ETSI TR 129 998-4-1 V6.0.0 (2004-12)

Technical Report

Universal Mobile Telecommunications System (UMTS); Open Service Access (OSA) Application Programming Interface (API) Mapping for Open Service Access; Part 4: Call Control Service Mapping; Sub-part 1: API to CAP Mapping (3GPP TR 29.998-04-1 version 6.0.0 Release 6)

Reference

RTR/TSGN-0529998-04-1v600

Keywords

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/chaircor/ETSI_support.asp

Copyright Notification

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

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 4 subpart 1 of a multi-part deliverable covering the 3rd Generation Partnership Project; Technical Specification Group Core Network; 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*	*Not Applicable*
29.198-03	Framework	*29.998-03*	*Not Applicable*
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	**29.998-04-1**	**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 – SIP 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	User Status and User Location – MAP mapping
29.198-07	Terminal Capabilities SCF	*29.998-07*	*Not Applicable*
29.198-08	Data Session Control SCF	29.998-08	Data Session Control – CAP mapping
29.198-09	*Generic Messaging SCF*	*29.998-09*	*Not Applicable*
29.198-10	*Connectivity Manager SCF*	*29.998-10*	*Not Applicable*
29.198-11	Account Management SCF	*29.998-11*	*Not Applicable*
29.198-12	Charging SCF	*29.998-12*	*Not Applicable*
29.198-13	Policy Management SCF	*29.998-13*	*Not Applicable*
29.198-14	Presence & Availability Management SCF	*29.998-14*	*Not Applicable*
29.198-15	Multi-media Messaging SCF	*29.998-15*	*Not Applicable*

1 Scope

The present document investigates how the OSA Call Control Interface Class methods defined in 3GPP TS 29.198-4 [5] can be mapped onto CAMEL Application Part (CAP) operations and Mobile Application Part (MAP) 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

- 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 Service 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 29.198-4: "Open Service Access (OSA); Application Programming Interface (API); Part 4: Call control".

[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 Call Control CAMEL Call Flows

4.1 Call Control Manager

The generic call manager interface class provides the management functions to the generic call SCFs. The application programmer can use this interface to create call objects and to enable or disable call-related event notifications.

4.1.1 enableCallNotification

enableCallNotification is used to enable call notifications to be sent to the application.

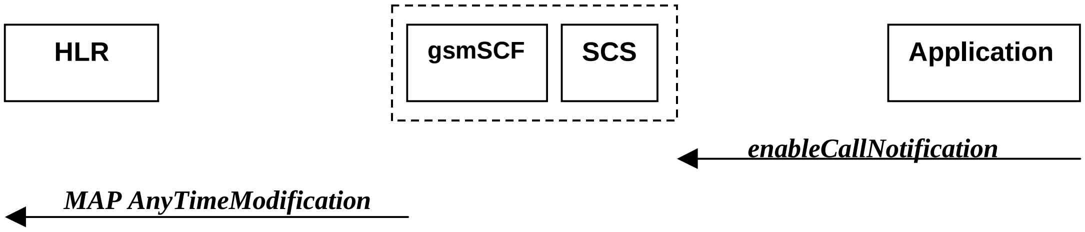

Figure 4-1: Call Flow for enableCallNotification

Two alternatives have been identified.

1. The application requests notifications to be enabled (see table 4-1).
2. HLR rejects CSI updates (see table 4-2).

Table 4-1: 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 ***enableCallNotification*** method
2	The gsmSCF sends a MAP ***AnyTimeModification*** to the HLR in order to Activate the necessary CAMEL Subscription Information (O-CSI, D-CSI, T-CSI, VT-CSI). NOTE: CAMEL phase 3 only allows for activation/deactivation of the CSI and not modification of the contents of the CSIs. The O-CSI and D-CSI will be activated if the originating address is present and the T-CSI and VT-CSI will be activated if the destination address is present.

Table 4-2: Error condition

Pre-conditions	gsmSCF had previously sent a MAP *AnyTimeModification* message to the HLR as a result of an *enableCallNotification* request from the application
1	HLR rejects the request to update the CSI
2	The gsmSCF sends an internal message to the SCS to indicate the up date failure
3	The SCS invokes the exception on ***enableCallNotification***

Table 4-3: Parameter Mapping

From: enableCallNotification	To: MAP AnyTimeModification
appInterface	
eventCriteria (TpCallEventCriteria) :	
DestinationAddress	subscriberIdentity (see note) modificationRequestFor-CSI
OriginationAddress	subscriberIdentity (see Note) modificationRequestFor-CSI
CallEventName (TpCallEventName)	CAMEL Subscription Information: - T-CSI; - VT-CSI; - O-CSI; - D-CSI.
CallNotificationType	
assignmentID	
	modificationRequestFor-SS-Info
	gsmSCF address
NOTE: In case an address range is used, a separate MAP AnyTimeModificationRequest shall be sent for every address in the range.	

4.1.2 disableCallNotification

disableCallNotification is used by the application to disable call notifications.

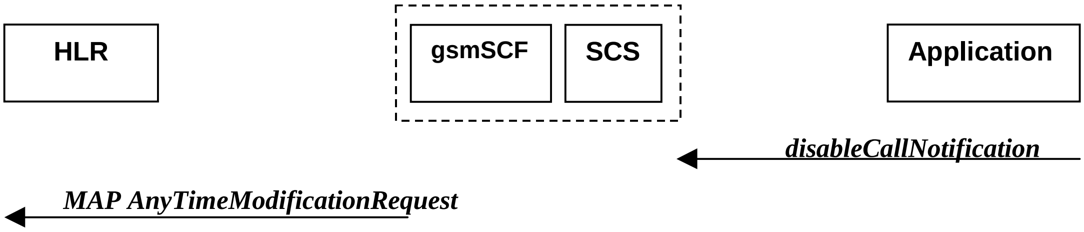

Figure 4-2: Call Flow for disableCallNotification

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 ***disableCallNotification*** method
2	The gsmSCF sends a MAP ***AnyTimeModification*** to the HLR in order to de-activate the CAMEL subscription Information (O-CSI, D-CSI, T-CSI, VT-CSI). NOTE: CAMEL Phase 3 only allows the capability to activate/deactivate CSI and not to modify the triggering information. The O-CSI and D-CSI will be deactivated if the originating address is present and the T-CSI and VT-CSI will be deactivated if the destination address is present.

Table 4-5: Parameter Mapping

From: disableCallNotification	To: MAP AnyTimeModification
assignmentID	
	gsmSCFAddress

4.1.3 changeCallNotification

changeCallNotification is used by the application to change the call notifications previously set by *enableCallNotification()*.

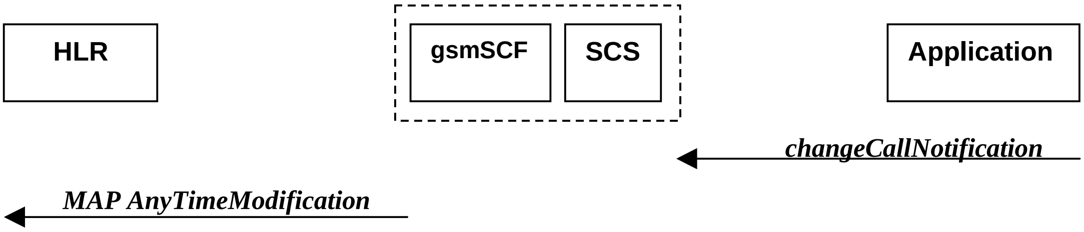

Figure 4-3: Call Flow for changeCallNotification

Table 4-6: Normal Operation

Pre-conditions	**Notifications have been enabled by the application.**
1	The application invokes the ***changeCallNotification*** method
2	The gsmSCF sends a MAP ***AnyTimeModification*** to the HLR in order to active and de-activate the CAMEL subscription Information (O-CSI, T-CSI, VT-CSI). The SCS and gsmSCF will have to determine which CSIs to active and which to de-activate in order to reflect the changed set of notifications. The O-CSI and D-CSI will be modified if the originating address is present and the T-CSI and VT-CSI will be modified if the destination address is present

Table 4-7: Parameter Mapping

From: changeCallNotification	**To: MAP AnyTimeModification**
assignmentID	
eventCriteria (TpCallEventCriteria):	
DestinationAddress	subscriberIdentity (see note) modificationRequestFor-CSI
OriginationAddress	subscriberIdentity (see note) modificationRequestFor-CSI
CallEventName (TpCallEventName)	CAMEL Subscription Information: - T-CSI; - VT-CSI; - O-CSI; - D-CSI.
CallNotificationType	
	modificationRequestFor-SS-Info
	gsmSCFAddress
NOTE: In case an address range is used, a separate MAP AnyTimeModificationRequest shall be sent for every address in the range	

4.1.4 getCriteria

getCriteria is used by the application to query the event criteria set with ***enableCallNotification***.

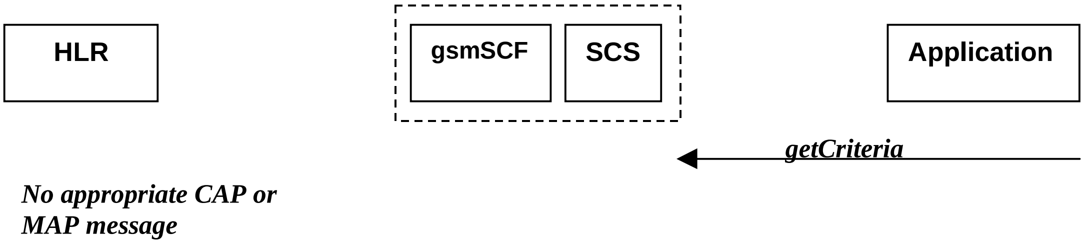

Figure 4-4: Call Flow for getCriteria

Table 4-8: Normal Operation

Pre-conditions	Notifications have been enabled by the application
1	The application invokes the ***getCriteria*** method
2	The SCS returns the criteria

Parameter Mapping

None.

4.1.5 setCallLoadControl

setCallLoadControl is a method used to control the number of invoked methods i.e. to restrict the load placed on the application server.

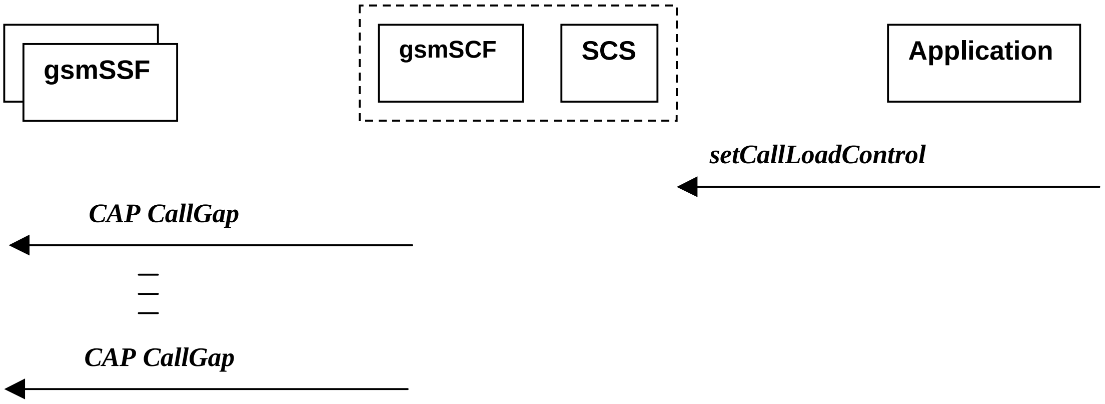

Figure 4-5: Call Flow for release

Table 4-9: Normal Operation

Pre-conditions	Call Control Manager is in active state
1	The application invokes the ***setCallLoadControl*** method
2	The SCS sends an equivalent message to the gsmSCF
3	The gsmSCF may invoke the CAP ***CallGap*** operations towards different gsmSSFs. CallGap can be sent in CAP only after the dialogue has been opened first by sending InitialDP.

Table 4-10: Parameter Mapping

From: setCallLoadControl	To: CAP CallGap
duration	gapIndicators duration
mechanism callLoadControlPerInterval	gapIndicators gapInterval
treatment ReleaseCause AdditionalTreatmentInfo InformationToSend	gapTreatment ReleaseCause InformationToSend
addressRange	gapCriteria basicGapCriteria calledAddressValue
assignmentID	

4.1.6 callNotificationInterrupted

callNotificationInterrupted indicates to the application that all event notifications have been interrupted, for example due to faults detected.

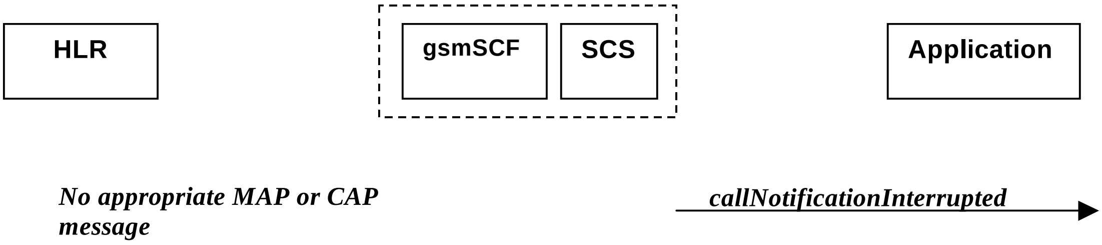
HLR
gsmSCF
SCS
Application
No appropriate MAP or CAP message
callNotificationInterrupted

Figure 4-6: Call Flow for callNotificationInterrupted

Table 4-11: Normal Operation

Pre-conditions	Call notifications have been enabled using the *enableNotification* method on the Call Manager interface
1	The SCS has detected, or has been informed of, a fault which prevents further events from being notified
2	The SCS invokes the ***callNotificationInterrupted*** method

Parameter Mapping

None.

4.1.7 callNotificationContinued

callNotificationContinued indicates to the application that all event notifications have been previously interrupted, have now started again.

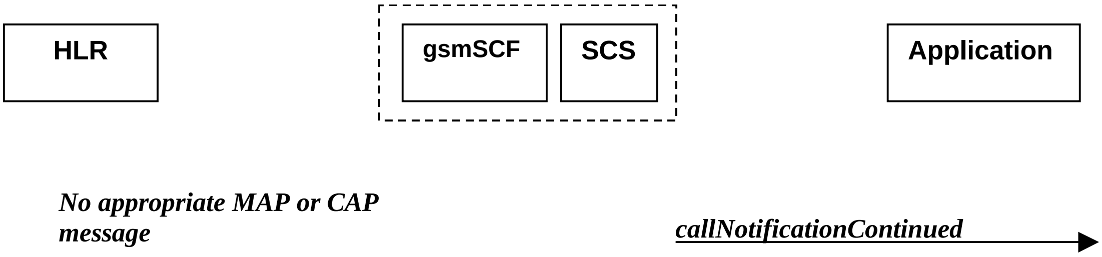

Figure 4-7: Call Flow for callNotificationContinued

Table 4-12: Normal Operation

Pre-conditions	Call notifications have been interrupted and *callNotificationInterrupted* method has been invoked
1	The SCS detects that call notifications are again possible.
2	The SCS invokes the ***callNotificationContinued*** method

Parameter Mapping

None.

4.1.8 callAborted

callAborted indicates to the application that the call object has aborted or terminated abnormally. No further communication will be possible between the call and the application.

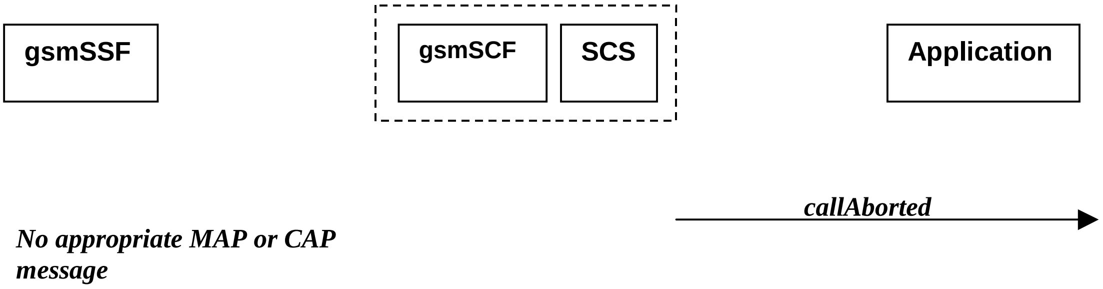

Figure 4-8: Call Flow for callAborted

Table 4-13: Normal Operation

Pre-conditions	
1	The SCS detect a catastrophic failure in its communication with the gsmSCF
2	The SCS, invokes the ***callAborted*** method. The call running in the network may continue and will not have been affected by this failure between the gsmSCF and the SCS

Parameter Mapping

None.

4.1.9 callEventNotify

callEventNotify notifies the application of the arrival of a call-related event.

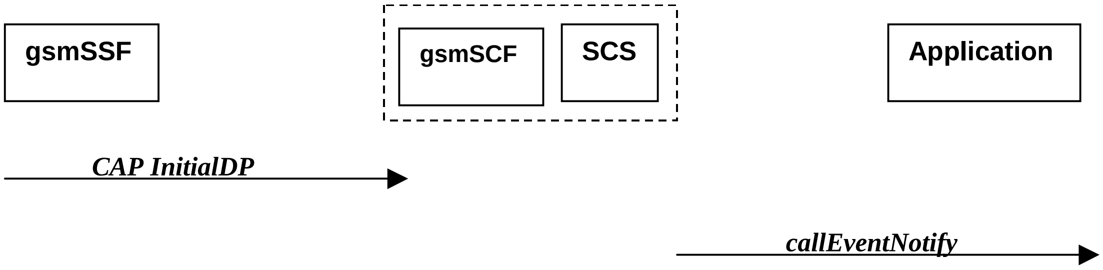

Figure 4-9: Call Flow for callEventNotify

Table 4-14: Normal Operation

Pre-conditions	Call notifications have been enabled using the *enableCallNotification* method on the Call Manager interface
1	A call arrives at the gsmSSF causing initial triggering to the gsmSCF CAP ***InitialDP***
2	The gsmSCF recognizes the need for an API service and passes the triggering information to the SCS
3	The SCS identifies the application responsible for handling the call and invokes the ***callEventNotify*** method

Table 4-15: Parameter Mapping

From: CAP InitialDP	To: *callEventNotify*
	callReference
	eventInfo (TpCallEventInfo) :
calledPartyNumber calledPartyBCDNumber BCD	destinationAddress
calling Party Number	originatingAddress
originalCalledPartyID	originalDestinationAddress
redirectingPartyID	redirectingAddress
	callAppInfo (TpCallAppInfoSet) :
	CallAppAlertingMechanism
	CallAppNetworkAccessType
ext-BasicServiceCode (1st priority)	CallAppBearerService CallAppTeleService
highLayerCompatibility (2nd priority)	CallAppTeleService
bearerCapability (2nd priority)	CallAppBearerService
callingPartysCategory	CallAppPartyCategory
	CallAppPresentationAddress
	CallAppGenericInfo
additionalCallingPartyNumber	CallAppAdditionalAddress
eventTypeBCSM	callEventName (see table 4-14)
	callNotificationType
	assignmentID
	appCall
serviceKey	(see note)
cGEncountered	
iPSSPCapabilities	
locationNumber	
redirectionInformation	
iMSI	
subscriberState	
locationInformation	
callReferenceNumber	
serviceInteractionIndicatorsTwo	
mscAddress	
timeAndTimezone	
gsm-ForwardingPending	
initialDPargExtension :	
naCarrierInformation	
gmscAddress	
cause	
cug-Index	
cug-Interlock	
cug-OutgoingAccess	
NOTE: Mapped to the method invocation.	

Table 4-16: eventTypeBCSM mapping to callEventName

From: CAP InitialDP parameter eventTypeBCSM	To: callEventNotify parameter callEventName in eventInfo
<no mapping available>	P_EVENT_NAME_UNDEFINED
<no mapping available>	P_EVENT_GCCS_OFFHOOK_EVENT
collectedInfo, termAttemptAuthorized	P_EVENT_GCCS_ADDRESS_COLLECTED_EVENT
analyzedInformation	P_EVENT_GCCS_ADDRESS_ANALYSED_EVENT
tBusy	P_EVENT_GCCS_CALLED_PARTY_BUSY
tBusy (see note)	P_EVENT_GCCS_CALLED_PARTY_UNREACHABLE
tNoAnswer	P_EVENT_GCCS_NO_ANSWER_FROM_CALLED_PARTY
routeSelectFailure	P_EVENT_GCCS_ROUTE_SELECT_FAILURE
<no mapping available>	P_EVENT_GCCS_ANSWER_FROM_CALL_PARTY
NOTE: Depending on the value of the *cause* parameter in the *initialDPArg extensions* parameter of the InitialDP operation.	

4.2 Call

The generic call interface represents the interface to the generic call SCF. It provides a structure to allow simple and complex call behaviour.

4.2.1 routeReq

routeReq is an asynchronous method which requests routing of the call (and inherently attached parties) to the destination party, via a passive call leg.

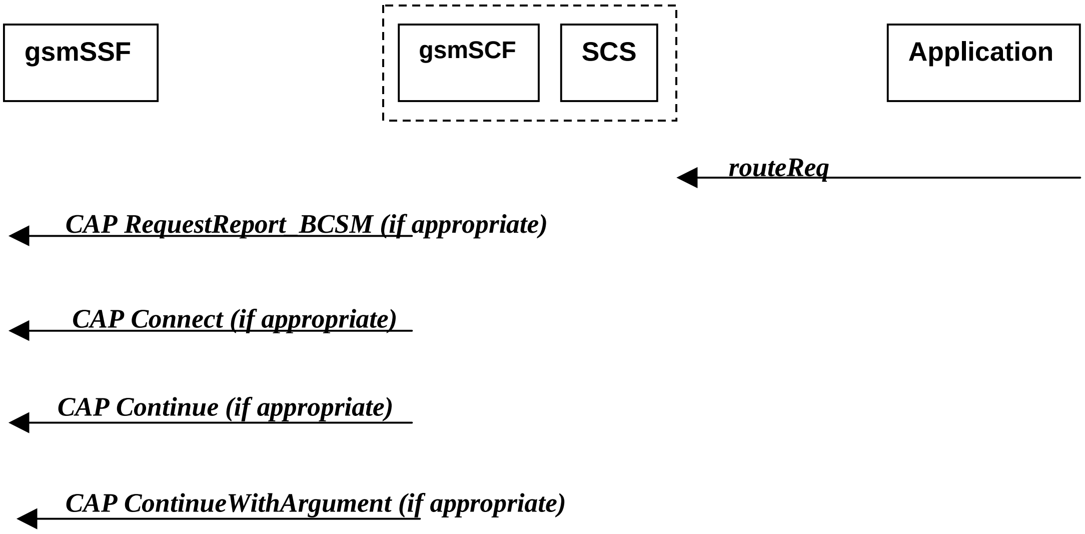
gsmSSF
gsmSCF
SCS
Application
routeReq
CAP RequestReport BCSM (if appropriate)
CAP Connect (if appropriate)
CAP Continue (if appropriate)
CAP ContinueWithArgument (if appropriate)

Figure 4-10: Call Flow for routeReq

Three alternatives have been identified:

1. The application changes the destination number (see table 4-17).
2. The application does not modify the destination address and does not provide any Application Information (see table 4-20).
3. The application does not modify the destination party number but modifies Application information (see table 4-23).

Table 4-17: Normal Operation

Pre-conditions	The application has been notified of a new call and the call object exists. The *setCallChargePlan* and *getCallInfoReq* methods may have been invoked
1	The application invokes the ***routeReq*** method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP ***RequestReportBSCM*** if the application needs to be informed about the outcome of the request
4	The gsmSCF sends a CAP ***Connect*** message

Table 4-18: Parameter Mapping

From: routeReq	To: CAP RequestReportBCSMEvent
callSessionID	
responseRequested (TpCallReportRequestSet) :	bcsmEvent :
MonitorMode (TpCallMonitorMode)	**monitorMode**
CallReportType (TpCallReportType)	**eventTypeBCSM**
AdditionalReportCriteria (TpCallReportAdditionalCriteria):	dPSpecificCriteria :
noAnswerDuration	applicationTimer
serviceCode	
	legID (see note)
targetAddress	
originatingAddress	
originalDestinationAddress	
redirectingAddress	
appInfo	
callLegSessionID	
NOTE: The legID for both the originating and the terminating leg are required for the disconnect event.	

Table 4-19:

From: routeReq	To: CAP Connect
callSessionID	
responseRequested	
targetAddress	destinationRoutingAddress
originatingAddress	
originalDestinationAddress	originalCalledPartyID
redirectingAddress	redirectingPartyID
appInfo (TpCallAppInfoSet) :	
CallAppAlertingMechanism	alertingPattern
CallAppNetworkAccessType	
CallAppTeleService	
CallAppBearerService	
CallAppPartyCategory	callingPartysCategory
PresentationAddress	genericNumbers (see note)
CallAppGenericInfo	
CallAppAdditionalAddress	genericNumbers
callLegSessionID	
	serviceInteractionIndicatorsTwo
	redirectionInformation
	suppressionOfAnnouncement
	oCSIApplicable
	na-Info :
	naCarrierInformation
	naOliInfo
	naChargeNumber
	connectArgExtension :
	cug-Interlock
	cug-OutgoingAccess
	nonCug-Call
NOTE: Operator specific function if CallAppAdditionalAddress is not used to map the genericNumbers parameter.	

Table 4-20

Pre-conditions	The application has been notified of a new call and the call object exists. The *setCallChargePlan* and *getCallInfoReq* methods may have been invoked
1	The application invokes the ***routeReq*** method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP ***RequestReportBSCM*** if the application needs to be informed about the outcome of the request
4	The gsmSCF sends a CAP ***Continue*** message

Table 4-21: Parameter Mapping

From: routeReq	To: CAP RequestReportBCSMEvent
callSessionID	
responseRequested (TpCallReportRequestSet):	bcsmEvent:
MonitorMode (TpCallMonitorMode)	**monitorMode**
CallReportType (TpCallReportType)	**eventTypeBCSM**
AdditionalReportCriteria (TpCallReportAdditionalCriteria):	**dPSpecificCriteria** :
noAnswerDuration	applicationTimer
serviceCode	
	legID (see note)
targetAddress	
originatingAddress	
originalDestinationAddress	
redirectingAddress	
appInfo	
callLegSessionID	
NOTE: The legID for both the originating and the terminating leg are required for the disconnect event.	

Table 4-22

From: routeReq	To: CAP Continue
callSessionID	
responseRequested	
targetAddress	
originatingAddress	
originalDestinationAddress	
redirectingAddress	
appInfo	
callLegSessionID	

Table 4-23

Pre-conditions	**The application has been notified of a new call and the call object exists. The** *setCallChargePlan* **and** *getCallInfoReq* **methods may have been invoked**
1	The application invokes the ***routeReq*** method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP ***RequestReportBSCM*** if the application needs to be informed about the outcome of the request
4	The gsmSCF sends a CAP ***ContinueWithArgument*** message

Table 4-24: Parameter Mapping

From: routeReq	To: CAP RequestReportBCSMEvent
callSessionID	
responseRequested (TpCallReportRequestSet):	bcsmEvent :
MonitorMode (TpCallMonitorMode)	**monitorMode**
CallReportType (TpCallReportType)	**eventTypeBCSM**
AdditionalReportCriteria (TpCallReportAdditionalCriteria):	**dPSpecificCriteria:**
noAnswerDuration	applicationTimer
serviceCode	
	legID (see note)
targetAddress	
originatingAddress	
originalDestinationAddress	
redirectingAddress	
appInfo	
callLegSessionID	
NOTE: The legID for both the originating and the terminating leg are required for the disconnect event.	

Table 4-25

From: routeReq	To: CAP ContinueWithArgument
callSessionID	
responseRequested	
targetAddress	
originatingAddress	
originalDestinationAddress	
redirectingAddress	
appInfo:	
CallAppAlertingMechanism	alerting Pattern
CallAppNetworkAccessType	
CallAppTeleService	
CallAppBearerService	
CallAppPartyCategory	callingPartysCategory
PresentationAddress	genericNumbers (see note)
CallAppGenericInfo	
CallAppAdditionalAddress	genericNumbers
callLegSessionID	
	serviceInteractionIndicatorsTwo
	suppressionOfAnnouncement
	na-Info:
	naCarrierInformation
	naOliInfo
	naChargeNumber
	continueWithArgumentArgExtension:
	cug-Interlock
	cug-OutgoingAccess
	nonCug-Call
NOTE: Operator specific function if CallAppAdditionalAddress is not used to map the genericNumbers parameter.	

4.2.2 routeRes

routeRes is an asynchronous method which indicates that the request to route the call to the destination was successful, and indicates the response of the destination party (for example, the call was answered, not answered, refused due to busy, etc.). For every trigger that was armed in the parameter **responseRequested** of the ***routeReq*** a ***routeRes*** method may be invoked.

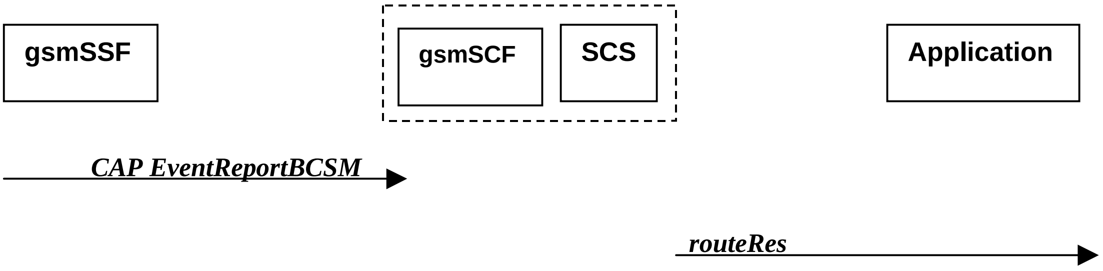

Figure 4-11: Call Flow for routeRes

Table 4-26: Normal Operation

Pre-conditions	Call routing attempted
1	If event reports have been requested, the gsmSSF sends a CAP ***EventReportBCSM*** to the gsmSCF
2	The gsmSCF sends an equivalent message to the SCS
3	The SCS invokes the ***routeRes*** method

Table 4-27: Parameter Mapping

From: CAP EventReportBCSM	To: routeRes
	callSessionID
	eventReport:
miscCallInfo	MonitorMode
	CallEventTime
eventTypeBCSM legID	CallReportType (TpCallReportType)
eventSpecificInformationBCSM	AdditionalReportInfo (TpCallAdditionalReportInfo)
	callLegSessionID

4.2.3 routeErr

routeErr is an asynchronous method which indicates that the request to route the call to the destination party was unsuccessful – the call could not be routed to the destination party (for example, the network was unable to route the call, parameters were incorrect, the request was refused, etc).

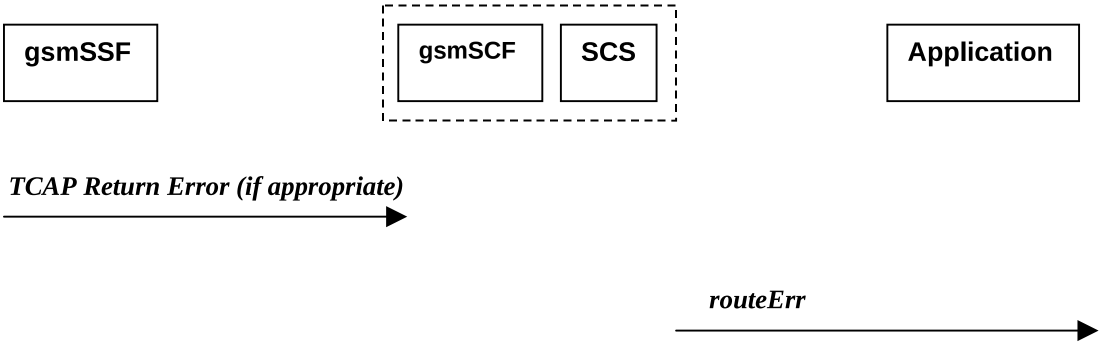

Figure 4-12: Call Flow for routeErr

Two scenarios are possible:

1. The gsmSCF receives a message from the gsmSSF indicating an error (see table 4-28).
2. The gsmSCF detects there is an error in the message from the SCS (see table 4-29).

Table 4-28: Normal Operation

Pre-conditions	Call routing attempted
1	The gsmSSF detects a call routing failure and sends an appropriate TCAP message returning an error to the gsmSCF
2	The gsmSCF sends an equivalent message to the SCS
3	The SCS detects an error with the ***routeReq*** method, or receives a TCAP Return Error, and invokes the ***routeErr*** method

Table 4-29

Pre-conditions	Call routing attempted
1	The gsmSCF detects an error in the parameters of the internal message from the SCS requesting a ***routeReq***
2	The gsmSCF sends an equivalent message to the SCS
3	The SCS invokes the ***routeErr*** method

Table 4-30: Parameter Mapping

From: TCAP Return Error	To: routeErr
	callSessionID
TC-U-ERROR TC-U-REJECT	error
	callLegSessionID

4.2.4 release

release is a method used to request the release of the call and associated objects.

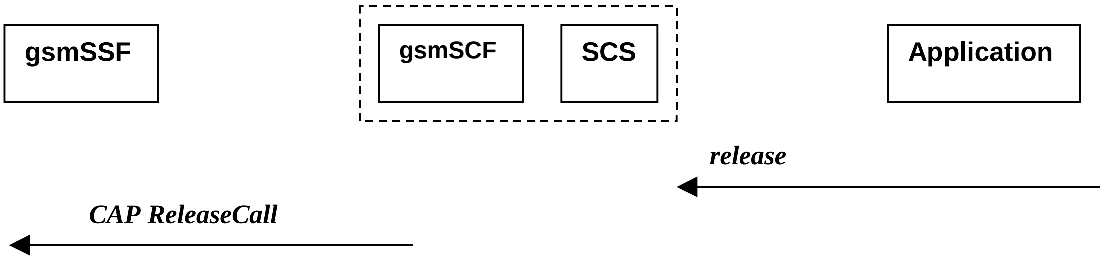

Figure 4-13: Call Flow for release

Table 4-31: Normal Operation

Pre-conditions	Call is in progress
1	The application invokes the ***release*** method
2	The SCS sends an equivalent message to the gsmSCF
3	The gsmSCF invokes the CAP ***ReleaseCall*** operation

Table 4-32: Parameter Mapping

From: release	To: CAP ReleaseCall
callSessionID	
cause (TpCallReleaseCause) :	
value (specified in ITU-T Recommendation Q.850)	Cause
location	

4.2.5 deassignCall

deassignCall is a method that requests that the relationship between the application and the call and associated objects be de-assigned. It leaves the call in progress, however, it purges the specified call object so that the application has no further control of call processing. If a call is de-assigned that has event reports or call information reports requested, then these reports will be disabled and any related information discarded.

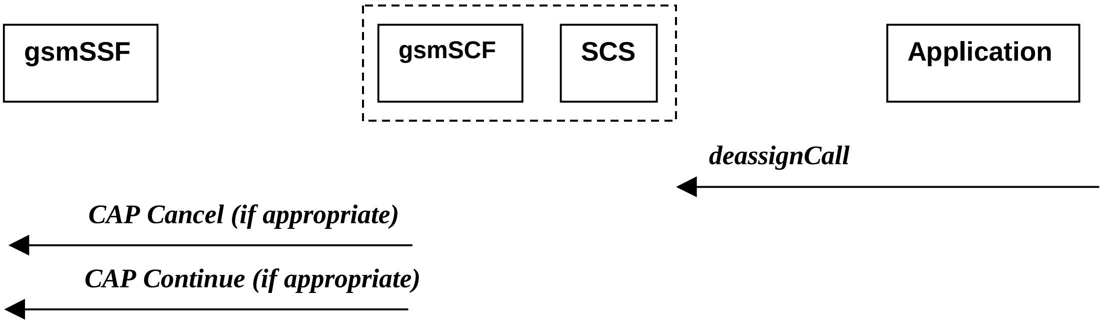

Figure 4-14: Call Flow for deassignCall

Table 4-33: Normal Operation

Pre-conditions	
1	The application invokes the ***deassignCall*** method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP ***Cancel*** operation to the gsmSSF if there are any reports pending.
4	The gsmSCF may send a CAP ***Continue*** to allow the interrupted call processing to continue. This is not sent if the call has already been established.

Table 4-34: Parameter Mapping

From: deassignCall	To: CAP Cancel
	AllRequests
callSessionID	

Table 4-35

From: deassignCall	To: CAP Continue
callSessionID	

4.2.6 getCallInfoReq

getCallInfoReq is an asynchronous method that requests information associated with the call to be provided at the appropriate time (for example, to calculate charging). This method must be invoked before the call is routed to a target address. The call object will exist after the call is ended if information is required to be sent to the application at the end of the call. The information will be sent after any call event report.

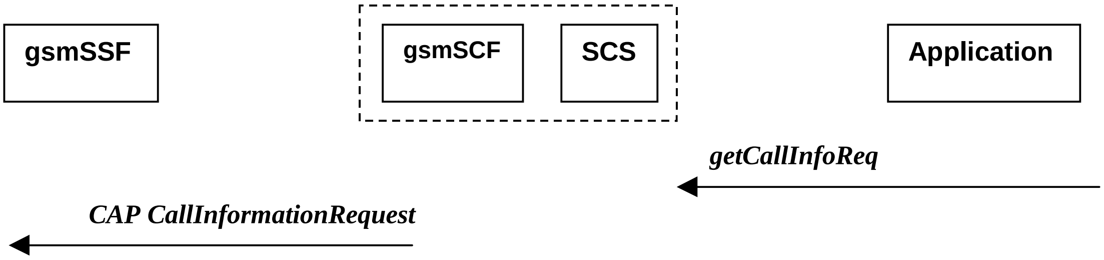

Figure 4-15: Call Flow for getCallInfoReq

Table 4-36: Normal Operation

Pre-conditions	
1	The application invokes the ***getCallInfoReq*** method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP ***CallInformationRequest*** operation to the gsmSSF

Table 4-37: Parameter Mapping

From: getCallInfoReq	To: CAP CallInformationRequest
callSessionID	
	RequestedInformationTypeList
callInfoRequested (TpCallInfoType):	RequestedInformationType
P_CALL_INFO_UNDEFINED	
P_CALL_INFO_TIMES	callAttemptElapsedTime callStopTime callConnectedElapsedTime
P_CALL_INFO_RELEASE_CAUSE	releaseCause
P_CALL_INFO_INTERMEDIATE	
	LegID

4.2.7 getCallInfoRes

getCallInfoRes is an asynchronous method that reports all the necessary information requested by the application, for example to calculate charging.

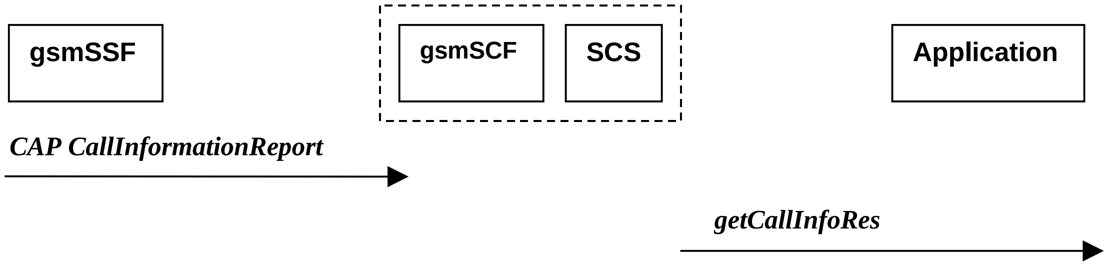

Figure 4-16: Call Flow for getCallInfoRes

Table 4-38: Normal Operation

Pre-conditions	Call is in progress
1	The gsmSCF receives a CAP ***CallInformationReport*** from the gsmSSF
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS identifies the correct application and invokes the ***getCallInfoRes*** method

Table 4-39: Parameter Mapping

From: CAP CallInformationReport	To: getCallInfoRes
	callSessionID
requestedInformationList	callInfoReport :
requestedInformationType :	CallInfoType
	P_CALL_INFO_UNDEFINED
callAttemptElapsedTime callStopTime callConnectedElapsedTime	P_CALL_INFO_TIMES
releaseCause	P_CALL_INFO_RELEASE_CAUSE
	P_CALL_INFO_INTERMEDIATE
requestedInformationValue :	
	CallInitiationStartTime
callStopTimeValue	CallEndTime
	CallConnectedToResourceTime
	CallConnectedToDestinationTime
releaseCauseValue	Cause
LegID	

4.2.8 getCallInfoErr

getCallInfoErr is an asynchronous method that reports that the original request was erroneous, or resulted in an error condition.

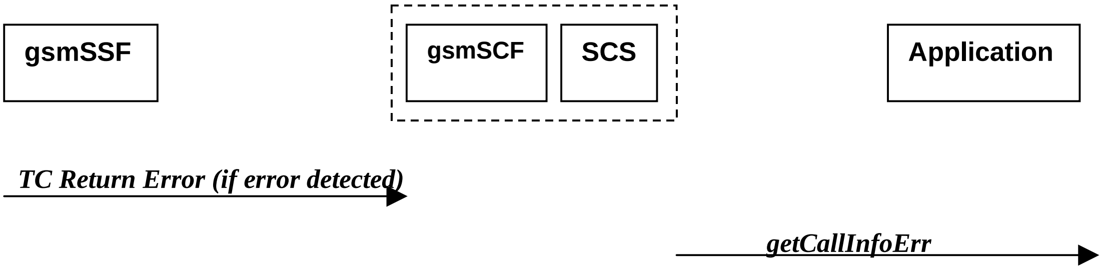

Figure 4-17: Call Flow for getCallInfoErr

Table 4-40: Normal Operation

Pre-conditions	The application has requested information associated with a call via the *getCallInfoReq* method
1	A call terminates abnormally and the gsmSSF sends an error in a TCAP message to the gsmSCF, or aborts the TCAP dialogue
2	The gsmSCF sends an equivalent message to the SCS
3	The SCS identifies the correct applications that requested the call information and invokes the ***getCallInfoErr*** method.

Table 4-41: Parameter Mapping

From:	**To:** *getCallInfoErr*
	callSessionID
TC Primitives TC-U-ABORT TC-P-ABORT TC-NOTICE TC-U-ERROR TC-L-CANCEL TC-U-CANCEL TC-L-REJECT TC-R-REJECT TC-U-REJECT	errorIndication

4.2.9 superviseCallReq

superviseCallReq is a method that is called by the application to supervise a call. The application can set a granted connection time for this call. If an application calls this method before it calls a ***routeReq()*** or a user interaction method the time measurement will start as soon as the call is answered by the B-party or the user interaction system.

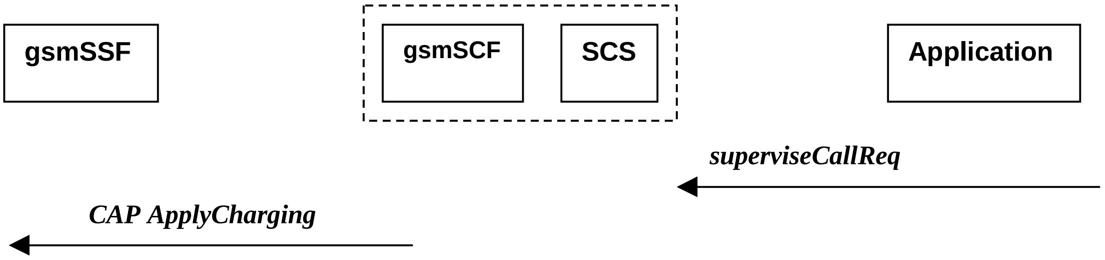

Figure 4-18: Call Flow for superviseCallReq

Table 4-42: Normal Operation

Pre-conditions	
1	The application invokes the ***superviseCallReq*** method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP ***ApplyCharging*** message to the gsmSSF

Table 4-43: Parameter Mapping

From: superviseCallReq	**To: CAP ApplyCharging**
callSessionID	
	AchBillingCharging Characteristics :
time	timeDurationCharging - maxCallPeriodDuration
	- tariffSwitchInterval
treatment (TpCallSuperviseTreatment) :	timeDurationCharging
P_CALL_SUPERVISE_RELEASE	- releaseIfdurationExceeded
P_CALL_SUPERVISE_RESPOND	
P_CALL_SUPERVISE_APPLY_TONE	- tone
	PartyToCharge

4.2.10 superviseCallRes

superviseCallRes is an asynchronous method that reports a call supervision event to the application.

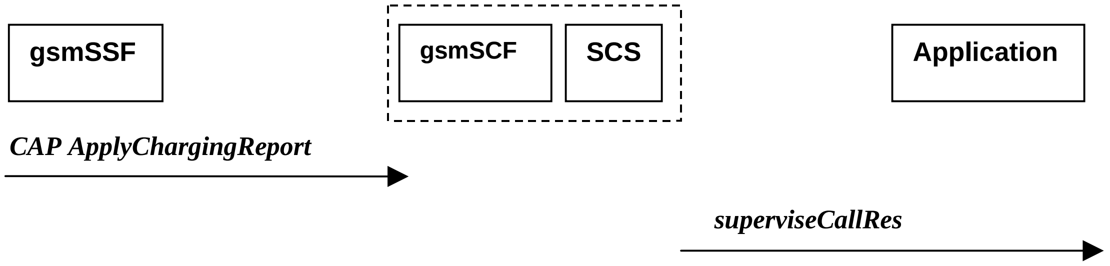

Figure 4-19: Call Flow for superviseCallRes

Table 4-44: Normal Operation

Pre-conditions	The application has invoked the supervise Call method
1	The gsmSCF receives an CAP ***ApplyChargingReport*** from the gsmSSF
2	The gsmSCF sends an equivalent internal message to the SCS
3	The SCS identifies the correct application and invokes the ***superviseCallRes*** method

Table 4-45: Parameter Mapping

From: CAP ApplyChargingReport	To: superviseCallRes
	callSessionID
CallResult	report (TpCallSuperviseReport) :
- CallReleasedAtTcpExpiry	- P_CALL_SUPERVISE_TIMEOUT
- CallActive	- P_CALL_SUPERVISE_CALL_ENDED
	- P_CALL_SUPERVISE_TONE_APPLIED
	- P_CALL_SUPERVISE_UI_FINISHED
CallResult - TimeInformation	usedTime
CallResult - PartyToCharge	

4.2.11 superviseCallErr

superviseCallErr is an asynchronous method that reports a call supervision error to the application.

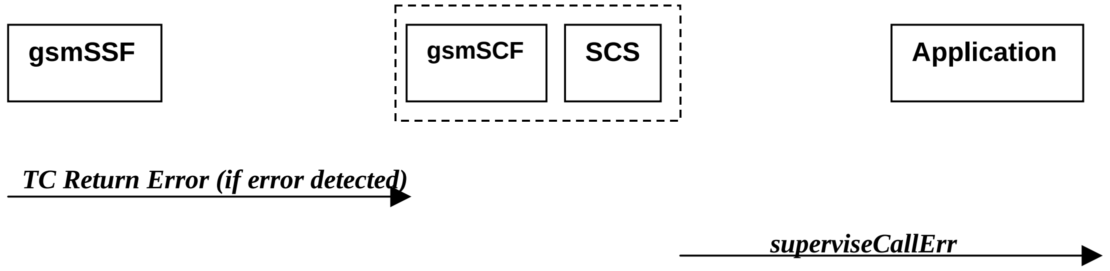

Figure 4-20: Call Flow for superviseCallErr

Table 4-46: Normal Operation

Pre-conditions	The application has requested information associated with a call via the *superviseCallReq* method
1	A call terminates abnormally and the gsmSSF sends an error in a TCAP message to the gsmSCF, or aborts the TCAP dialogue
2	The gsmSCF sends an equivalent message to the SCS
3	The SCS identifies the correct applications that requested the call information and invokes the ***superviseCallErr*** method

Table 4-47: Parameter Mapping

From:	**To:** *superviseCallErr*
	callSessionID
TC Primitives TC-U-ABORT TC-P-ABORT TC-NOTICE TC-U-ERROR TC-L-CANCEL TC-U-CANCEL TC-L-REJECT TC-R-REJECT TC-U-REJECT	**errorIndication**

4.2.12 setAdviceOfCharge

setAdviceOfCharge is a method that allows the application to determine the charging information that will be send to the end-users terminal.

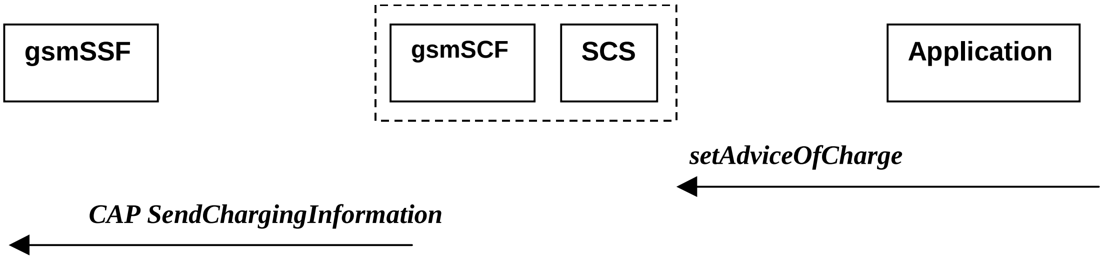

Figure 4-21: Call Flow for setAdviceOfCharge

Table 4-48: Normal Operation

Pre-conditions	
1	The application invokes the ***setAdviceOfCharge*** method
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP ***SendChargingInformation*** message to the SSF

Table 4-49: Parameter Mapping

From: setAdviceOfCharge	To: CAP SendChargingInformation
callSessionID	
aOCInfo: - CurrentCAI	SCIBillingChargingCharateristics aOCBeforeAnswer aOCInitial - or - SCIBillingChargingCharateristics aOCAfterAnswer cAI-GSM0224
- NextCAI	SCIBillingChargingCharateristics aOCBeforeAnswer aOCSubsequent cAI-GSM0224
tariffSwitch	SCIBillingChargingCharateristics aOCBeforeAnswer aOCSubsequent tariffSwitchInterval - or - SCIBillingChargingCharateristics aOCAfterAnswer tariffSwitchInterval
	partyToCharge

4.2.13 setCallChargePlan

setCallChargePlan is a method that allows the application to include charging information in network generated CDR.

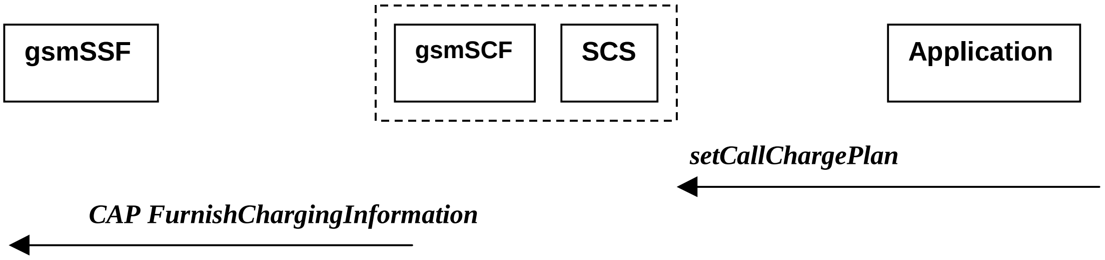

Figure 4-22: Call Flow for setCallChargePlan

Table 4-50: Normal Operation

Pre-conditions	
1	The application invokes the ***setCallChargePlan***
2	The SCS sends an equivalent internal message to the gsmSCF
3	The gsmSCF sends a CAP ***FurnishChargingInformation*** message to the SSP

Table 4-51: Parameter Mapping

From: setCallChargePlan	To: CAP FurnishChargingInformation
callSessionID	
callChargePlan ChargeOrderType TransparentCharge ChargePlan AdditionalInfo	FCIBillingChargingCharacteristics fCIBCCCAMELsequence1 freeFormatData
callChargePlan PartyToCharge	FCIBillingChargingCharacteristics fCIBCCCAMELsequence1 partyToCharge
	FCIBillingChargingCharacteristics fCIBCCCAMELsequence1 appendFreeFormatData

An alternative scenario would be to map ***setCallChargePlan*** method to the CAP ***ApplyCharging*** protocol operation.

4.2.14 callFaultDetected

callFaultDetected indicates to the application that a fault has been detected in the call.

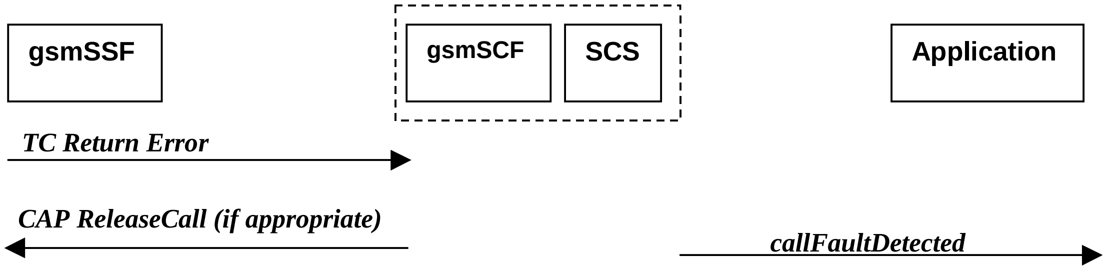

Figure 4-23: Call Flow for callFaultDetected

Table 4-52: Normal Operation

Pre-conditions	A call exists and the SCS detects an error. No *routeReq* method has been invoked yet
1	The gsmSSF may detect a fault and sends an appropriate dialogue error message to the gsmSCF
2	The gsmSCF may detect a fault an send an error message to the SCS
3	The SCS detects a fault and invokes the ***callFaultDetected*** method
4	The SCS sends an equivalent message to the gsmSCF if appropriate
5	The gsmSCF sends a CAP ***ReleaseCall*** if appropriate

Table 4-53: Parameter Mapping

From: Dialogue Error	To: callFaultDetected
	callSessionID
TC_U_ABORT	fault

4.2.15 callEnded

callEnded will be invoked when the call has ended. Furthermore, the operation contains an indication on the reason why the call has been ended. Also the operation will always be invoked when the call has ended and not only when the application has requested its interest in this event.

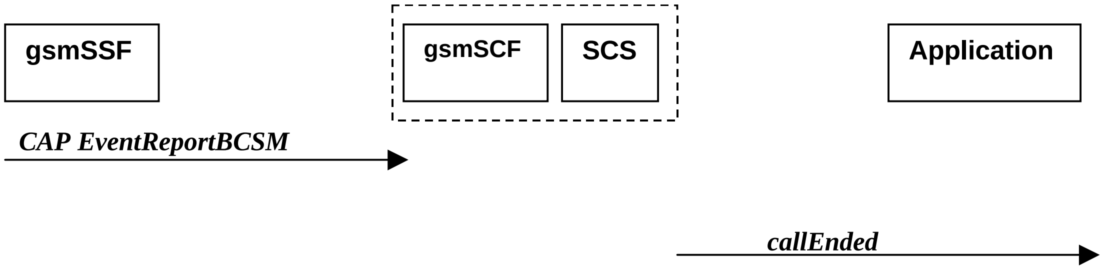

Figure 4-24: Call Flow for callEnded

Table 4-54: Normal Operation

Pre-conditions	There is an application monitoring the call in some way
1	The gsmSSF detects a release from the calling or called party leg. CAP eventReportBCSM is sent if requested by the gsmSCF The BCSM event indicated may be either abandon or disconnect depending on the phase of the call
2	The gsmSCF sends an equivalent message to the SCS
3	The SCS invokes the ***callEnded*** method

Table 4-55: Parameter Mapping

From: CAP EventReportBCSM	To: callEnded
	callSessionID
eventTypeBCSM	
	report
legID	callLegSessionID
eventSpecificInformationBCSM: releaseCause	cause
miscCallInfo	

5 Detailed Parameter Mappings

This clause contains detailed parameter mappings for data types that are used in the Parameter Mapping tables in the previous clauses.

5.1 TpCallMonitorMode

Table 5-1

TpCallMonotirMode	monitorMode
P_CALL_MONITOR_MODE_INTERRUPT	interrupted
P_CALL_MONITOR_MODE_NOTIFY	notifyAndContinue
P_CALL_MONITOR_MODE_DO_NOT_MONITOR	transparent

5.2 TpCallReportType

Table 5-2

TpCallReportType	eventTypeBCSM
P_CALL_REPORT_UNDEFINED	analyzedInformation
P_CALL_REPORT_PROGRESS	<no mapping available>
P_CALL_REPORT_ALERTING	<no mapping available>
P_CALL_REPORT_ANSWER	oAnswer tAnswer
P_CALL_REPORT_REFUSED_BUSY	oCalledPartyBusy tBusy
P_CALL_REPORT_NO_ANSWER	oNoAnswer tNoAnswer
P_CALL_REPORT_DISCONNECT	tDisconnect
P_CALL_REPORT_REDIRECTED	<no mapping available>
P_CALL_REPORT_SERVICE_CODE	<no mapping available>
P_CALL_REPORT_ROUTING_FAILURE	routeSelectFailure

5.3 TpCallEventName

Table 5-3

TpCallEventName	eventTypeBCSM
P_EVENT_NAME_UNDEFINED	<no mapping available>
P_EVENT_GCCS_OFFHOOK_EVENT	<no mapping available>
P_EVENT_GCCS_ADDRESS_COLLECTED_EVENT	O-CSI (see note) O-BcsmTriggerDetectionPoint: collectedInfo T-CSI/VT-CSI: T-BcsmTriggerDetectionPoint: termAttemptAuthorized
P_EVENT_GCCS_ADDRESS_ANALYSED_EVENT	O-CSI O-BcsmTriggerDetectionPoint analysedInfo
P_EVENT_GCCS_CALLED_PARTY_BUSY	T-CSI/VT-CSI: T-BcsmTriggerDetectionPoint: tBusy
P_EVENT_GCCS_CALLED_PARTY_UNREACHABLE	mapped to the cause value returned with TBusy : T-CSI/VT-CSI: T-BcsmTriggerDetectionPoint: tBusy
P_EVENT_GCCS_NO_ANSWER_FROM_CALLED_PARTY	T-CSI/VT-CSI: T-BcsmTriggerDetectionPoint: tNoAnswer
P_EVENT_GCCS_ROUTE_SELECT_FAILURE	O-CSI: O-BcsmTriggerDetectionPoint: routeSelectFailure
P_EVENT_GCCS_ANSWER_FROM_CALL_PARTY	T-CSI/VT-CSI: T-BcsmTriggerDetectionPoint: tAnswer
NOTE: O-CSI applies when the value for CallNotificationType is P_ORIGINATING, T-CSI applies when the value for CallNotificationType is P_TERMINATING.	

5.4 TpCallAdditionalReportInfo

Table 5-4

TpCallAdditionalReportInfo	eventSpecificInformationBCSM
RefusedBusy	oCalledPartyBusy busyCause or tBusySpecificInfo busyCause callForwarded (no mapping)
CallDisconnect	oDisconnectSpecificInfo - releaseCause tDisconnectSpecificInfo - releaseCause
ForwardAddress	oAnswerSpecificInfo - destinationAddress; or - Call (no mapping); - forwardedCall (no mapping). tAnswerSpecificInfo - destinationAddress; or - Call (no mapping); - forwardedCall (no mapping).
ServiceCode	<no mapping available>
RoutingFailure	routeSelectFailureSpecificInfo - failureCause
	tNoAnswerSpecificInfo - callForwarded

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 2001	CN_12	NP-010371	001	--	Missing description of "setCallLoadControl mapping to CAP" (N5-010432)	4.0.0	4.1.0
Sep 2001	CN_13	NP-010474	002	--	Updates and corrections to data mapping to CAP	4.1.0	4.2.0
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

History

Document history		
V6.0.0	December 2004	Publication