BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **8**, No 3

Sofia • 2008

Integrating Non Call-Related User Interactions in SIP Environment

Ivaylo Atanasov, Evelina Pencheva

Department of Telecommunications, Technical University of Sofia, 1000 Sofia

Abstract: *Session Initiation Protocol (SIP) is used for multimedia control in next generation networks. The paper presents a research on implementing non call-related user interactions in SIP-based networks. The SIP is considered as a mediator between user interactions in packet networks and third party applications. A mapping between messages of protocols used for delivery of short messages in packet networks, SIP messages and methods of User Interaction Open Service Access' open interfaces is suggested. Different scenarios for handling SIP user interaction sessions are examined.*

Keywords: *User interactions, Open Service Access, Session Initiation Protocol.*

I. Introduction

Next Generation Networks (NGN) will employ IP technology for transport of voice, data, and multimedia. The telephony, mobile, and data networks of today will not disappear in this setting, but will be seen more as access networks that plug into the IP core network. Of course, this requires some kind of adaptation, for example gateways or inter-working units. IP is likely to be an integrating technology on network level. The control protocol used for establishing multimedia session is supposed to be Session Initiation protocol (SIP) [6].

One of the main features of NGN is considered to be the third party application control. External applications accesses network functions through application programming interfaces via secure framework. Many services will derive an added value from combining different network functions. Among the technologies that provide open access to network functions is Open Service Access (OSA). The OSA defines UMTS service architecture that enables application developers to make use of network functionality through open standardized interfaces, i.e. the OSA API's. The OSA API's allow for 3rd party applications access to call control, data session control, mobility, user interaction [1], charging and other network functions. Recommendations for implementing OSA define mapping of OSA interfaces onto network protocols. User Interaction API to Short Message Service (SMS) mapping [2] is among the mappings currently defined for OSA interfaces.

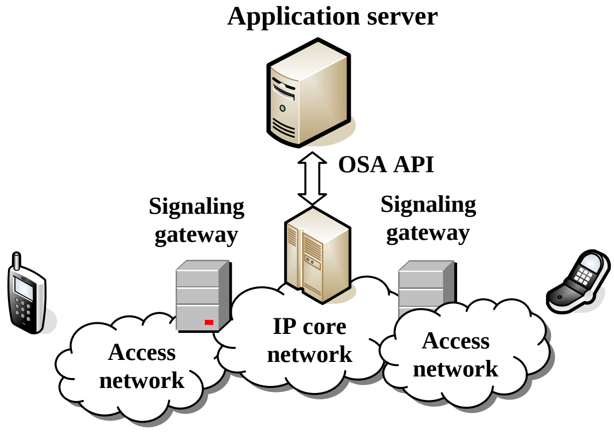

Fig. 1. Open access to network functions

The User Interaction interface is used by applications to interact with end users. The API supports call and non-call related User Interaction. In mapping the User Interaction interface to SMS call flows, only non-call related User Interaction is applicable.

Two alternatives may be used in cellular networks to provide SMS service:

- Mobile Application Part (MAP) signaling for interactions between SMS-G/IWMSC and mobile station and
- CAMEL Application Protocol (CAP) signaling for interactions between gsmSCF and gsmSSF.

The SIP protocol is developed to set up, modify and terminate sessions, but it also may be used in applications that have little or nothing to do with telephony such as instant messaging. Still, there is no mapping of SIP messaging functionality onto OSA User Interaction API.

The paper presents a view of how the SIP protocol can be used as an integrating protocol for provisioning non-call related user interaction service in cellular networks and OSA applications. Firstly in the paper we present a possible architecture for integrating non-call related user interactions in SIP environment. Then we investigate alternatives for handling SIP user interaction sessions. A mapping between MAP/CAP messages, SIP messages and OSA User Interaction API's methods is given. As an illustration, an example of 3rd party user interaction control is presented.

II. Integration of non-call related user interactions in SIP environment

This section describes how SIP protocol can be used to provide the generic user interaction service through open interfaces to mobile end users.

A. Instant messaging

Instant Messaging (IM), when SIP is as transport protocol, uses the MESSAGE request [5]. The parties engaged in an IM "conversation" exchange short messages in almost real-time. The participants do not establish a dialog by themselves, but the MESSAGE request might be sent within or outside a dialog. The message body as a MIME attachment carries the actual message content. The support of "text/plain" format is mandatory for user agents supporting MESSAGE request and optionally they might support other format as "text/html" or another. The delivery of MESSAGE request to the final destination normally is indicated by receiving a "200 OK" response. However, the message body of the "200 OK" response is not the one that contains the IM response, but a separate MESSAGE request is used. A "202 Accepted" response is indicating to the sending user agent that an intermediary hop is about to forward the message hopefully to the final one. Still, neither response is a confirmation that the message is shown to the end-user.

B. Inter-working between MAP/CAP and SIP

An inter-working unit, let call it Signaling Gateway (SG), is needed to handle MAP/CAP signaling and SIP signaling in case of interaction between cellular network and SIP network. The SG embodies MAP/CAP entity which provides MAP/CAP interface and SIP entity which provides SIP interface, as it is shown in Fig 2. The SG receives MAP/CAP messages, translates them to SIP messages and sends them to the next hop. Likewise, on receiving SIP messages, the SG translates them to the corresponding MAP/CAP messages. The mapping between MAP/CAP signaling and SIP signaling in the context of user interaction is shown in Table 1.

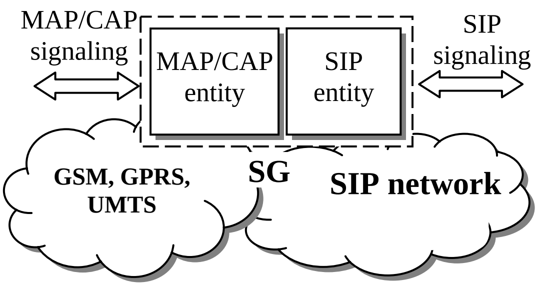

Fig. 2. MAP/CAP – SIP signaling gateway

The MAP entity may be SMS-G/IWMSC, and the CAP entity is gsmSCF. For triggering services as a result of messaging events, the MAP/CAP entity in the SG starts the CAMEL call-unrelated state machine for SMS. For mobile originating SMS, the MO SMS state model describes the actions in MSC and SGSN. For mobile terminating SMS, the MT SMS state model is used to describe the actions and states in the nodes [3]. Trigger detection points in the state models are defined to invoke service control.

Table 1. Mapping between OSA UI API methods, SIP messages and MAP/CAP messages

Original method/message		Translated original method/message		Translated translated method/message	
Direction	name	Direction	name	Direction	name
Application→ SCS-SIP	createUI	SCS-SIP→ SIP-gsmSCF		SIP-gsmSCF →gsmSSF	
Application→ SCS-SIP	createUI	SCS-SIP→ SIP-SMS-GIWMSC		SIP-SMS-G/IWMSC → MSC/HLR	
fgtApplication→ SCS-SIP	enableUI Notification	SCS-SIP→ SIP-gsmSCF	SUBSCRIBE	SIP-gsmSCF→HLR	MAP AnyTime Modification
Application→ SCS-SIP	disnableUI Notification	SCS-SIP→ SIP-gsmSCF	SUBSCRIBE	SIP-gsmSCF→HLR	MAP AnyTime Modification
gsmSSF→ SIP-gsmSCF	initialDPSMS	SIP-gsmSCF→ SCS-SIP	NOTIFY	SCS-SIP → Application	userInteraction EventNotify
gsmSSF→ SIP-gsmSCF	Dialog Abort (if appropriate)	SIP-gsmSCF→ SCS-SIP		SCS-SIP → Application	userInteraction Aborted
MSC/HLR→ SIP-SMS-GI/WMSC	Dialog Abort (if appropriate)	SIP-SMS-G/IWMSC → SCS-SIP		SCS-SIP → Application	userInteraction Aborted
gsmSSF→ SIP-gsmSCF		SIP-gsmSCF→ SCS-SIP		SCS-SIP → Application	userInteraction Notification Interrupted
MSC/HLR→ SIP-SMS-G/IWMSC		SIP-SMS-G/IWMSC → SCS-SIP		SCS-SIP → Application	userInteraction Notification Interrupted
gsmSSF→ SIP-gsmSCF		SIP-gsmSCF→ SCS-SIP	.	SCS-SIP → Application	userInteraction Notification Continued
MSC/HLR→ SIP-SMS-G/IWMSC		SIP-SMS-G/IWMSC → SCS-SIP		SCS-SIP → Application	userInteraction Notification Continued
gsmSSF→SIP-gsmSCF	Dialog Abort (if appropriate)	SIP-gsmSCF→ SCS-SIP		SCS-SIP → Application	userInteraction FaultDetected
MSC/HLR→ SIP-SMS-G/IWMSC	Dialog Abort (if appropriate)	SIP-SMS-G/IWMSC → SCS-SIP		SCS-SIP → Application	userInteraction FaultDetected
Application→ SCS-SIP	sendInfoReq	SCS-SIP→ SIP-SMS-GMSC	MESSAGE	SIP-SMS-GMSC →HLR HLR→ SIP-SMS-GMSC SIP-SMS-GMSC → MSC	MAP SendRouting InfoForSMreq MAP SendRouting InfoForSMConf MAP MTForward ShortMessage
Application→ SCS-SIP	sendInfoReq	SCS-SIP→ SIP-gsmSCF	MESSAGE	SIP-gsmSCF→ gsmSSF SIP-gsmSCF→ gsmSSF SIP-gsmSCF→ gsmSSF SIP-gsmSCF→ gsmSSF	RequestReport SMS Event (if appropriate) FurnishCharging InfoSMS (if appropriate) ConnectSMS (if appropriate) ContinueSMS (if appropriate)

Table 1. Mapping between OSA UI API methods, SIP messages and MAP/CAP messages (continue)

Original method/message		Translated original method/message		Translated translated method/message	
MSC→ SIP-SMS-GMSC	MAP MTForward SMConf	SIP-SMS-GMSC → SCS-SIP	2xx	SCS-SIP → Application	sendInfoRes
gsmSSF→ SIP-gsmSCF	EventReport SMS	SIP-gsmSCF→ SCS-SIP	2xx	SCS-SIP → Application	sendInfoRes
MSC/HLR→ SIP-SMS-GMSC	MAP SendRouting InfoForSMConf (error) MAP MTForward SMConf (error)	SIP-SMS-GMSC → SCS-SIP SIP-SMS-GMSC → SCS-SIP	4xx 5xx 6xx	SCS-SIP → Application	sendInfoErr
gsmSSF→ SIP-gsmSCF	EventReport SMS	SIP-gsmSCF→ SCS-SIP	4xx, 5xx, 6xx	SCS-SIP → Application	sendInfoErr
gsmSSF→ SIP-gsmSCF	TCAP ReturnError (if appropriate)	SIP-gsmSCF→ SCS-SIP	4xx, 5xx, 6xx	SCS-SIP → Application	sendInfoErr
Application→ SCS-SIP	sendInfoAnd CollectReq	SCS-SIP→ SIP-SMS-G/IWMSC	MESSAGE	SIP-SMS-G/IWMSC →HLR HLR→ SIP-SMS-G/IWMSC SIP-SMS-G/IWMSC →MSC	MAP SendRouting InfoForSMreq MAP SendRouting InfoFor SMConf MAP MTForward Short Message
Application→ SCS-SIP	sendInfoAnd CollectReq	SCS-SIP→ SIP-gsmSCF	MESSAGE	SIP-gsmSCF →gsmSSF SIP-gsmSCF →gsmSSF SIP-gsmSCF →gsmSSF	RequestReportSMS Event (if appropriate) FurnishCharging InfoSMS (if appropriate) ConnectSMS (if appropriate)
MSC→ SIP-SMS-IWMSC	MAP MOForward-ShortMessage	SIP-SMS-IWMSC → SCS-SIP	MESSAGE	SCS-SIP → Application	sendInfoAnd Collect Res
gsmSSF→ SIP-gsmSCF	EventReportSMS (if appropriate) InitialDPSMS	SIP-gsmSCF→ SCS-SIP	MESSAGE	SCS-SIP → Application	sendInfoAnd Collect Res
MSC/HLR→ SIP-SMS-GMSC	MAP SendRouting InfoForSMConf(error) MAP MTForward SMConf (error)	SIP-SMS-GMSC → SCS-SIP SIP-SMS-GMSC → SCS-SIP	4xx 5xx 6xx	SCS-SIP → Application	sendInfoAnd Collect Err
gsmSSF→ SIP-gsmSCF	EventReportSMS	SIP-gsmSCF→ SCS-SIP	4xx, 5xx, 6xx	SCS-SIP → Application	sendInfoAnd Collect Err
gsmSSF→ SIP-gsmSCF	TCAP Return Error (if appropriate)	SIP-gsmSCF→ SCS-SIP	4xx, 5xx, 6xx	SCS-SIP → Application	sendInfoAnd Collect Err

In case of sending a SIP message to another SIP entity, the SIP entity is involved in non-INVITE client transaction. In case of receiving a SIP message from another SIP entity, the SIP entity is involved in non-INVITE server transaction [6].

C. SIP enabled OSA service capability server

OSA names the existing network functions which are accessible for 3rd party applications through API Service Capability Servers (SCS). The SCS which is accessible for multimedia session establishment in IP-based networks, supports SIP as a control protocol.

The SIP enabled OSA SCS supporting User interaction API incorporates a SIP entity handling SIP signaling and an OSA entity which is responsible for the dialog with the OSA application, as it is shown in Fig. 3. The mapping between SIP messages and methods of Generic User Interaction Service Interface is shown in Table 1. The OSA SCS supports an application view on the User Interaction object [1].

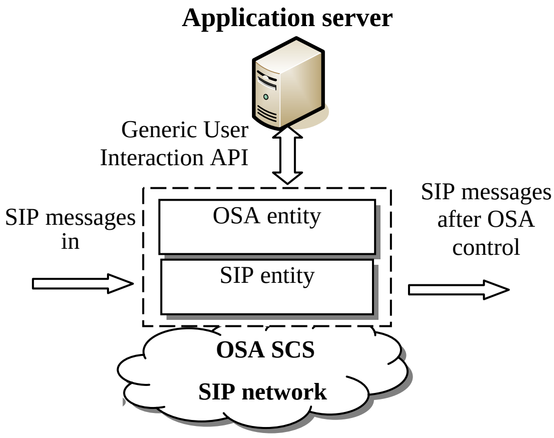

Fig. 3. OSA SCS supporting SIP signaling

Service point triggers [4] are those points in the SIP signaling on which filter criteria can be set. Filter criteria in the SIP entity triggers one or more service points triggers in order to send the related request to specific application server. The set of filter criteria is stored in service profile for given user. On reception a MESSAGE request, the SIP entity in the OSA SCS checks if the filter criteria are met and if so, forwards the request to the OSA entity in the OSA SCS. The OSA entity in the OSA SCS is responsible for the OSA dialog with the application.

III. Handling SIP user interaction sessions

The template of handling IP-media sessions defined in [4] can be applied to handling SIP user interaction sessions. The OSA SCS can utilize five basic modes of processing SIP MESSAGE requests. Applications can be built using combinations of these five modes of operation between the SG and OSA SCS. The OSA SCS can change from one mode of operation to another during the lifetime of user interaction the application is managing.

A. OSA SCS as user agent server or redirect server

In this mode, the application has preliminary subscribed for events concerning incoming short messages. When a mobile subscriber sends a short message to the application, the SIP entity in the SG acts as user agent client and creates a MESSAGE request which is sent towards the OSA SCS as shown in Fig. 4. The SIP entity in the OSA SCS acts as a user agent server or redirect server. The OSA entity in the OSA SCS invokes 'userInteractionEventNotify' method to notify the application about the incoming short message.

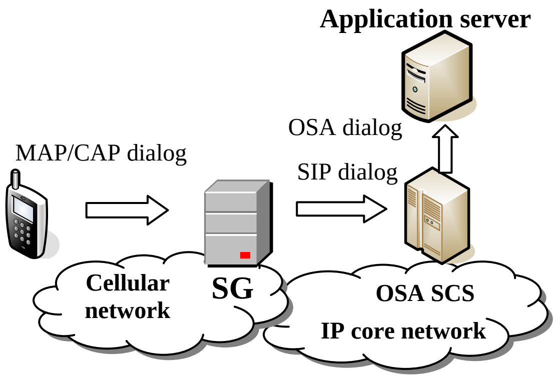

Fig. 4. OSA SCS acting as user agent server or redirect server

B. OSA SCS acting as user agent client

In this mode of operation, the application can send a short message to a mobile subscriber. It invokes “sendInfoReq” method of the OSA entity in OSA SCS. The SIP entity in OSA SCS creates MESSAGE request and sends it towards the SIP entity in the SG as shown in Fig. 5. The CAP/MAP entity in the SG translates the MESSAGE request into corresponding MAP/CAP signaling message and starts mobile terminated SMS transaction (see Table 1).

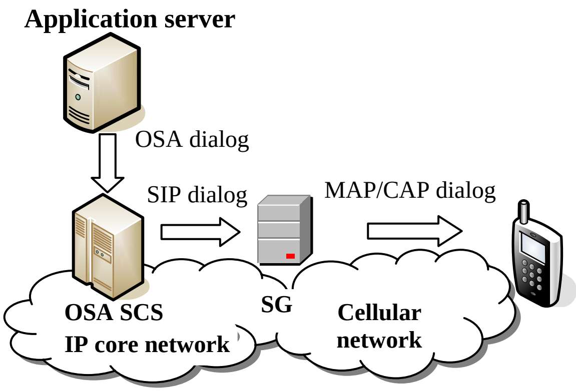

Fig. 5. OSA SCS acting as user agent client

C. OSA SCS acting as a SIP proxy

In this mode the application can be involved in forwarding short messages to another destination, adding or removing SMS destinations and gathering SMS statistics. As a precondition, the application needs to register its interest in events of particular short messages. When a short message of interest arrives in the SG, the SIP entity creates a MESSAGE request which is sent towards the OSA SCS. The SIP entity in the OSA SCS acts as a SIP proxy as shown in Fig. 6. The OSA entity in the OSA SCS and invokes ‘userInteractionEvent Notify’ method on the application. After application logic processing, the OSA entity in the OSA SCS transfers the control to the SIP entity in the OSA SCS, which in turn can add, remove or modify the header contents containing in the MESSAGE request according to changes requested by the application.

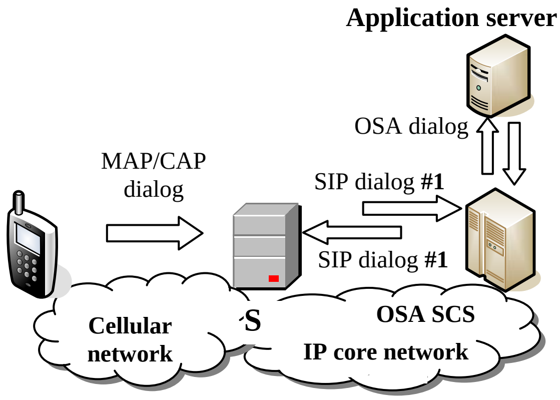

Fig. 6. OSA SCS acting as a SIP proxy

D. OSA SCS as a third party call control/ B2BUA Mode

If the application performs 3rd party control, the SIP entity in the OSA SCS acts as a B2BUA. There are several kinds of 3rd party user interaction control, for example:

- Routing B2BUA: the SIP entity in the OSA SCS receives a MESSAGE request from the SIP entity in the SG, terminates it and generates a new MESSAGE request, which is based on the instructions received from the application.

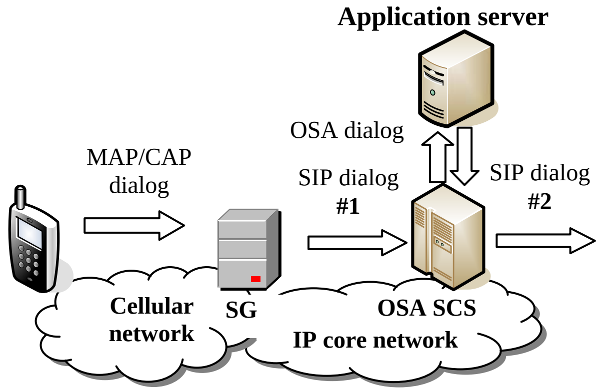

Fig. 7. OSA SCS performing 3rd party user interaction control acting as a routing B2BUA

In this mode of operation (Fig. 7) the incoming MESSAGE request is proxied by the SIP entity in the SG to the SIP entity in the OSA-SIP gateway, which then on behalf of the application, generates a new MESSAGE request for a different SIP dialog. The MESSAGE request is sent to the SIP proxy in the SG which then proxies it towards the destination. In this mode the application might be involved in barring of mobile originating and/or mobile terminating short messages.

- Initiating B2BUA: the SIP entity in the OSA SCS initiates two requests, which are logically connected together at the application side. This is the scenario in which the application sends two correlated short messages to different destinations.

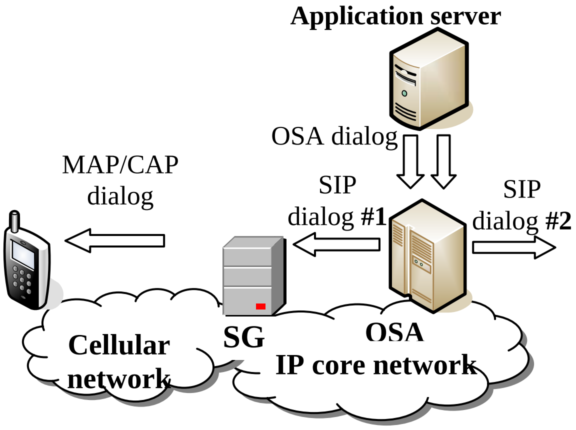

Fig. 8. OSA SCS performing 3rd party user interaction control acting as a initiating B2BUA

In this mode of operation (Fig. 8) the SIP entity in the OSA SCS initiates two MESSAGE requests with different SIP dialogs. The application is responsible for correlating the two dialogs. These requests are proxied through the SIP entity in the SG which then proxies them towards the destinations. In this mode the SIP entity in the OSA SCS behaves as a B2BUA for multiple SIP dialogs.

E. OSA SCS not involved or no longer involved

In this mode the application was either never involved in the SIP session signaling or has determined to be no longer involved. The incoming MESSAGE request is proxied by the SIP entity in the SG towards the destination as shown in Fig. 9.

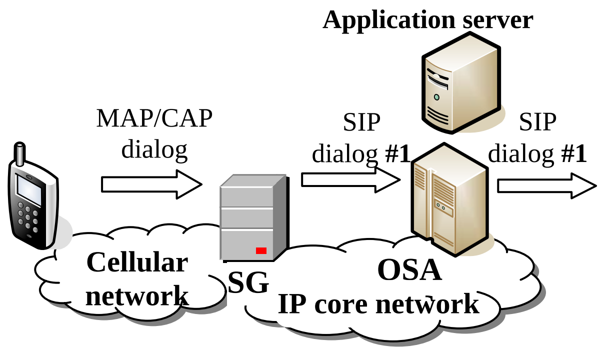

Fig. 9. A MESSAGE request is passed through the OSA SCS without application server involvement

IV. Use case

Let us consider an example of application which sends greeting messages to mobile subscribers on special occasion. Fig. 10 shows the corresponding message flow. First the application requests creation of a new user interaction object and then invokes “sendInfoReq” method to send a message. The OSA SCS creates a MESSAGE request and forwards it to the SG. The SG then translates the SIP request into appropriate MAP message which starts in turn MAP mobile terminating SMS transaction for sending the short message to the mobile subscriber. When the short message is delivered, the MSC servicing the mobile subscriber confirms message delivery. In the SG the confirmation message is translated into “200 OK” response which is transferred to the OSA SCS. On receiving “200 OK”, the OSA SCS invokes “sendInfoRes” method which informs the application about the completion of the “sendInfoRes”.

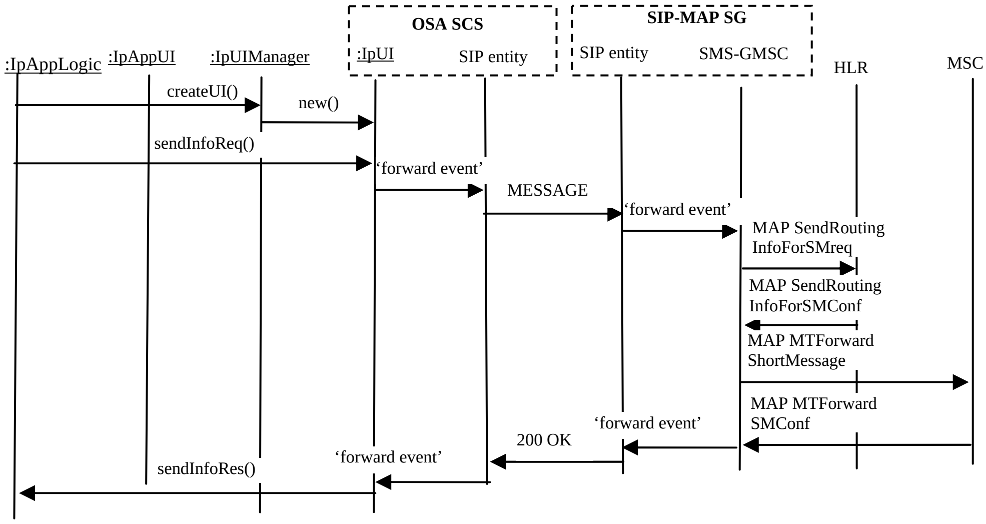

Fig. 10. Message sequence diagram for sending SMS from OSA application to mobile subscriber

Fig. 11 shows the mandatory parameters of method “sendInfoReq” and SIP request MESSAGE. In “sendInfoReq”, the parameter “info” specifies the greeting to send to the mobile subscriber, and a part of “variableInfo” contains the destination address (the MSISDN number). In MESSAGE, the destination address is in a form of telephone URL, and the greeting is in the MESSAGE body.

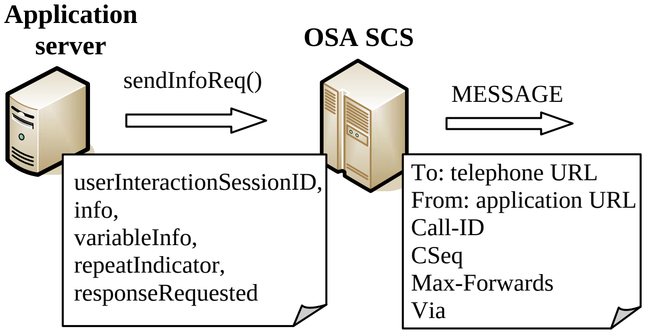

Fig. 11. OSA method sendInfoReq translated in SIP MESSAGE request

In the MAP “MTForwardShortMessage” message, the destination address is in the form of the mobile subscriber IMSI, while the greeting itself is encoded in “ShortMessage PDU” message.

V. Conclusion

The paper suggests a possible architecture for handling user interaction sessions over SIP network. A mapping between MAP/CAP signaling and SIP signaling in the context of non-call related user interaction is suggested. The OSA User Interaction API methods are mapped onto SIP messages. Different modes of operation between SIP entities in the MAP/CAP-SIP signaling gateway and OSA SCS are investigated.

The OSA User Interaction interface provides also call-related functions to send information to or gather information from the user involved in call. An announcement could be played to the subscriber involved in two-party call by setting up a SIP call session to an IP host representing a media server.

The biggest driver for SIP on the Internet is not only call control but also its extensions for provisioning functions that are call-unrelated. SIP allows integration of all types of communications and in combination with Open Service Access may be used to provide customized applications tailored to user preferences.

References

1. 3GPP TS 29.198-5 “Open Service Access (OSA); Application Programming Interface (API); Part 5: User Interaction Service Capability Feature (SCF)”. V. **6.3.1**, 2004.
2. 3GPP TS 29.998-05-04 “Open Service Access (OSA); Application Programming Interface (API) Mapping for OSA; Part 5: User Interaction Service Mapping; Subpart 4: API to SMS Mapping”. V. **7.0.0**, 2007.
3. 3GPP TS 23.078 “Customized Application for Mobile Network Enhanced Logic (CAMEL) Phase 4; Stage 2”. V. **7.9.0**, 2007.
4. 3GPP TR 29.198-04-05 “Open Service Access (OSA); Mapping for Open Service Access; Part 4: Call Control Service mapping; Subpart 4: Multiparty Call Control ISC. V. **7.0.0**, 2007.
5. Campbell, B., J. Rosenberg, H. Schulzrinne, C. Huitema, D. Gurle. RFC 3428 “Session Initiation Protocol (SIP) Extension for Instant Messaging”. 2002.
6. Rosenberg, J., H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. RFC 3261 “SIP: Session Initiation Protocol”. 2002.