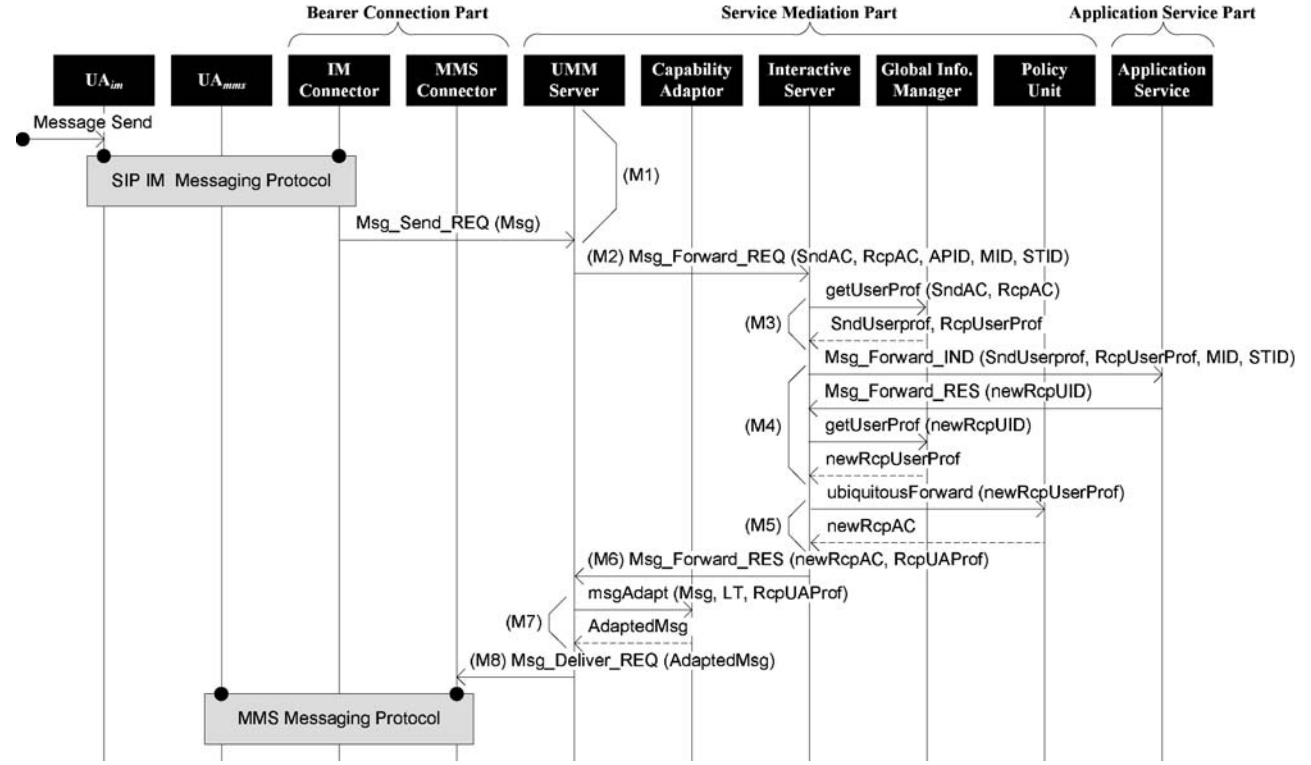


**Fig. 6** The message flow for the message forward procedure

device. Then the MMS Connector exchanges the OMA UAPRof with UA$_n$ through the MM1 interface [37]. After that, the MMS Connector returns the OMA UAProf to the Interactive Server by invoking the UAProf_RES(UAProf) function.

*Step O5.* The Interactive Server invokes the updateUserProf(userAC, UAProf) function to update the terminal capability field in the UserProf.

*Step O6.* The Interactive Server ends the procedure by invoking the Info_Update_RES() function.

## 3.2 The message forward procedure

The message forward procedure processes the delivery of a unified message. The message receiver may turn on multiple terminal devices. In this procedure, the UMF algorithm (to be elaborated in Section 5) is executed to select a suitable terminal device for the message receiver. To illustrate the message forward procedure, we consider the scenario where a message sender uses an IM UA, UA$_{im}$, to send a SIP instant message to the message receiver who activates three types of UAs, including an IM UA, an E-mail UA, and an MMS UA. For other scenarios, the behavior of this procedure is similar, and the description is omitted. Figure 6 illustrates the message flow, whose details are described as follows:

*Step M1.* The UA$_{im}$ sends an instant message to the IM connector through the SIP proxy server following the SIP IM protocol [9]. The IM Connector invokes the Msg_Send_REQ(Msg) function to deliver the message to the UMM Server, where Msg carries the instant message.

*Step M2.* The UMM server converts the incoming message to a unified message, and then invokes the Msg_Forward_REQ(SndAC, RcpAC, APID, MID, STID) function to process the header part of the message, where SndAC stores sender's SIP address, RcpAC stores receiver's SIP address, APID stores the application ID, MID carries the message ID, and STID is the ID for the service template (to be elaborate in Section 4).

*Steps M3.* The Interactive Server calls the getUserProf(SndAC, RcpAC) function of Global Information Manager to retrieve the UserProfs for the message sender and receiver.

*Step M4.* The Interactive Server invokes the Msg_Forward_IND(SndUserProf, RcpUserProf, MID, STID) function to request the application service to derive the UID of the message receiver, where the application service is determined by the APID obtained in Step M2. The derivation of the UID is done by the iMMS mechanism (to be elaborated in Section 4). The result is returned to the Interactive Server through the Msg_Forward_RES(newRcpUID)

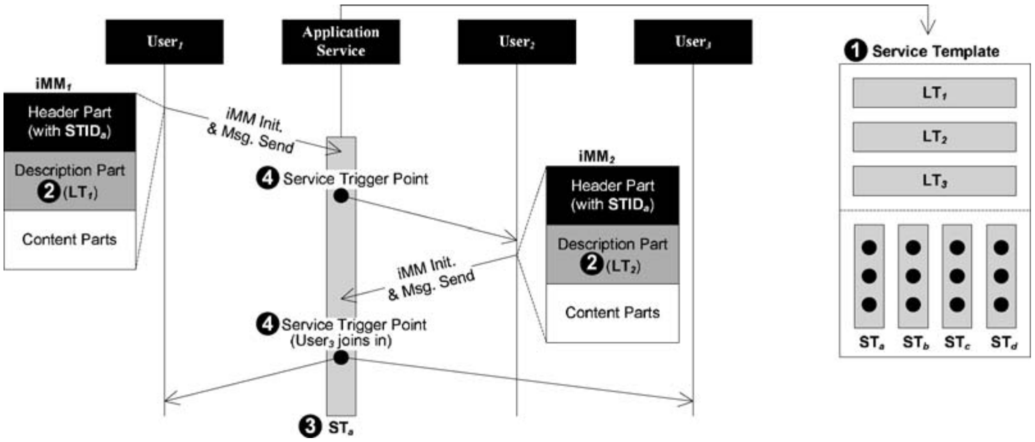


**Fig. 7** The iMMS framework

function call. The Interactive Server invokes the getUserProf(newRcpUID):newRcpUserProf function to get a UserProf by using the obtained UID, where the result is carried in newRcpUserProf.

*Step M5.* The Interactive Server calls the ubiquitousForward(newRcpUserProf) function to request the Policy Unit to select a target device for the message receiver. By using the newRcpUserProf, the Policy Unit finds out the corresponding policies, and then exercises the UMF algorithm. Assume that the MMS UA, UA$_{mms}$, is selected. The policy Unit responds the Interactive Server the corresponding account (i.e., MSISDN) for UA$_{mms}$, which is carried in the newRcpAC.

*Step M6.* The Interactive Server invokes the Msg_Forward_RES(newRcpAC, RcpUAProf) function to inform the UMM Server to change the account field in the message header as newRcpAC.

*Step M7.* The UMM Server calls the msgAdapt(Msg, LT, RcpUAProf) function to request the Capability Adaptor to adapt the description and content parts of the message so that it can be displayed on the target device.

*Step M8.* Based on the result (i.e., AdaptedMsg) obtained in Step M7, the UMM Server uses the XML-based unified schema to re-encapsulate the message as an MMS message, and then instructs the MMS Connector to deliver the message to the receiver by invoking the Msg_Deliver_REQ(AdaptedMsg) function.

## 4 The iMMS mechanism

An interactive application service may contain one or more actions. For each action, there is a series of corresponding message exchanges. We define a session as the series of message exchanges for an action. To more efficiently deliver these messages, we implement the iMMS mechanism on the application services. The iMMS mechanism is exercised before the user sends or replies a message. Figure 7 illustrates the iMMS framework. For an interactive application service, we define a service template as shown in Fig. 7(1). A service template consists of *Layout Templates* (LTs) and *Session Templates* (STs). An LT (Fig. 7(2)) provides a predefined description part. An ST (Fig. 7(3)) predefines the number of the service trigger points[6] (Fig. 7(4)) in a session and the operations at every trigger point. Figure 8 shows the message flow for iMMS, where a user uses the MMS UA, UA$_{mms}$, to get a service template from an application service. The details are given below.

*Step I1.* The UA$_{mms}$ sends an iMMS initialization request to the MMS Connector.

*Step I2.* Upon receipt of the iMM initialization message, the MMS Connector invokes the Msg_Init_REQ(userAC, APID, PreMID) function to forward the request to the Interactive Server, where PreMID carries the MID for the previous message.

*Step I3.* The Interactive Server calls the getUserprof(userAC) function of Global Information Manager to get the UserProf of the user. Then it invokes the Msg_Init_IND(UserProf, PreMID) function of the application service to select a service template (containing STs and LTs) for the message.

*Step I4.* According to the UserProf and PreMID, the application service generates a list containing a new MID, STs, and LTs, and then returns the list to the UA$_{mms}$ by executing the Msg_Init_RES(MID, STOption, LTOption) function of the MMS Connector.

After getting the service template, the user can select one ST and LT in the service template and then creates a message.

[6] The service trigger point specifies the necessary operations (e.g., to invite a user to join the message communication) that should be done at a particular session stage.

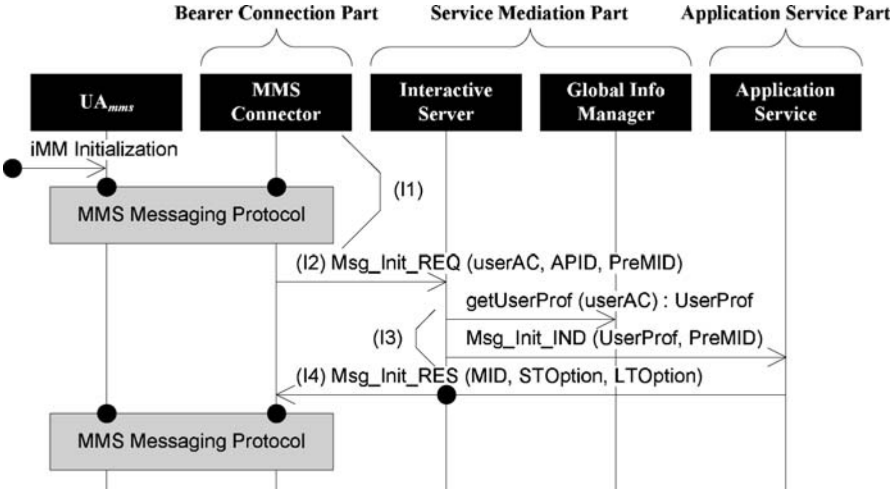


**Fig. 8** The message flow for the iMM mechanism

## 5 The UMF algorithm

The UMF algorithm is performed by the Policy Unit to select a target device for the message receiver. The values of all five fields (including Presence, Battery_Status, Screen_Resolution, Bandwidth, and Billing_Rate as shown in Fig. 4 (1)) of terminal capability information in a UserProf record are referenced. The inputs of the UMF algorithm include a deviceList (containing the names of the terminal devices the user has), UserProf record, U-Policy, and S-Policy for the user. Figure 9 illustrates the flow chart for the algorithm, whose description is given below:

In Steps 1 and 2, the UMF gets the inputs and sets the counter $i$ to 1. This counter is used for a "for" loop. In Step 3, the UMF queries the User_Preference field in the UserProf to determine whether the S-Policy or U-Policy will be referenced for the priorities of the five fields of terminal capability. In Step 4, UMF sorts the names of the five fields according to the priority values (obtained in Step 3) and generates a set $S = \{F_1, F_2, F_3, F_4, F_5\}$ where $F_i$ is the field name. In Step 5, for each device, $D_j$, in the deviceList, UMF retrieves its value corresponding to $F_i$, which is denoted as $v_{i,j}$. Step 6 checks the name of $F_i$. Then one of the following five cases will be executed:

Presence. Step 7 excludes the device that has the value $v_{i,j} =$ **off-line**.

Battery_Status. Step 8 excludes the device that has the value $v_{i,j} =$ **BATTERY_CRITICAL**.

Billing_Rate. Step 9 selects a device that has the minimum billing rate stored in $v_{i,j}$. Two kinds of rates are considered, that is, $\mathbf{BRate}_N$ (i.e., billing rate in the barrier network) and $\mathbf{BRate}_S$ (i.e., billing rate of the application service). Step 10 excludes the unselected devices from the deviceList.

Screen_Resoultion. Step 11 selects a device that has the maximum screen resolution, $\mathbf{Pixel}_x \times \mathbf{Pixel}_y$. Step 12 excludes the unselected devices from the deviceList.

Bandwidth. Step 13 selects a device that has the maximum transmission rate, i.e., **DRate**. Step 14 excludes the unselected devices from the deviceList.

In Step 15, we set $i \leftarrow i + 1$ for the next loop. Step 16 checks whether there is only one device in the deviceList. If so, Step 18 returns the device as the target device, and UMF ends. Otherwise (i.e., more than one device in the deviceList), UMF runs the next loop. Step 17 checks whether the five fields have been checked. If so, all devices in the deviceList are returned, and UMF ends. Note that **BATTERY_CRITICAL**, **DRate**, and **BRate** are defined in OSA MExE QoS API [2].

## 6 Performance evaluation

The signal of UMM is delivered through the message services in the heterogeneous networks. The delay of the message processing in the heterogeneous networks may affect the signal delivery for the terminal information update procedure. The Global Information Manager may provide obsolete information during the transition of the update of users' presence and terminal capability information. Consider that a user $u$ changes from the message service, $M_A$, to the message service, $M_B$. User $u$ turns off the UA of $M_A$ (by executing Phase I of the terminal information update procedure), and then turns on the UA of $M_B$ (by executing Phase II of the terminal information update procedure) as described in Section 3. Let $T$ be the period when Phases I and II are executed. During $T$, since the Global Information Manager has not been updated, any message termination to $u$ can not be delivered to $u$ instantly.

It is important to see how the delay of the message processing in the heterogeneous networks affects transmission of message terminations to $u$. Suppose that the message terminations to a message service for $u$ form a Possion distribution

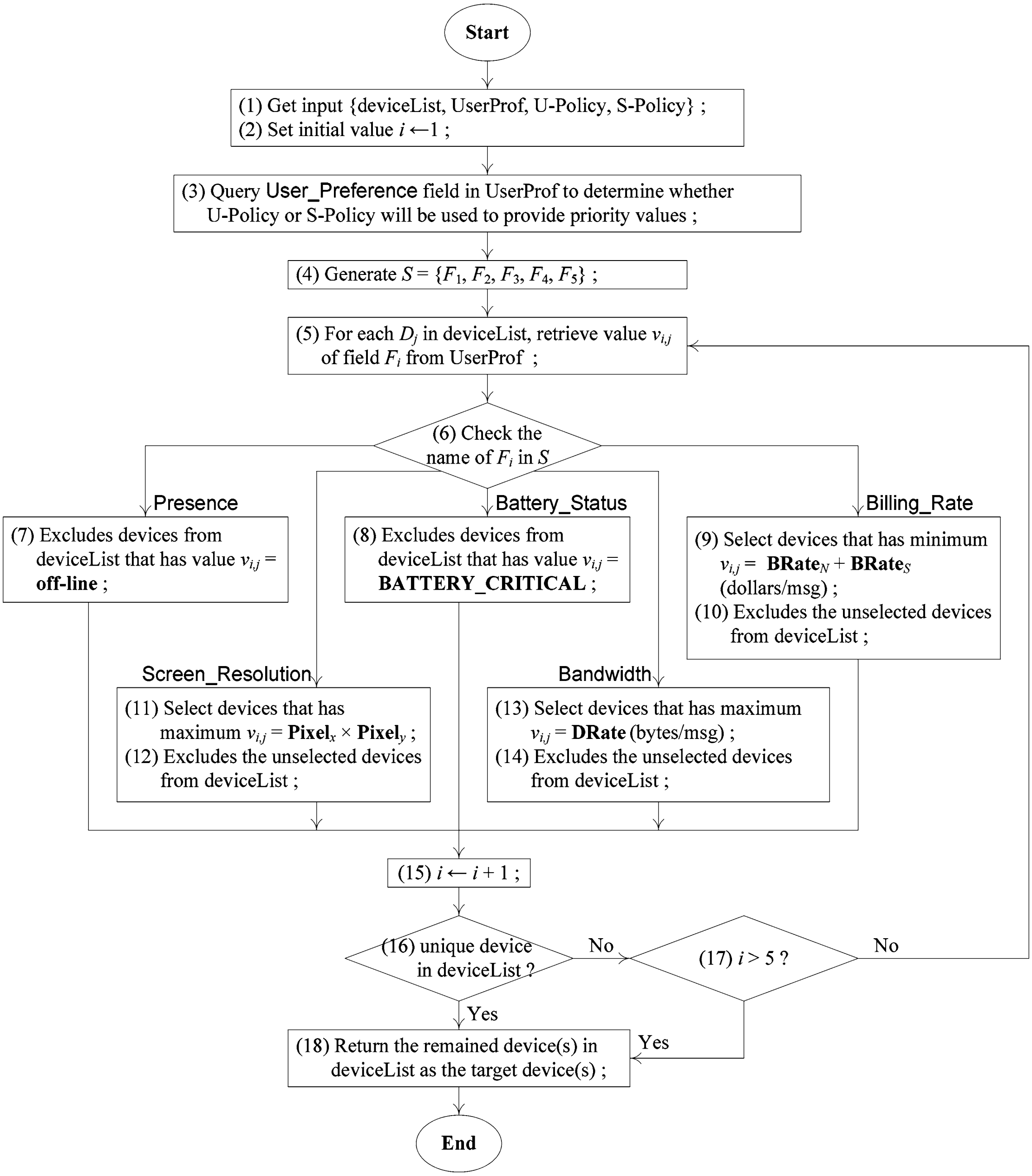

Start
(1) Get input {deviceList, UserProf, U-Policy, S-Policy} ;
(2) Set initial value $i \leftarrow 1$ ;
(3) Query User_Preference field in UserProf to determine whether U-Policy or S-Policy will be used to provide priority values ;
(4) Generate $S = \{F_1, F_2, F_3, F_4, F_5\}$ ;
(5) For each $D_j$ in deviceList, retrieve value $v_{i,j}$ of field $F_i$ from UserProf ;
(6) Check the name of $F_i$ in $S$
Presence
Battery_Status
Billing_Rate
(7) Excludes devices from deviceList that has value $v_{i,j}$ = off-line ;
(8) Excludes devices from deviceList that has value $v_{i,j}$ = BATTERY_CRITICAL ;
(9) Select devices that has minimum $v_{i,j}$ = BRate$_N$ + BRate$_S$ (dollars/msg) ;
(10) Excludes the unselected devices from deviceList ;
Screen_Resolution
Bandwidth
(11) Select devices that has maximum $v_{i,j}$ = Pixel$_x$ × Pixel$_y$ ;
(12) Excludes the unselected devices from deviceList ;
(13) Select devices that has maximum $v_{i,j}$ = DRate (bytes/msg) ;
(14) Excludes the unselected devices from deviceList ;
(15) $i \leftarrow i + 1$ ;
(16) unique device in deviceList ?
No
(17) $i > 5$ ?
No
Yes
Yes
(18) Return the remained device(s) in deviceList as the target device(s) ;
End

**Fig. 9** The flow chart for the UMF algorithm

with rate $\lambda$, and $T$ is a random variable with a Gamma density function $f(t)$ with mean $1/\mu$, standard deviation $\sigma$, and Laplace transform $f^*(s)$ where

$$f^*(s) = \left(\frac{\mu}{\mu + \mu^2\sigma^2 s}\right)^{1/(\mu\sigma)^2}. \tag{1}$$

Note that our derivation applies to any message delay distribution whose Laplace transform has closed forms. The Gamma distribution is selected to present the message processing delay distribution because it can be used to approximate many distributions as well as the measured data obtained from the PCS fields [19–24].

Let random number $K$ be the number of $u$'s message terminations to a message service during period $T$. From [20], the probability that $K = k$ during period $T = t$ can be expressed as

$$\Pr[K = k \mid T = t] = \frac{(\lambda t)^k}{k!} e^{-\lambda t}.$$

Then, the probability $\Pr[K = k]$ (that $k$ message terminations to a message service for $u$ during period $T$) can be obtained as follows.

$$
\begin{aligned}
&\Pr[K = k]\\
&= \int_{t=0}^{\infty} \Pr[K = k \mid T = t] f(t) dt\\
&= \left(\frac{\lambda^k}{k!}\right) \int_{t=0}^{\infty} t^k f(t) e^{-\lambda t} dt\\
&= \left(\frac{\lambda^k}{k!}\right) \left[(-1)^k \frac{d^k}{ds^k} f^*(s)\right]\Bigg|_{s=\lambda}\\
&= \left(\frac{\lambda^k}{k!}\right) \left[(-1)^k \frac{d^k}{ds^k} \left(\frac{\mu}{\mu + \mu^2\sigma^2 s}\right)^{1/(\mu\sigma)^2}\right]\Bigg|_{s=\lambda}\\
&= \binom{\frac{1}{(\mu\sigma)^2} + k - 1}{k} \left(\frac{\mu}{\mu + \mu^2\sigma^2 s}\right)^{1/(\mu\sigma)^2} \left(\frac{\mu^2\sigma^2\lambda}{\mu + \mu^2\sigma^2\lambda}\right)^k\\
&= \binom{\frac{1}{(\mu\sigma)^2} + k - 1}{k} (1 - x)^{1/(\mu\sigma)^2} x^k \qquad (2)
\end{aligned}
$$

where

$$x = \frac{\mu^2\sigma^2\lambda}{\mu + \mu^2\sigma^2\lambda}.$$

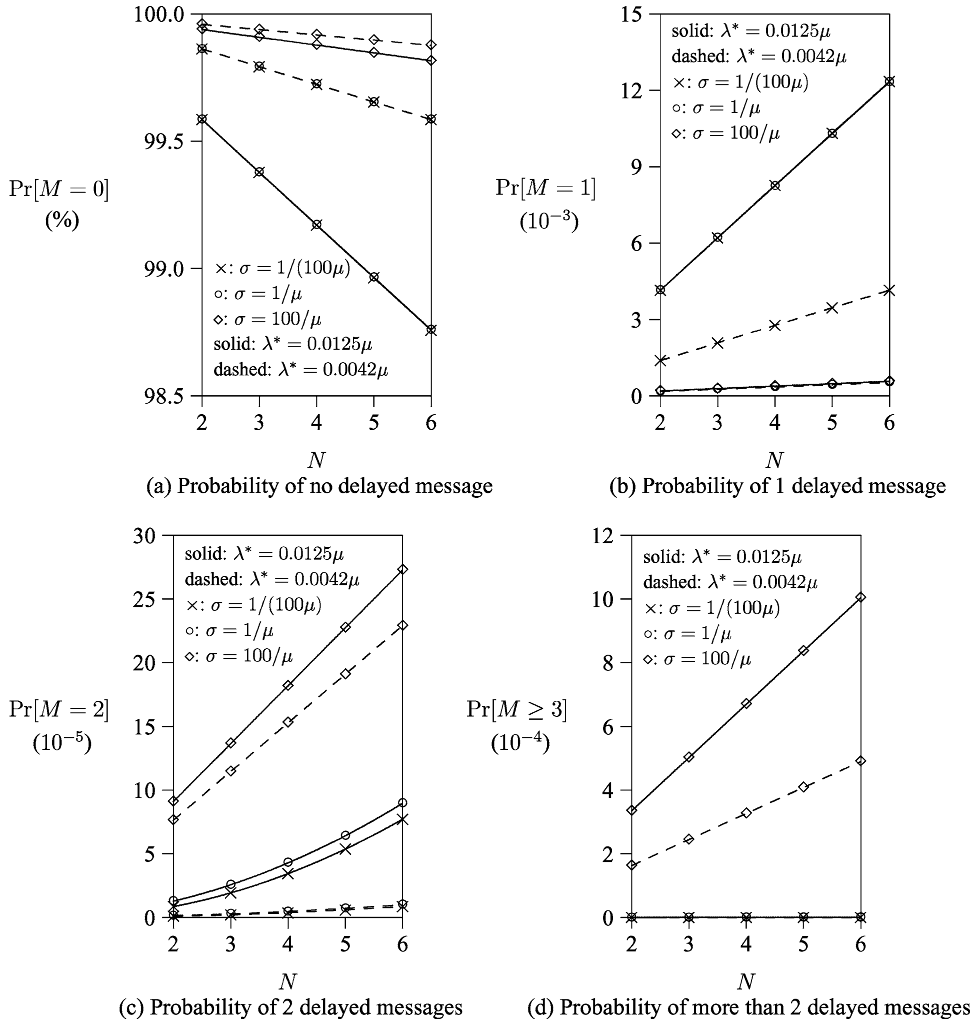
Suppose that $u$ subscribes $N$ message services. We use the term "delayed message" to represent the message termination that has been delayed due to the message processing in the heterogeneous networks. Let $\Pr[M = k]$ be the probability that $u$ has $k$ delayed messages. Thus, we have

$$\Pr[M = 0] = (\Pr[K = 0])^N , \qquad (3)$$

$$\Pr[M = 1] = \binom{N}{1} \Pr[K = 1] (\Pr[K = 0])^{N-1} , \qquad (4)$$

$$
\begin{aligned}
\Pr[M = 2] &= \binom{N}{2} (\Pr[K = 1])^2 (\Pr[K = 0])^{N-2}\\
&\quad + \binom{N}{1} \Pr[K = 2] (\Pr[K = 0])^{N-1} , \qquad (5)
\end{aligned}
$$

**Fig. 10** Effects of $\lambda^*$, $N$ and $\sigma$ on $\Pr[M]$



(a) Probability of no delayed message

(b) Probability of 1 delayed message

(c) Probability of 2 delayed messages

(d) Probability of more than 2 delayed messages and

$$\Pr[M \geq 3] = 1 - \sum_{i=0}^{2} \Pr[M = i]. \qquad (6)$$

Because the message termination traffic to a message service for $u$ is a Poisson with rate $\lambda$, the net message termination traffic to $u$ is $\lambda^* = N\lambda$.

In the following, we study the effects of the net message termination rate $\lambda^*$, the standard deviation $\rho$ of message processing delay, and the number $N$ of message services on the $\Pr[M]$ performance, where $\lambda^*$ and $\rho$ are normalized by $\mu$. Suppose that there is one message termination per hour to $u$. We consider two scenarios: In the first scenario, $u$ turns off an MMS UA (where the average messaging processing time is 40 seconds) and then turns on an IM UA (where the average message processing time is 5 seconds). The average total message processing time is 45 seconds. We set $\lambda^* = 0.0125\mu$. In the second scenario, $u$ turns off an IM UA and then turns on an E-mail UA (where the messaging processing time is 10 seconds). The average total message processing time is 15 seconds. We set $\lambda^* = 0.0042\mu$. For other scenarios, we observe similar results, which are not shown in this paper. Using (3)–(6), the following phenomena are observed:

*Effects of* $\lambda^*$. Figure 10 plots $\Pr[M = k]$ (where $k \geq 0$) as a function of $N$, where $\lambda^* = 0.0125\mu$ and $\lambda^* = 0.0042\mu$. For all cases, $\Pr[M = 0] > 98.5\%$ for $\lambda^* = 0.0125\mu$ (see solid lines), and $\Pr[M = 0] > 99.5\%$ for $\lambda^* = 0.0042\mu$ (see dashed lines). In other words, in Scenario 2 (i.e., $\lambda^* = 0.0042\mu$), the delayed message probability can be ignored. In Scenario 1 (i.e., $\lambda^* = 0.0125\mu$), the delayed message probability is reasonably small, which is less than 1.5%. $\Pr[M = 2]$ and $\Pr[M \geq 3]$ are in the order of $10^{-5}$ and $10^{-4}$, respectively. Thus the probability for multiple delayed termination messages can be ignored in all cases.

*Effects of* $N$. Figure 10 also shows that $\Pr[M = 0]$ decreases as $N$ increases. This phenomenon can be explained by using the stochastic phenomenon [18]. With a fixed net arrival rate, if there are more arrival sources (i.e., more message services), it is more likely to observe arrivals during a fixed period. Thus, more message services the user subscribes will degrade the performance. However, this effect is not significant. For example, when $N = 5$, by adding one more message service, $\Pr[M = 0]$ is degraded by 0.2% (for $\lambda^* = 0.0125\mu$) and 0.02% (for $\lambda^* = 0.0042\mu$). To summarize, the performance degradation due to adding extra message services can be ignored.

*Effects of* $\sigma$. The standard deviation $\sigma$ of message processing delay determines the probability of observing long or short message processing times. When $\sigma > 1/\mu$, more long or short message processing periods are observed, which implies larger $\Pr[M = 0]$, $\Pr[M = 2]$, and $\Pr[M \geq 3]$. This analysis is consistent with our observation in Fig. 10. As $\sigma$ is larger than $1/\mu$, $\Pr[M = 0]$, $\Pr[M = 2]$, and $\Pr[M \geq 3]$ increase, and $\Pr[M = 1]$ decreases. Since $\Pr[M = 2]$ and $\Pr[M \geq 3]$ are very small in all cases, the effect of $\sigma$ can be ignored under the normal operational conditions.

## 7 Conclusion

We designed a *Ubiquitous and Unified Multimedia Messaging* (UMM) platform to integrate different message services. The UMM platform adopts a gateway-based approach. Our design does not modify the protocols of the message services and need not involve the network operator. The UMM platform accommodates the UMF algorithm and iMMS mechanism to select a suitable message service for the message receiver and to efficiently deliver messages between the applications and the end users, respectively.

In our design, it is important to investigate if the processing delay (in the heterogeneous networks) for UMM signaling will cause delay to message termination delivery. We used an analytical model to investigate the performance of the designed platform. Our study indicated the following. For most of use scenarios, the probability for the delayed messages can be ignored (less than 1.5%) in our study. We may ignore the performance degradation due to adding extra message services (the degradation is less than 0.2% in our study). The effects of standard deviation $\sigma$ of the message processing delays can be ignored in our designed platform.

The current implementation of the UMM platform concentrates on provision of the UM concept and is a prototype. The UMM platform provides the privacy control setting (which is stored in the UserProf) for each user to prevent reveal of the sensitive information to the application services. Each connector of UMM implements a timeout monitor for failure message delivery. To deploy UMM as a public system, there may be more security, robustness, and failure-recovery issues that should be addressed, which will be considered as our future work.

**Acknowledgment** The authors would like to thank the editor and the three anonymous reviewers for their valuable comments. P. Lin's work was sponsored in part by the National Science Council (NSC), R.O.C., under the contract number NSC 94-2627-E-002-001- and NSC 95-2221-E-002-091-MY3, Ministry of Economic Affairs (MOEA), R.O.C., under contract number 93-EC-17-A-05-S1-0017, Chunghwa Telecom Labs, Microsoft, Telcordia Applied Research Center, Taiwan Network Information Center (TWNIC), and Iventec Appliances Corp.

## References

1. 3GPP, 3rd Generation Partnership Project; Technical Specification Group Terminals; Presence Servece; Architecture and Functional Description. Technical Report 3GPP TS 23.841 version 6.0.0, 2002.

2. 3GPP, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Mobile Execution Environment (MExE); Functional description. Technical Report 3GPP TS 23.057 version 6.2.0, 2003.
3. 3GPP, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Messaging Service (MMS); Service Aspects. Technical Report 3GPP TS 22.140 version 6.6.0, 2004.
4. 3GPP, 3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional Description. Technical Report 3GPP TS 23.140 version 6.7.0, 2004.
5. 3GPP, 3rd Generation Partnership Project; Technical Specification Group Terminals; Presence Service; Architecture and functional description. Technical Report 3GPP TS 23.141 version 6.6.0, 2004.
6. 3GPP, 3rd Generation Partnership Project; Technical Specification Group Terminals; Virtual Home Environment (VHE)/Open Service Access (OSA). Technical Report 3GPP TS 23.127 version 6.1.0, 2004.
7. S. Arbanowski and S. van Der Meer, Service personalization for unified messaging systems, in: *Proceedings of IEEE International Symposium on Computers and Communications* (July 1999) pp. 156–163.
8. S. Coulombe and G. Grassel, Multimedia adaptation for the multimedia messaging service, IEEE Communications Magazine, 42 (July 2004) 120–126.
9. M. Day, J. Rosenberg, and H. Sugano, A Model for Presence and Instant Messaging, Technical Report RFC 2778 (February 2000).
10. P.E. Faltstrom, 164 Number and DNS. Technical Report RFC 2916 (September 2000).
11. N. Freed and N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples. Technical Report RFC 2049 (November 1996).
12. N. Freed and N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures. Technical Report RFC 2048 (November 1996).
13. N. Freed and N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. Technical Report RFC 2045 (November 1996).
14. N. Freed and N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Three: Message Header Extensions for Non-ASCII Text. Technical Report RFC 2047 (November 1996).
15. N. Freed and N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. Technical Report RFC 2046 (November 1996).
16. M. Handley and V. Jacobson, Session Description Protocol (SDP). Technical Report RFC 2327 (April 1998).
17. P. Healy, D. Barber, and B. Nolan, Developing unified messaging system Apps in Java, in: *Proceedings of the International Conference on Principles and Practice of Programming in Java*, Vol. 52, no. 2 (June 2003) pp. 137–138.
18. L. Kleinrock, *Queuing Systems: Volume I—Theory* (Wiley, New York, 1976).
19. P. Lin, and Y.-B. Lin, Channel allocation for GPRS, IEEE Transactions on Vehicular Technology 50(2) (2001) 375–387.
20. P. Lin and Y.-B. Lin, Implementation and performance evaluation for mobility management of a wireless PBX network, IEEE Journal on Selected Areas in Communications 19(6) (2001) 1138–1146.
21. P. Lin and Y.-B. Lin, Channel allocation for GPRS with buffering mechanisms, ACM Wireless Networks 9(5) (2003) 431–441.
22. P. Lin, C.-H. Gan, and C.-C. Hsu, OVSF code channel assignment with dynamic code set and buffering adjustment for UMTS, IEEE Transactions on Vehicular Technology 54(2) (2005) 591–602.
23. P. Lin, W.-R. Lai, and C.-H. Gan, Modeling opportunity driven multiple access in UMTS, IEEE Transactions on Wireless Communications 3(5) (2004) 1669–1677.
24. P. Lin, Y.-B. Lin, and I. Chlamtac, Modeling frame synchronization for UMTS high-speed downlink packet access, IEEE Transactions on Vehicular Technology 52(1) (2003) 132–141.
25. Z. Miao, R. Impey, and Y. Baozong, Seamless messaging in a multimedia network environment, IEEE International Conference on Signal Processing 2 (2002) 1031–1034.
26. M.-E. Mostafa, MMS—the modern wireless solution for multimedia messaging, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 5 (Septmber 2002) 2466–2472.
27. J. Myers, C. Mellon, and M. Rose, Post Office Protocol Version 3, Technical Report RFC 1939 (May 1996).
28. N. Patel, *Multimedia Messaging Forecast (2003–2008)* Germany, Ireland and the UK, Strategy Analytics (November 2003).
29. Radicati Group, *Instant Messaging and Presence Market Trends, 2003–2007, http://www.radicati.com* (2003).
30. H. Schulzrinne, U. Columbia, A. Rao, and R. Lanphier, Real Time Streaming Protocol (RTSP). Technical Report RFC 2326 (April 1998).
31. H. Sinnreich and A. Johnston, *Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol*. (John Wiley & Sons, Inc., New York, 2001).
32. J.-M.S. Wams and M. van Steen, Pervasive Messaging, in: *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications* (March 2003) pp. 499–504.
33. J.-M.S. Wams and M. van Steen, Unifying user-to-user messaging system, IEEE Internet Computing 8 (2004) 76–82.
34. WAP Forum, WAG UAProf. Technical Report WAP-248-UAPROF-20010530-p (2001).
35. WAP Forum, Wireless Application Protocol; MMS Architecture Overview. Technical Report WAP-205-MMSArchOverview-20010425-a (2001).
36. WAP Forum, XHTML Mobile Profile. Technical Report WAP-277-XHTMLMP-20011029-a (2001).
37. WAP Forum, MMS Client Transactions Specification. Technical Report WAP-206-MMSCTR-20020115-a (2002).
38. World Wide Web Consortium (W3C), SOAP Version 1.2 Part 1: Messaging Framework. Technical Report W3C Recommendation (June 2003).
39. World Wide Web Consortium (W3C), Extensible Stylesheet Language (XSL) Version 1.1. Technical Report W3C Working Draft (December 2004).
40. World Wide Web Consortium (W3C), XML Schema Part 0: Primer, 2nd edn. Technical Report W3C Recommendation (October 2004).
41. World Wide Web Consortium (W3C), XML Schema Part 1: Structures, 2nd edn. Technical Report W3C Recommendation (October 2004).
42. World Wide Web Consortium (W3C), XML Schema Part 2: Datatypes, 2nd edn. Technical Report W3C Recommendation (October 2004).
43. World Wide Web Consortium (W3C). Synchronized Multimedia Intergration Language (SMIL 2.0), 2nd edn. Technical Report W3C Recommendation (January 2005).
44. World Wide Web Consortium (W3C), XML Schema: Component Designators. Technical Report W3C Working Draft (March 2005).
45. World Wide Web Consortium (W3C), XSL Transformations (XSLT) Version 2.0. Technical Report W3C Working Draft (April 2005).

**Phone Lin (M'02-SM'06)** received his BSCSIE degree and Ph.D. degree from National Chiao Tung University, Taiwan, R.O.C. in 1996 and 2001, respectively. From August 2001 to July 2004, he was an Assistant Professor in Department of Computer Science and Information Engineering (CSIE), National Taiwan University, R.O.C. Since August 2004, he has been an Associate Professor in Department of CSIE and in Graduate Institute of Networking and Multimedia, National Taiwan University, R.O.C. His current research interests include personal communications services, wireless Internet, and performance modeling. Dr. Lin is an Associate Editor for IEEE Transactions on Vehicular Technology, a Guest Editor for IEEE Wireless Communications special issue on Mobility and Resource Management, and a Guest Editor for ACM/Springer MONET special issue on Wireless Broad Access. He is also an Associate Editorial Member for the WCMC Journal. Dr. Lin has received many research awards. He was a recipient of Research Award for Young Researchers from Pan Wen-Yuan Foundation in Taiwan in 2004, a recipient of K. T. Li Young Researcher Award honored by ACM Taipei Chapter in 2004, a recipient of Wu Ta You Memorial Award of National Science Council (NSC) in Taiwan in 2005, a recipient of Fu Suu-Nien Award of NTU in 2005 for his research achievements. Dr. Lin is listed in WHO'S WHO IN SCIENCE AND ENGINEERING(R) in 2006. Dr. Lin is a Senior Member, IEEE. P. Lin's email and website addresses are plin@csie.ntu.edu.tw and http://www.csie.ntu.edu.tw/∼plin, respectively.

**Shan-Hung Wu** received the B.S. degree from Department of Information Management, National Central University, Jhongli, Taiwan, and M.S. degree from Department of Computer Science and Information, National Taiwan University, Taipei, Taiwan. He is currently a Ph.D. candidate at Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan. His research interests include distributed data management, pervasive computing, wireless and sensor networks, and performance modeling.

**Chung-Min Chen** is Director of Telcordia Applied Research Center in Taiwan. His research interests span across areas in distributed computing, data engineering, telecommunication and network management. He received a Ph.D. in Computer Science from the University of Maryland, College Park and a B.S. in Computer Science and Information Engineering from National Taiwan University.

**Ching-Feng Liang** received M.S. degree in electronic engineering from National Taiwan University of Science & Technology (NTUST) in 1993 and joined the Information & Communication Laboratory (ICL) of Industrial Technology Research Institute (ITRI) as an engineer. Liang has led more than 10 projects of Taiwan Ministry of Economic Affairs (MoEA) to study and develop the technologies of mobile network and services including GPRS/3G core network, WLAN/Cellular interworking and number portability service. Liang received the ITRI Award in 2005 and the Outstanding Project Award of Taiwan MoEA in 2003. Liang is currently the manager of the Core Network Department of ICL/ITRI.