

**Universal Mobile Telecommunications System (UMTS);
LTE;
Open Service Access (OSA) Application Programming
Interface (API) Mapping for Open Service Access;
Part 5: User Interaction Service Mapping;
Subpart 1: API to CAP Mapping
(3GPP TR 29.998-05-1 version 8.0.0 Release 8)**



Reference

RTR/TSGC-0029998-05-1v800

Keywords

LTE, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2009.
All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™**, **TIPHON™**, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

LTE™ is a Trade Mark of ETSI currently being registered

for the benefit of its Members and of the 3GPP Organizational Partners.

GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

Contents

Intellectual Property Rights	2
Foreword.....	2
Foreword.....	4
Introduction	4
1 Scope	5
2 References	5
3 Definitions and abbreviations.....	6
3.1 Definitions	6
3.2 Abbreviations	6
4 Generic Message Transfer Service CAMEL Call Flows	6
4.1 User Interaction	6
4.1.1 createUI	6
4.1.2 createUICall.....	7
4.1.3 enableUINotification	7
4.1.4 disableUINotification.....	8
4.1.5 userInteractionEventNotify.....	8
4.1.6 userInteractionAborted	9
4.1.7 userInteractionNotificationInterrupted.....	10
4.1.8 userInteractionNotificationContinued.....	10
4.1.9 userInteractionFaultDetected	11
4.1.10 sendInfoReq	12
4.1.11 sendInfoRes	14
4.1.12 sendInfoErr	16
4.1.13 sendInfoAndCollectReq.....	17
4.1.14 sendInfoAndCollectRes	20
4.1.15 sendInfoAndCollectErr	21
4.1.16 release	22
4.1.17 abortActionReq.....	23
4.1.18 abortActionRes	23
4.1.19 abortActionErr	24
Annex A: Change history	25
History	26

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

Structure of the OSA API Mapping (3GPP TR 29.998)

The present document is part 5 subpart 1 of a multi-part deliverable covering the Open Service Access (OSA); Application Programming Interface (API) Mapping for OSA.

Table: Overview of the OSA APIs & Protocol Mappings 29.198 & 29.998-family

OSA API specifications 29.198-family						OSA API Mapping - 29.998-family	
29.198-01	Overview					29.998-01	Overview
29.198-02	Common Data Definitions					29.998-02	<i>Not Applicable</i>
29.198-03	Framework					29.998-03	<i>Not Applicable</i>
Call Control (CC) SCF	29.198-04-1 Common CC data definitions	29.198-04-2 Generic CC SCF	29.198-04-3 Multi-Party CC SCF	29.198-04-4 Multi-media CC SCF	29.198-04-5 Conf. CC SCF	29.998-04-1	Generic Call Control – CAP mapping
						29.998-04-2	Generic Call Control – INAP mapping
						29.998-04-3	Generic Call Control – Megaco mapping
						29.998-04-4	Multiparty Call Control – ISC mapping
29.198-05	User Interaction SCF					29.998-05-1	User Interaction – CAP mapping
						29.998-05-2	User Interaction – INAP mapping
						29.998-05-3	User Interaction – Megaco mapping
						29.998-05-4	User Interaction – SMS mapping
29.198-06	Mobility SCF					29.998-06-1	User Status and User Location – MAP mapping
						29.998-06-2	User Status and User Location – SIP mapping
29.198-07	Terminal Capabilities SCF					29.998-07	<i>Not Applicable</i>
29.198-08	Data Session Control SCF					29.998-08	Data Session Control – CAP mapping
29.198-09	<i>Generic Messaging SCF</i>					29.998-09	<i>Not Applicable</i>
29.198-10	Connectivity Manager SCF					29.998-10	<i>Not Applicable</i>
29.198-11	Account Management SCF					29.998-11	<i>Not Applicable</i>
29.198-12	Charging SCF					29.998-12	<i>Not Applicable</i>
29.198-13	Policy Management SCF					29.998-13	<i>Not Applicable</i>
29.198-14	Presence & Availability Management SCF					29.998-14	<i>Not Applicable</i>
29.198-15	Multi Media Messaging SCF					29.998-15	<i>Not Applicable</i>
29.198-16	Service Broker SCF					29.998-16	<i>Not Applicable</i>

1 Scope

The present document investigates how the OSA User Interaction Interface Class methods defined in 3GPP TS 29.198-5 [5] can be mapped onto CAMEL Application Part operations and Mobile Application Part operations.

The mapping of the OSA API to the CAP and relevant MAP operations is considered informative, and not normative. An overview of the mapping TR is contained in the introduction of the present document as well as in 3GPP TR 29.998-1 [10].

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardised interface, i.e. the OSA APIs. The API specification is contained in the 3GPP TS 29.198 series of specifications. An overview of these is available in the introduction of the present document as well as in 3GPP TS 29.198-1 [1]. The concepts and the functional architecture for the Open Service Access (OSA) are described by 3GPP TS 23.198 [3]. The requirements for OSA are defined in 3GPP TS 22.127 [2].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TS 29.198-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview".
- [2] 3GPP TS 22.127: "Service Requirement for the Open Services Access (OSA); Stage 1".
- [3] 3GPP TS 23.198: "Open Service Access (OSA); Stage 2".
- [4] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
- [5] 3GPP TS 29.198-5: "Open Service Access (OSA); Application Programming Interface (API); Part 5: Generic user interaction".
- [6] 3GPP TS 29.002: "Mobile Application Part (MAP) specification".
- [7] 3GPP TS 29.078: "Customised Applications for Mobile network Enhanced Logic (CAMEL); CAMEL Application Part (CAP) specification".
- [8] 3GPP TS 22.101: "Service Aspects; Service Principles".
- [9] ITU-T Recommendation Q.850: "Usage of cause and location in the Digital Subscriber Signalling System No. 1 and the Signalling System No. 7 ISDN User Part".
- [10] 3GPP TR 29.998-1: "Open Service Access (OSA); Application Programming Interface (API) Mapping for OSA; Part 1: General Issues on API Mapping".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 29.198-1 [1] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 29.198-1 [1] apply.

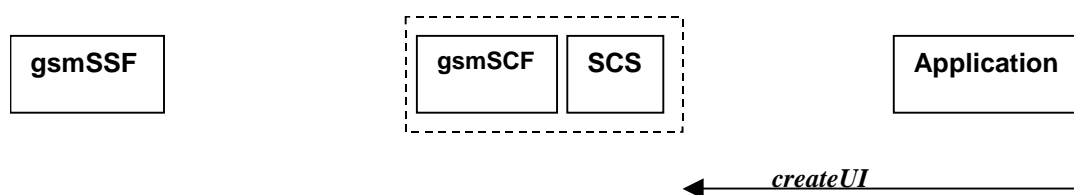
4 Generic Message Transfer Service CAMEL Call Flows

4.1 User Interaction

The User Interaction interface is used by applications to interact with end users. The API only supports Call User Interaction.

4.1.1 createUI

createUI is a method that is used to create a new (non call related) user interaction object.



NOTE: There are no associated CAP call flows.

Figure 4-1: Call Flow for createUI

Table 4-1: Normal Operation

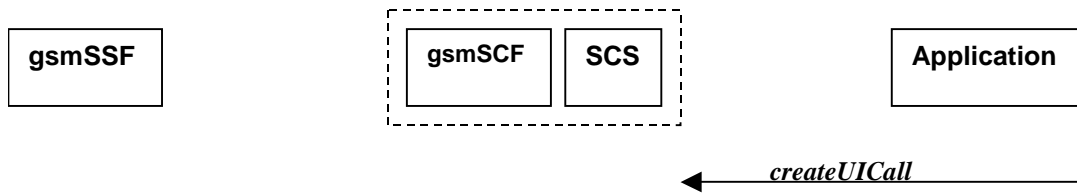
Pre-conditions	The application has been instructed to initiate a non call related User Interaction
1	The application invokes the <i>createUI</i> method
2	The SCS creates a new UI object

Parameter Mapping

None.

4.1.2 createUICall

createUICall is a method that is used to create a new call related user interaction object.



NOTE: There are no associated CAP call flows.

Figure 4-2: Call Flow for createUICall

Table 4-2: Normal Operation

Pre-conditions	The application has been requested to initiate a call related User Interaction
1	The application invokes the <i>createUICall</i> method
2	The SCS creates a new <i>UICall</i> object

Parameter Mapping

None.

4.1.3 enableUINotification

enableUINotification is a method that enables the reception of a user initiated user interaction.

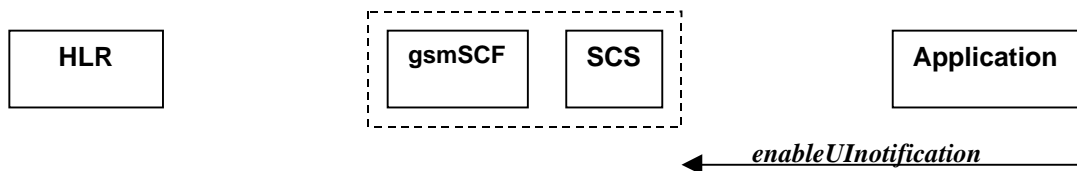


Figure 4-3: Call Flow for enableUINotification

Table 4-3: Normal Operation

Pre-conditions	An agreement is established between the network operator and the service provider for the event notification to be enabled
1	The application invokes the <i>enableUINotification</i> method
2	The SCS stores the request

Parameter Mapping

None.

4.1.4 disableUINotification

disableUINotification is a method that allows the application to remove notification for UI related actions previously set.

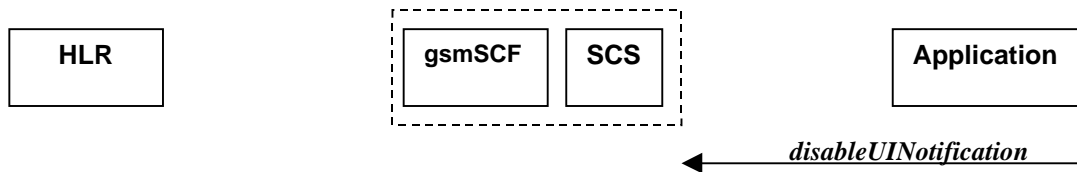


Figure 4-4: Call Flow for disableUINotification

Table 4-4: Normal Operation

Pre-conditions	An agreement is established between the network operator and the service provider for the event notification to be disabled
1	The application invokes the <i>disableUINotification</i> method
2	The request is disabled in the SCS.

Parameter Mapping

None.

4.1.5 userInteractionEventNotify

userInteractionEventNotify is a method that notifies the application of a user initiated request for user interaction.

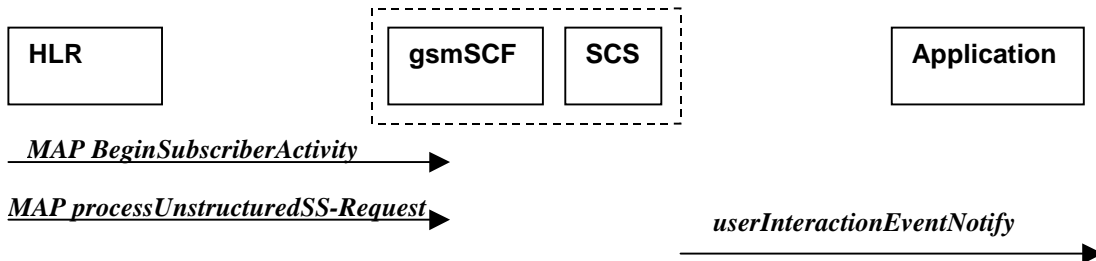


Figure 4-5: Call Flow for userInteractionEventNotify

Table 4-4: Normal Operation

Pre-conditions	
1	The gsmSCF receives a MAP <i>processUnstructuredSS-Request</i> message from the HLR. This operation may be preceded by MAP <i>beginSubscriberActivity</i> (see note)
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS identified the correct application that enable the notification request from the subscriber and invokes the <i>userInteractionEventNotify</i> method
NOTE:	The MAP <i>beginSubscriberActivity</i> is sent in case of MAP version 1.

Table 4-5: Parameter Mapping

From: processUnstructuredSS-Request	To: <i>userInteractionEventNotify</i>
	ui
	eventInfo (TpCallEventInfo):
msisdn	OriginatingAddress
	DestinationAddress
	ServiceCode
	DataTypeIndication
ussd-DataCodingScheme ussd-String	DataString
	assignmentID
	applInterface (output)

4.1.6 userInteractionAborted

userInteractionAborted is a method that indicates to the application that the User Interaction service instance has terminated or closed abnormally. No further communication will be possible between the User Interaction service instance and the application.

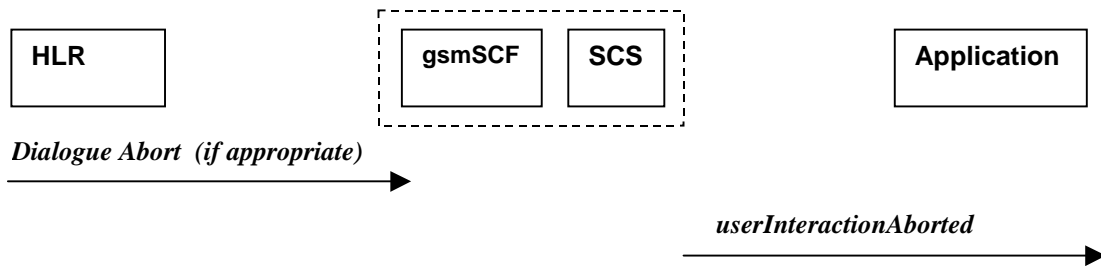


Figure 4-6: Call Flow for userInteractionAborted (scenario 1)

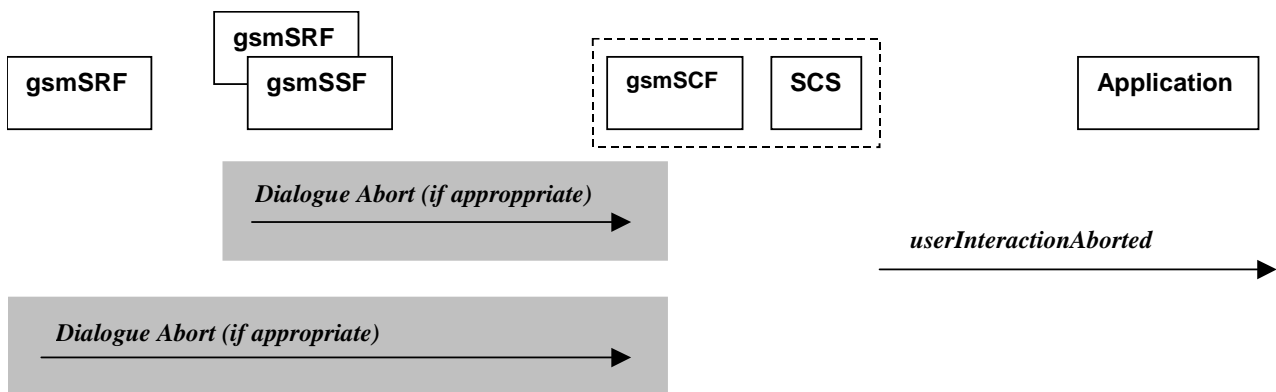


Figure 4-7: Call Flow for userInteractionAborted (scenario 2)

Parameter Mapping

None.

4.1.7 userInteractionNotificationInterrupted

userInteractionNotificationInterrupted is a method that indicates to the application that all user interaction event notifications have been temporarily interrupted.

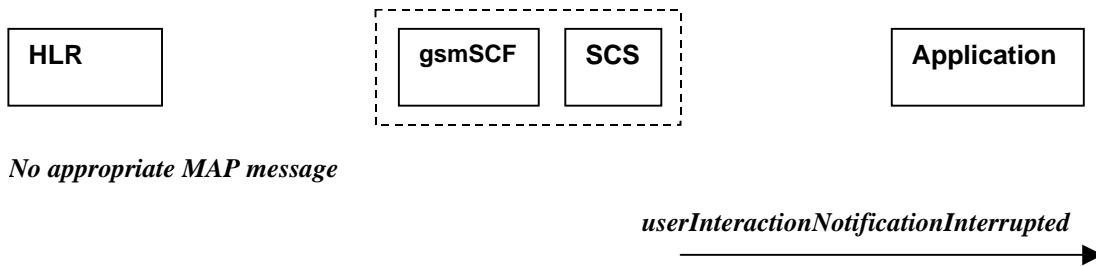


Figure 4-8: Call Flow for userInteractionNotificationInterrupted

Table 4-6: Normal Operation

Pre-conditions	User interaction event notifications have been enabled using the <i>enableUINotification</i> method on the <i>UIManager</i> interface
1	The SCS has detected, or has been informed of, a fault which prevents further user interaction events from being notified
2	The SCS invokes the <i>userInteractionNotificationInterrupted</i> method

Parameter Mapping

None.

4.1.8 userInteractionNotificationContinued

userInteractionNotificationContinued is a method that indicates to the application that user interaction event notifications will again be possible.

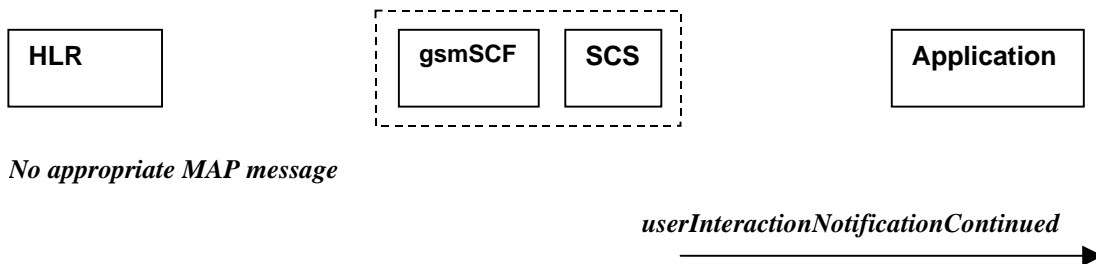


Figure 4-9: Call Flow for userInteractionNotificationContinued

Table 4-7: Normal Operation

Pre-conditions	User interaction event notifications have been interrupted and <i>userInteractionNotificationInterrupted</i> method has been invoked
1	The SCS detects that user interaction event notifications are again possible
2	The SCS invokes the <i>userInteractionNotificationContinued</i> method

Parameter Mapping

None.

4.1.9 userInteractionFaultDetected

userInteractionFaultDetected is a method that indicates to the application that a fault has been detected in the user interaction. This method is invoked e.g. if the call has been deassigned.

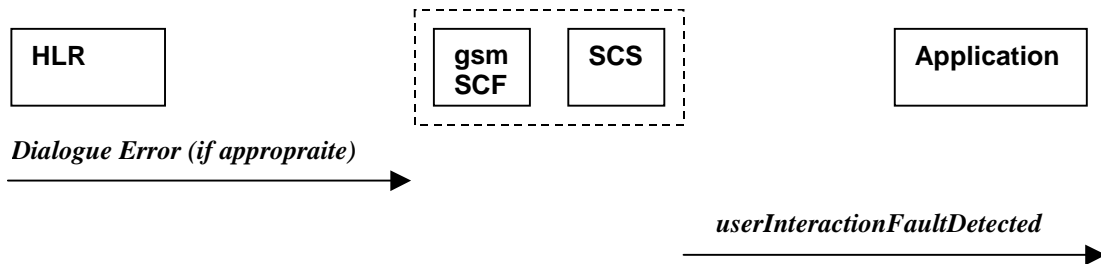


Figure 4-10: Call Flow for userInteractionFaultDetected (scenario 1)

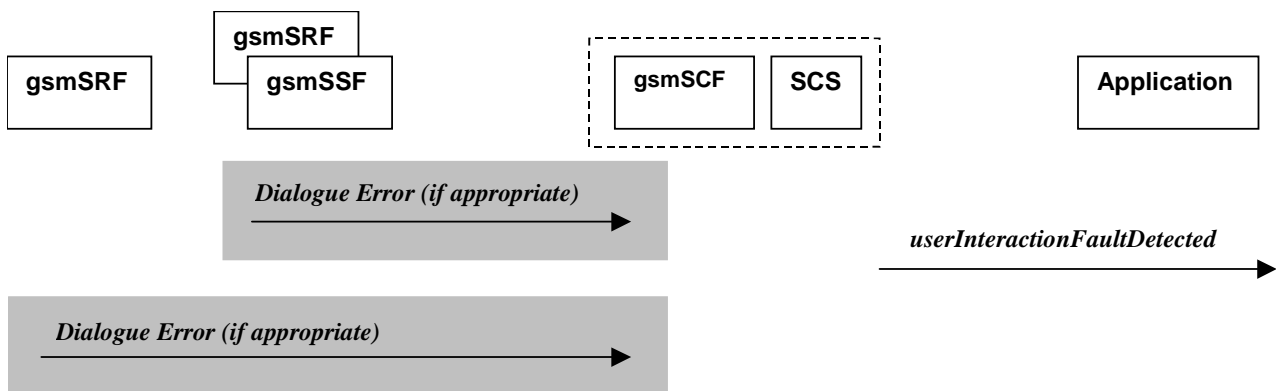


Figure 4-11: Call Flow for userInteractionFaultDetected (scenario 2)

Three Alternatives have been identified:

1. USSD based interaction between the MS and the gsmSCF (see table 4-8).
2. Interaction between a gsmSRF internal to the gsmSSF and the gsmSCF (see table 4-9).
3. Interaction between a gsmSRF and the gsmSCF (see table 4-10).

Table 4-8: Normal Operation

Pre-conditions	USSD interaction is in progress and a dialogue is running between the HLR and gsmSCF
1	The gsmSCF detects or receives an indication that there is an error in the user interaction
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS invokes the <i>userInteractionFaultDetected</i> method to the appropriate application

Table 4-9

Pre-conditions	User interaction is in progress between the gsmSRF and the gsmSCF
1	The gsmSCF detects or receives an indication that there is an error in the user interaction
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS invokes the <i>userInteractionFaultDetected</i> method to the appropriate application

Table 4-10

Pre-conditions	User interaction is in progress between the gsmSRF and the gsmSCF
1	The gsmSCF detects or receives an indication that there is an error in the user interaction
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS invokes the <i>userInteractionFaultDetected</i> method to the appropriate application

Table 4-11: Parameter Mapping

From: Dialogue Error	To: <i>userInteractionFaultDetected</i>
	<i>userInteractionIdentifier</i>
	<i>fault</i>
ReturnError	

4.1.10 sendInfoReq

sendInfoReq is an asynchronous method that sends information to the user.

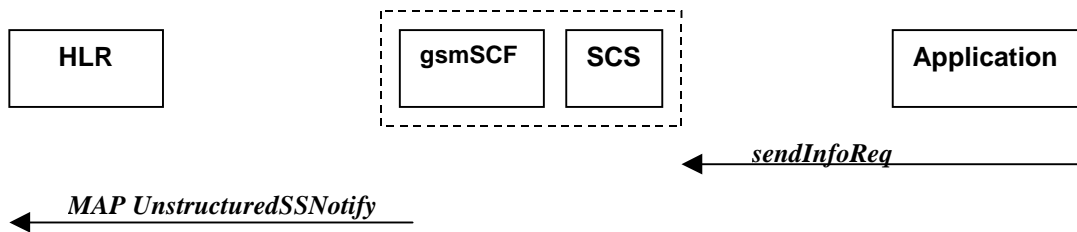


Figure 4-12: Call Flow for sendInfoReq (scenario 1)

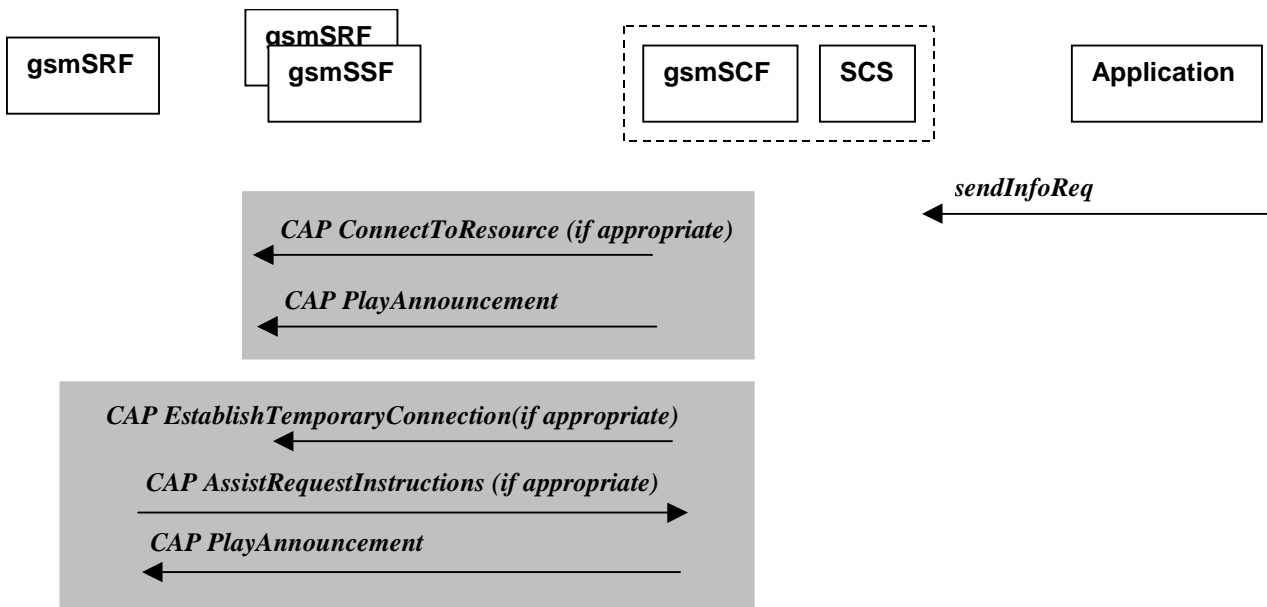


Figure 7-13: Call Flow for sendInfoReq (scenario 2)

Three Alternatives have been identified:

1. USSD based interaction between the MS and the gsmSCF (see table 4-12).
2. Interaction between a gsmSRF internal to the gsmSSF and the gsmSCF (see table 4-13).
3. Interaction between a gsmSRF internal to the gsmSSF and the gsmSCF (see table 4-14).

Table 4-12: Normal Operation

Pre-conditions	USSD interaction
1	The application invokes the sendInfo method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a MAP UnstructuredSS-Notify message to the HLR. If processUnstructuredSS-Request was previously received its result component may be sent containing ussd-DataCodingScheme and ussd-String (see note)
NOTE: For call-related USSD cases, the USSD is sent to the calling party.	

Table 4-13

Pre-conditions	
1	The application invokes the sendInfoReq method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF is aware of a gsmSRF internal to the gsmSSF. The gsmSCF sends CAP ConnectToResource , and CAP PlayAnnouncement messages the gsmSSF (see note)
NOTE: The user interaction shall apply to all parties connected to the call segment for the user interactions initiated by the connectToResource and establishTemporaryConnection operations.	

Table 4-14

Pre-conditions	
1	The application invokes the sendInfoReq method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF is aware of an external gsmSRF. The gsmSCF sends CAP EstablishTemporaryConnection message the gsmSSF
4	On receipt of the CAP AssistRequestInstructions message from the gsmSRF, the gsmSCF sends the CAP PlayAnnouncement message to the gsmSRF (see note)
NOTE: The user interaction shall apply to all parties connected to the call segment for the user interactions initiated by the connectToResource and establishTemporaryConnection operations.	

Table 4-15: Parameter Mapping

From: sendInfoReq	To: MAP unstructuredSS-Notify
userInteractionSessionID	
info (choice)	
infoID	
infoData	ussd-DataCodingScheme ussd-String
infoAddress	
variableInfoSet	
repeatIndicator	
responseRequested	
assignmentID	
	alertingPattern
	msisdn

Table 4-16

From: sendInfoReq	To: CAP PlayAnnouncement
userInteractionSessionID	
info (choice) infoID	InformationToSend (choice) inbandInfo messageID (choice) elementaryMessageID text messageContent attributes elementaryMessageIDs variableMessage elementaryMessageID variableParts (sequence of the following choices) integer number time date price numberOfRepetitions duration interval tone toneID duration
infoData	
infoAddress	
variableInfoSet	The contents are directly mapped to variableParts above
variablePartInteger	
variablePartAddress	
variablePartTime	
variablePartDate	
variablePartPrice	
repeatIndicator	This integer value is directly mapped to numberOfRepetitions above disconnectFromIPForbidden (according to responseRequested)
responseRequested	requestAnnouncementComplete
assignmentID	

4.1.11 sendInfoRes

sendInfoRes is an asynchronous method that informs the application about the start or the completion of a *sendInfoReq()*. This response is called only if the application has requested a response.



Figure 4-14: Call Flow for sendInfoRes (scenario 1)

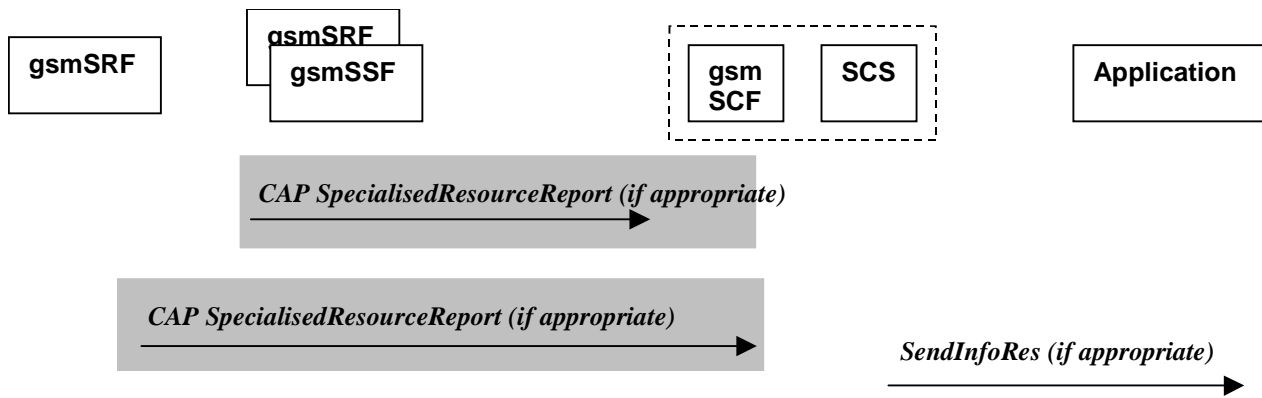


Figure 4-15: Call Flow for sendInfoRes (scenario 2)

Three Alternatives have been identified:

1. USSD based interaction between the MS and the gsmSCF (see table 4-17).
2. Interaction between a gsmSRF internal to the gsmSSF and the gsmSCF (see table 4-18).
3. Interaction between a gsmSRF internal to the gsmSSF and the gsmSCF (see table 4-19).

Table 4-17: Normal Operation

Pre-conditions	The application has previously invoked the <i>sendInfoReq</i> method and has requested a notification
1	The gsmSCF receives an MAP unstructured SS-Notify acknowledgement message from the HLR
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS identifies the correct application and invokes the sendInfoRes method

Table 4-18

Pre-conditions	The application has previously invoked the <i>sendInfoReq</i> method and has requested a notification
1	The gsmSCF receives a CAP SpecialisedResourceReport message from the gsmSSF indicating that the announcement has been played to the subscriber
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS identifies the correct application and invokes the sendInfoRes method

Table 4-19

Pre-conditions	The application has previously invoked the <i>sendInfoReq</i> method and has requested a notification
1	The gsmSCF receives a CAP SpecialisedResourceReport message from the gsmSRF indicating that the announcement has been played to the subscriber
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS identifies the correct application and invokes the sendInfoRes method

Table 4-20: Parameter Mapping

From: CAP SpecialisedResourceReport	To: sendInfoRes
	userInteractionSessionID
	assignmentID
	response

4.1.12 sendInfoErr

sendInfoErr is an asynchronous method that indicates that the request to send information was unsuccessful.

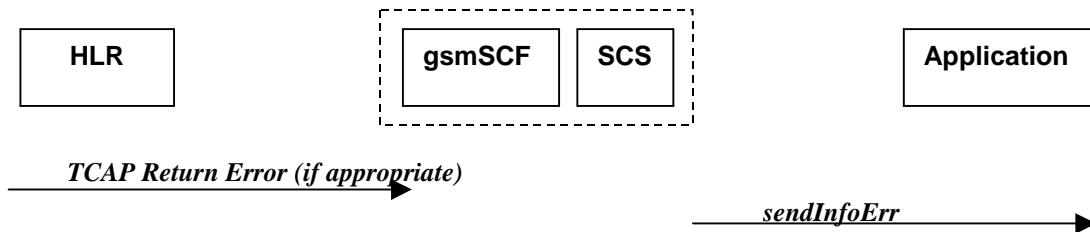


Figure 4-16: Call Flow for sendInfoErr (scenario 1)

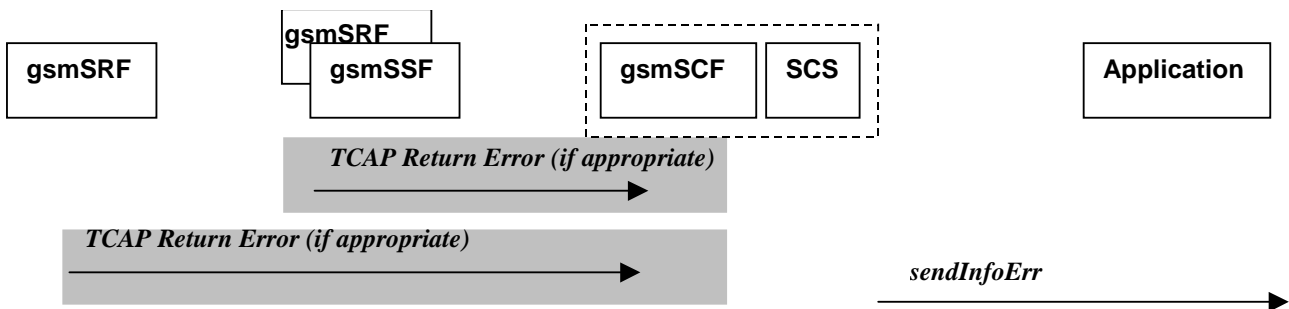


Figure 4-17: Call Flow for sendInfoErr (scenario 2)

Table 4-21: Normal Operation

Pre-conditions	The application has previously invoked the <i>sendInfoReq</i> method
1	The gsmSCF receives a message from the either the HLR, the gsmSSF or the gsmSRF indicating an error in the previous <i>sendInfoReq</i> method. Alternatively the gsmSCF may internal detect that the application has incorrectly sent the information
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS identifies the correct application and invokes the <i>sendInfoErr</i> method
For:	
1.	USSD based interaction between the MS and the CSE.
2.	Interaction between a gsmSRF internal to the gsmSSF and the CSE.
3.	Interaction between a gsmSRF internal to the gsmSSF and the CSE.

Table 4-22: Parameter Mapping

From: TCAP Return Error	To: sendInfoErr
	userInteractionSessionID
InvokeID	assignmentID
Error	error

4.1.13 sendInfoAndCollectReq

sendInfoAndCollectReq is an asynchronous method that plays an announcement or sends other information to the user and collects some information from the user. The announcement usually prompts for a number of characters (for example, these are digits or text strings such as "YES" if the user's terminal device is a phone).

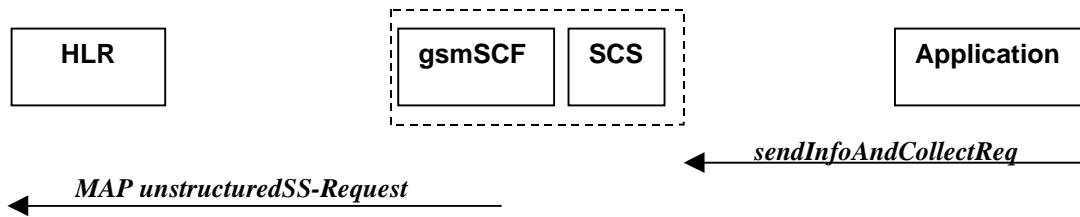


Figure 4-18: Call Flow for sendInfoAndCollectReq (scenario 1)

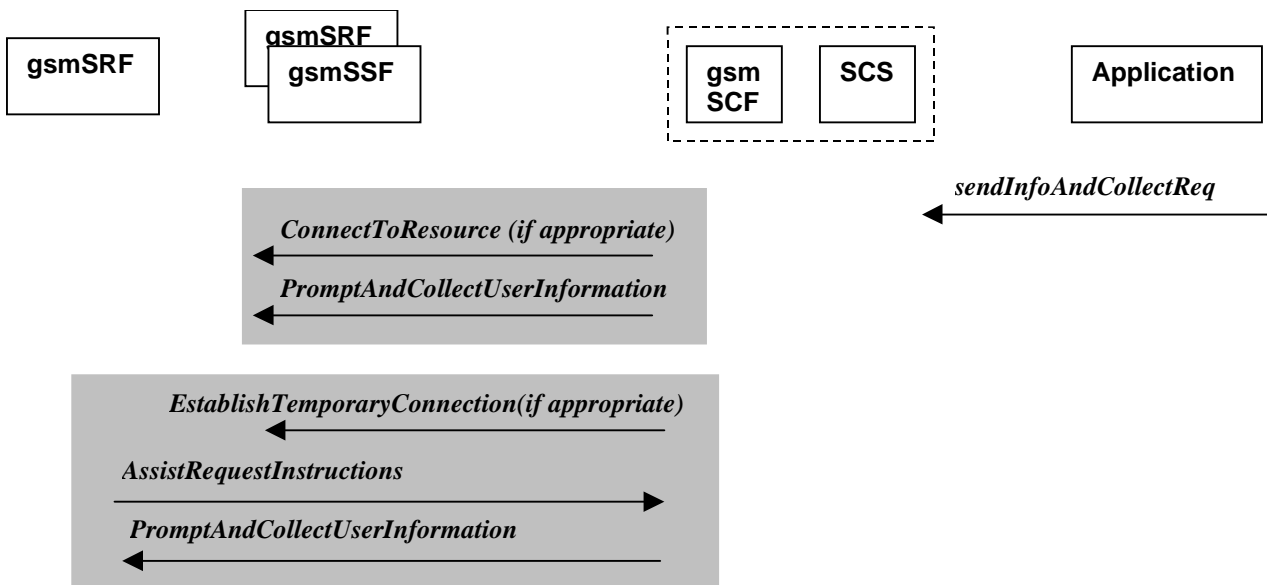


Figure 4-19: Call Flow for sendInfoAndCollectReq (scenario 2)

Three Alternatives have been identified:

1. USSD based interaction between the MS and the gsmSCF (see table 4-23).
2. Interaction between a gsmSRF internal to the gsmSSF and the gsmSCF (see table 4-24).
3. Interaction between a gsmSRF internal to the gsmSSF and the gsmSCF (see table 4-25).

Table 4-23: Normal Operation

Pre-conditions	USSD interaction
1	The application invokes the <i>sendInfoAndCollectReq</i> method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a MAP <i>unstructuredSS-Request</i> message

Table 4-24

Pre-conditions	
1	The application invokes the sendInfoAndCollectReq method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF is aware of a gsmSRF internal to the gsmSSF. The gsmSCF sends CAP ConnectToResource and PromptAndCollectUserInformation messages the gsmSSF

Table 4-25

Pre-conditions	
1	The application invokes the sendInfoAndCollectReq method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF is aware of an external gsmSRF. The gsmSCF sends CAP EstablishTemporaryConnection , message the gsmSSF
4	On receipt of the CAP AssistRequestInstructions message from the gsmSRF, the gsmSCF sends the CAP PromptAndCollectUserInformation message to the gsmSRF

Table 4-26: Parameter Mapping

From: sendInfoAndCollectReq	To: MAP unstructuredSS-Request
userInteractionSessionID	
info (choice)	
infoID	
infoData	ussd-DataCodingScheme ussd-String
infoAddress	
variableInfo	
criteria	
responseRequested	
	alertingPattern
	msisdn
assignmentID	

Table 4-27

From: sendInfoAndCollectReq	To: CAP PromptAndCollectUserInformation
userInteractionSessionID	
	disconnectFromIPForbidden (always true)
info (choice)	
infoID	InformationToSend (choice) inbandInfo messageID (choice) elementaryMessageID text messageContent attributes elementaryMessageIDs variableMessage elementaryMessageID variableParts (sequence of the following choices) integer number time date price numberOfRepetitions duration interval tone toneID duration
infoData	
infoAddress	
variableInfo	The contents are directly mapped to variableParts above
variablePartInteger	
variablePartAddress	
variablePartTime	
variablePartDate	
variablePartPrice	
criteria	collectedInfo
	collectedDigits
minLength	minimumNbOfDigits
maxLength	maximumNbOfDigits
endSequence	endOfReplyDigit
	cancelDigit
	startDigit
startTimeout	firstDigitTimeOut
interCharTimeout	interDigitTimeOut
	errorTreatment
	interruptableAnnInd
	voiceInformation
	voiceBack
responseRequested	
assignmentID	

4.1.14 sendInfoAndCollectRes

sendInfoAndCollectRes is an asynchronous method that returns the information collected to the application.

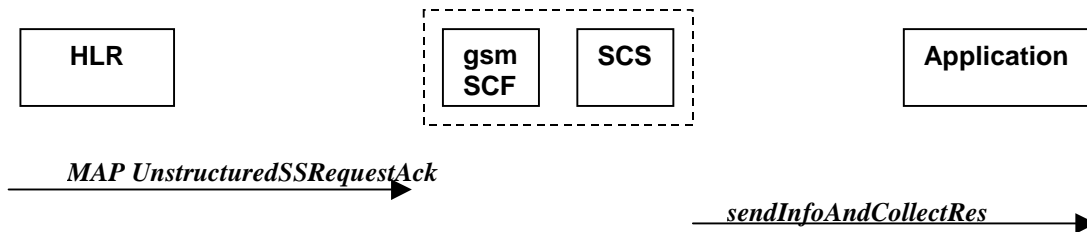


Figure 4-20: Call Flow for sendInfoAndCollectRes (scenario 1)

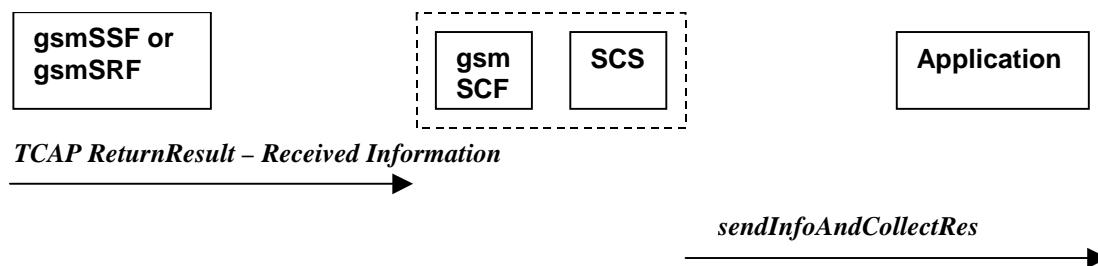


Figure 4-21: Call Flow for sendInfoAndCollectRes (scenario 2)

Two Alternatives have been identified:

1. USSD based interaction between the MS and the gsmSCF (see table 4-28).
2. Interaction with an gsmSRF internal to gsmSSF or external (see table 4-29).

Table 4-28: Normal Operation

Pre-conditions	The application has invoked a <i>sendInfoAndCollectReq()</i>
1	The gsmSCF receives a MAP UnstructuredSS-Request acknowledgement message from the HLR
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS invokes the sendInfoAndCollectRes method to the correct applications

Table 4-29

Pre-conditions	The application has invoked a <i>sendInfoAndCollectReq()</i>
1	The gsmSCF receives a TCAP ReturnResult from the gsmSSF or the gsmSRF depending on whether a direct or indirect gsmSRF is used containing the Received Information
2	The gsmSCF sends an equivalent internal operation to the SCS
3	The SCS identifies the correct application instance and invokes the sendInfoAndCollectRes method

Table 4-30: Parameter Mapping

From: MAP unstructuredSS-Request acknowledgement	To: sendInfoAndCollectRes
	userInteractionSessionID
	assignmentID
	response
ussd-DataCodingScheme ussd-String	info

Table 4-31

From: TCAP Return Result (Received Information)	To: sendInfoAndCollectRes
	userInteractionSessionID
	assignmentID
	response
DigitsResponse	info (only the digits are mapped)

4.1.15 sendInfoAndCollectErr

sendInfoAndCollectErr is an asynchronous method that indicates that the request to send information and collect a response was unsuccessful.

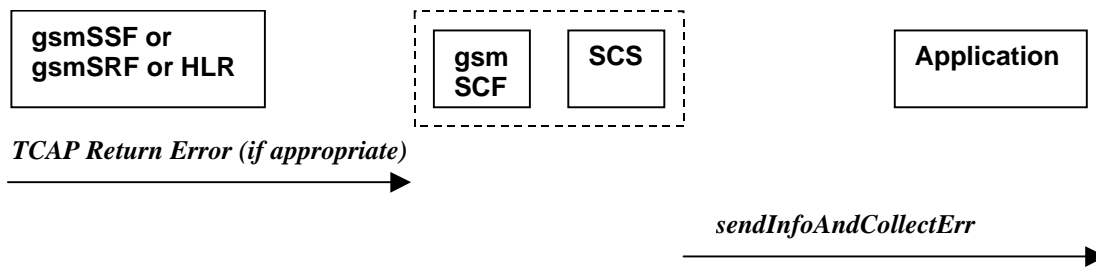


Figure 4-22: Call Flow for sendInfoAndCollectErr

Two Alternatives have been identified:

1. USSD based interaction between the MS and the gsmSCF (see table 4-32).
2. Interaction with an gsmSRF internal to gsmSSF or external gsmSRF (see table 4-33).

Table 4-32: Normal Operation

Pre-conditions	The application has invoked a <i>sendInfoAndCollectReq()</i>
1	The gsmSCF detects an error in the <i>sendInfoAndCollectReq</i> method or receives a message from the HLR indicating an error that there is an error in <i>sendInfoAndCollectReq</i> method
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS invokes the <i>sendInfoAndCollectErr</i> method to the correct application

Table 4-33

Pre-conditions	The application has invoked a <i>sendInfoAndCollectReq()</i>
1	The gsmSCF either detects an error or receives a TCAP Error from the gsmSSF or the gsmSRF depending on whether a direct or indirect gsmSRF is used
2	The gsmSCF sends an equivalent internal operation to the SCS
3	The SCS identifies the correct application instance and invokes the <i>sendInfoAndCollectErr</i> method

Table 4-34: Parameter Mapping

From: TCAP Return Error	To: sendInfoAndCollectErr
	userInteractionSessionID
	assignmentID
error	error

4.1.16 release

release is a method that requests that the relationship between the application and the user interaction object be released. It causes the release of the used user interaction resources and interrupts any ongoing user interaction.

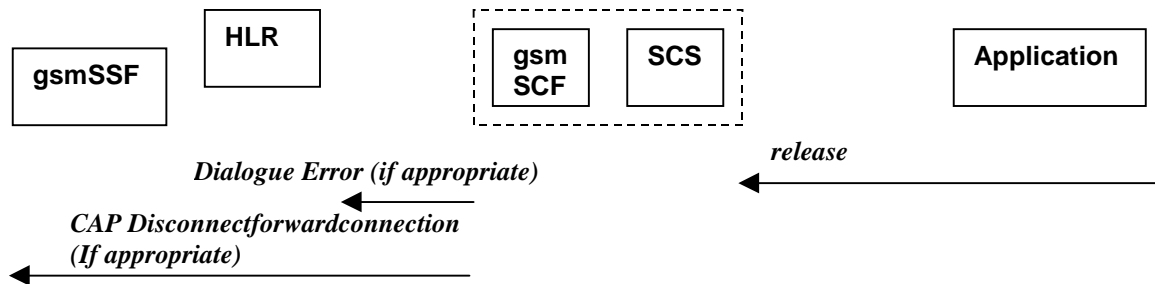


Figure 4-23: Call Flow for release

Two Alternatives have been identified:

1. USSD based interaction.
2. Interaction with a gsmSRF internal to gsmSSF or external gsmSRF (see table 4-36).

Table 4-35: Normal Operation

Pre-conditions	The gsmSCF has an open dialogue with the HLR
1	The application invokes a <i>release</i>
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a TCAP Abort message to the HLR if appropriate

Table 4-36

Pre-conditions	The application has previously invoked the <i>sendInfoAndCollectErr</i> . The gsmSCF is waiting for a response form the user
1	The application invokes a <i>release</i>
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP DisconnectForwardConnection to the gsmSSF

Table 4-37: Parameter Mapping

From: release	To: Dialogue Error
userInteractionSessionID	
	TC-U-ABORT TC-P-ABORT

Table 4-38:

From: release	To: CAP DisconnectForwardConnection
userInteractionSessionID	

4.1.17 abortActionReq

abortActionReq is an asynchronous method that aborts a user interaction operation, e.g. a *sendInfoReq*, from the specified call. The call remains otherwise unaffected. The user interaction call service interrupts the current action on the specified call.

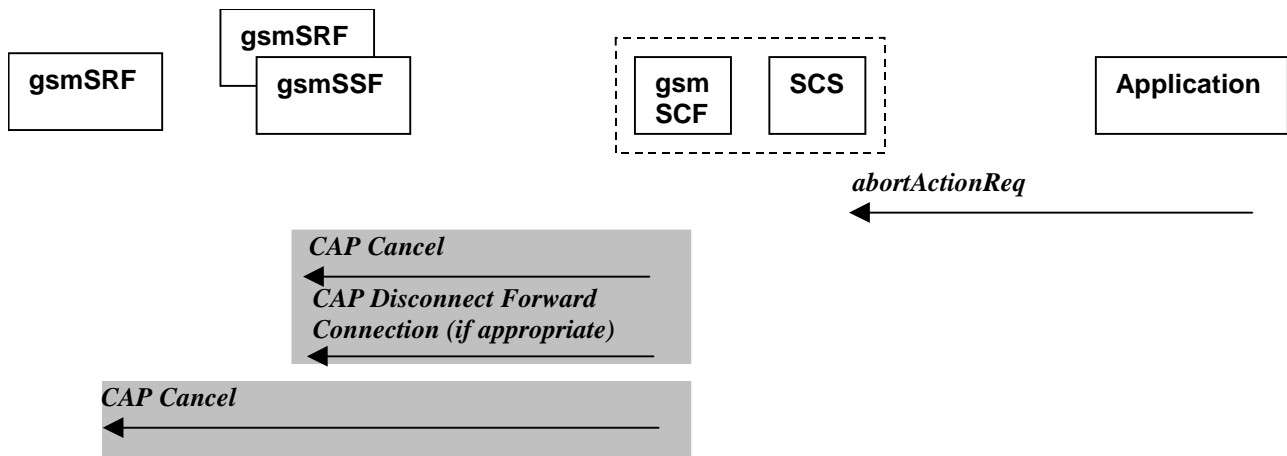


Figure 4-24: Call Flow for abortActionReq

Table 4-39: Normal Operation

Pre-conditions	The application has previously invoked e.g. the <i>sendInfoAndCollectReq</i> . The gsmSCF is waiting for a response from the user
1	The application invokes a <i>abortActionReq</i>
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP Cancel message to the gsmSSF or the gsmSRF as appropriate and may send a CAP DisconnectForwardConnection to the gsmSSF if appropriate

Table 4-40: Parameter Mapping

From: abortActionReq	To: CAP Cancel
userInteractionSessionID	
assignmentID	InvokeID
	allRequests

4.1.18 abortActionRes

abortActionRes is an asynchronous method that confirms that the request to abort a user interaction operation on a call was successful.

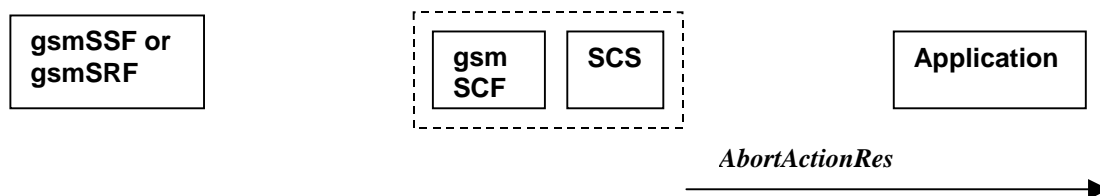


Figure 4-25: Call Flow for abortActionRes

There is no equivalent CAP/MAP mapping message.

Table 4-41: Normal Operation

Pre-conditions	The application has previously invoked the <i>abortActionReq</i> . The <i>gsmSCF</i> has sent the necessary instruction to the <i>gsmSSF</i> or the <i>gsmSRF</i> and is running a timer awaiting for any possible error return message. This timer expires and no errors are returned
2	The <i>gsmSCF</i> determines that the CAP Cancel operation was successful. The <i>gsmSCF</i> sends an equivalent internal message to the <i>SCS</i>
3	The <i>SCS</i> invokes the <i>abortActionRes</i> method to the appropriate application.

4.1.19 abortActionErr

abortActionErr is an asynchronous method that indicates that the request to abort a user interaction on a call resulted in an error.

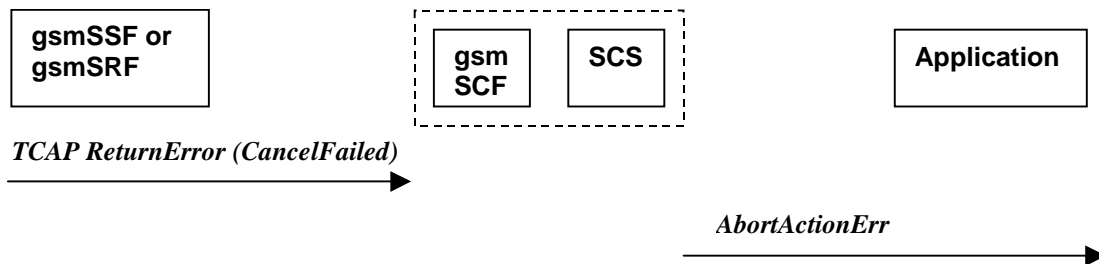


Figure 4-26: Call Flow for abortActionErr

Table 4-42: Parameter Mapping

From: TCAP error primitive	To: <i>abortActionErr</i>
	userInteractionSessionID
	assignmentID
TC-U-ERROR	error

Annex A: Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
Mar 2001	CN_11	NP-010131	011	--	CR 29.998: for moving TR 29.998 from R99 to Rel 4 (N5-010159)	3.2.0	4.0.0
Jun 2002	CN_16	--	--	--	Automatically upgraded to Rel-5 (i.e. no change/CR). The overview of the enlarged 29.198/29.998-family was updated in the Introduction.	4.0.0	5.0.0
Dec 2004	CN_26	--	--	--	Automatically upgraded to Rel-6 (i.e. no change/CR). The overview of the enlarged 29.198/29.998-family was updated in the Introduction.	5.0.0	6.0.0
Mar 2007	CT_35	--	--	--	Automatic upgrade to R7 (no CR needed)	6.0.0	7.0.0
Dec 2008	CT_42	--	--	--	Upgraded unchanged from Rel-7	7.0.0	8.0.0

History

Document history		
V8.0.0	January 2009	Publication